

# Pengertian Tipe Data String di dalam Pascal

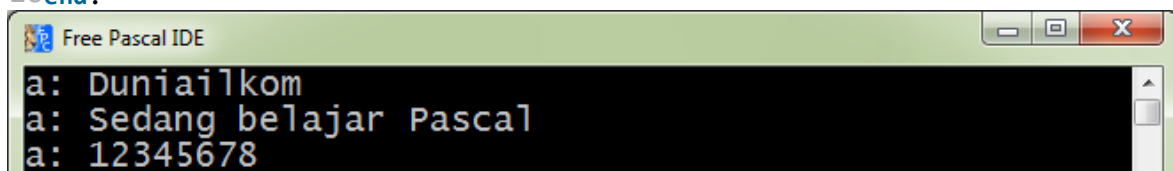
**Tipe data String** adalah tipe data yang bisa menampung banyak karakter sekaligus. Secara internal di dalam pascal, string merupakan **array** dari tipe data **char**.

Dalam compiler *Turbo Pascal*, maksimal jumlah karakter yang bisa ditampung di dalam string adalah sebanyak **255 karakter**. Compiler *FreePascal* menambahkan beberapa jenis tipe data lain seperti **AnsiString** yang bisa menampung lebih banyak karakter (tidak dibatasi).

## Cara Penggunaan Tipe Data String Pascal

Untuk mendeklarasikan sebuah variabel sebagai **string**, kita tinggal menambahkannya di bagian **var**. Berikut contoh penggunaannya:

```
1
2 program tipe_string;
3 uses crt;
4 var
5   a:string;
6 begin
7   clrscr;
8   a:='Duniaikom';
9   writeln('a: ',a);
10
11  a:='Sedang belajar Pascal';
12  writeln('a: ',a);
13
14  a:='12345678';
15  writeln('a: ',a);
16  readln;
17 end.
```



Pada kode program diatas, saya mendefinisikan variabel **a** sebagai **string**, kemudian mengubah nilainya sepanjang kode program dan menampilkan hasilnya menggunakan perintah **writeln**.

Secara default, tipe data **string** akan menyediakan tempat untuk 255 karakter. Kita bisa menetapkan seberapa besar 'ruang' yang disediakan. Caranya, dengan menambahkan angka di dalam kurung siku sebagai penanda jumlah karakter. Sebagai contoh, perintah **a: string[10]** akan membuat variabel **a** hanya bisa diisi dengan **10 karakter saja**. Jika karakter yang diinput lebih dari 10, sisanya akan dibuang. Berikut contohnya:

```

1
2program tipe_string;
3uses crt;
4var
5  a:string[10];
6begin
7  clrscr;
8  a:='DekaOKTV - Learn Programming';
9  writeln('a: ',a);
10
11 a:='Sedang belajar Pascal';
12 writeln('a: ',a);
13
14 a:='1234567890123456789';
15 writeln('a: ',a);
16 readln;
17end.

```

Seperti yang terlihat, karakter yang bisa disimpan kedalam variabel a hanya 10 karakter saja.

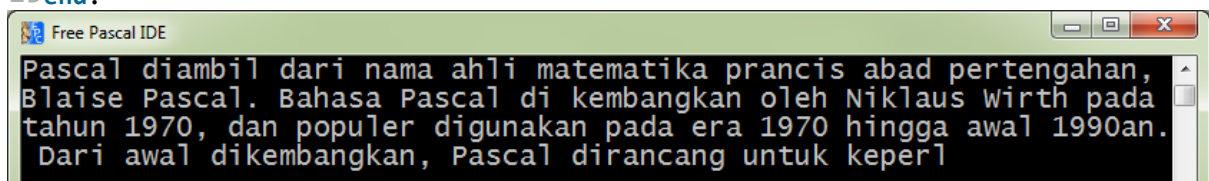
## Mengenal Tipe Data AnsiString FreePascal

Di dalam compiler **Turbo Pascal**, maksimal karakter yang bisa disimpan ke dalam string adalah **255**. Jika lebih dari itu, sisa karakter akan dibuang. Berikut contohnya:

```

1
2program tipe_string;
3uses crt;
4var
5  a:string;
6begin
7  clrscr;
8  a:='Pascal diambil dari nama ahli matematika prancis abad pertengahan, ' +
9    'Blaise Pascal. Bahasa Pascal di kembangkan oleh Niklaus Wirth pada ' +
10   'tahun 1970, dan populer digunakan pada era 1970 hingga awal 1990an. ' +
11   'Dari awal dikembangkan, Pascal dirancang untuk keperluan akademik. ' +
12   'Atas alasan itu pula Pascal masih digunakan di berbagai sekolah dan ' +
13   'universitas di seluruh dunia (termasuk Indonesia).';
14 writeln(a);
15 readln;
16end.

```



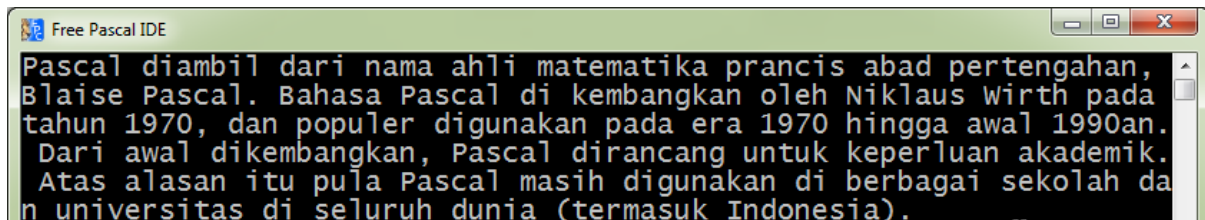
Variabel a saya isi dengan string yang cukup panjang, tanda ' + ' digunakan untuk menyambung sebuah string dengan string lain, sehingga saya bisa memecahnya

menjadi beberapa baris. Ini adalah operator penyambungan string di dalam pascal.

Seperti yang terlihat, kalimat tersebut akan terpotong pada karakter ke 255.

Compiler **FreePascal** memperkenalkan tipe data **AnsiString** yang tidak membatasi jumlah string. Berikut contoh penggunaannya:

```
1
2 program tipe_string;
3 uses crt;
4 var
5   a:ansistring;
6 begin
7   clrscr;
8   a:='Pascal diambil dari nama ahli matematika prancis abad pertengahan, ' +
9     'Blaise Pascal. Bahasa Pascal di kembangkan oleh Niklaus Wirth pada ' +
10    'tahun 1970, dan populer digunakan pada era 1970 hingga awal 1990an. ' +
11    'Dari awal dikembangkan, Pascal dirancang untuk keperluan akademik. ' +
12    'Atas alasan itu pula Pascal masih digunakan di berbagai sekolah dan ' +
13    'universitas di seluruh dunia (termasuk Indonesia).';
14   writeln(a);
15   readln;
16 end.
```



# Pengertian Tipe Data Boolean Pascal

**Tipe data boolean** sebenarnya sangat sederhana. Tipe data ini hanya bisa diisi dengan salah satu dari 2 nilai: **TRUE** atau **FALSE**. Tipe data **boolean** banyak digunakan untuk percabangan kode program, atau untuk memutuskan apa yang mesti dijalankan ketika sebuah kondisi terjadi.

Sebagai contoh, kita bisa membuat kode program untuk menentukan apakah sebuah angka genap atau ganjil berdasarkan input dari pengguna. Untuk keperluan ini kita harus mengecek terlebih dahulu apakah angka itu bisa dibagi 2 (untuk angka genap), atau tidak bisa dibagi 2 (untuk angka ganjil). Tipe data boolean bisa digunakan untuk menampung kondisi seperti ini, benar atau salah (*True* atau *False*).

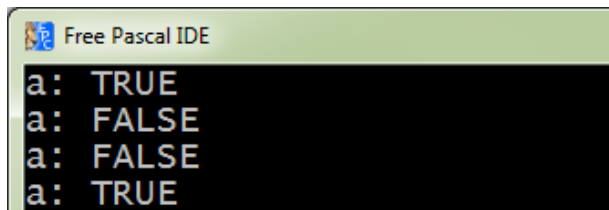
Contoh penggunaan tipe data boolean akan lebih jelas jika anda telah mempelajari kondisi percabangan program seperti IF (yang akan saya jelaskan pada tutorial terpisah). Untuk sementara anda hanya perlu memahami bahwa tipe data boolean adalah tipe data yang hanya memiliki 2 isi: **TRUE** atau **FALSE**.

---

## Cara Penggunaan Tipe Data Boolean Pascal

Langsung saja kita lihat contoh program untuk mendefenisikan dan mengisi variabel dengan tipe data **boolean** di pascal:

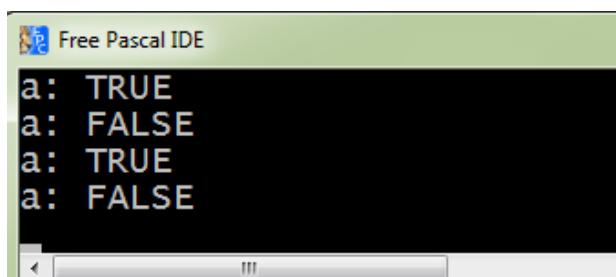
```
1
2 program tipe_boolean;
3 uses crt;
4 var
5   a: boolean;
6 begin
7   clrscr;
8
9   a:= true;
10  writeln('a: ',a);
11
12  a:= false;
13  writeln('a: ',a);
14
15  a:= not true;
16  writeln('a: ',a);
17
18  a:= not false;
19  writeln('a: ',a);
20  readln;
21 end.
```



```
a: TRUE
a: FALSE
a: FALSE
a: TRUE
```

Dalam contoh diatas, saya membuat variabel **a** sebagai **boolean**, kemudian mengisinya dengan **true** dan **false**. Pada contoh ketiga dan keempat, saya menggunakan perintah **not** untuk membalik nilai boolean. Perintah '**not true**' akan menghasilkan **false**, sedangkan '**not false**' akan menghasilkan **true**. Perhatikan bahwa nilai untuk boolean ini (**true** atau **false**) harus ditulis **tanpa tanda kutip**. Jika anda menambahkan tanda kutip, ini menjadi **string**. Dengan kata lain '**true**' tidak sama dengan **true** (perhatikan tanda kutipnya). Selain menulis langsung nilai boolean, kita juga bisa menginput hasilnya dari sebuah operasi perbandingan. Berikut contohnya:

```
1
2 program tipe_boolean;
3 uses crt;
4 var
5   a: boolean;
6 begin
7   clrscr;
8
9   a:= 1=1;
10  writeln('a: ',a);
11
12  a:= 9<8;
13  writeln('a: ',a);
14
15  a:= 999<>998;
16  writeln('a: ',a);
17
18  a:= 'dekaoktv' = 'dekaoktv.com';
19  writeln('a: ',a);
20  readln;
21 end.
```



```
a: TRUE
a: FALSE
a: TRUE
a: FALSE
```

Salah satu konsep yang harus anda pahami di dalam pemrograman adalah penggunaan tanda '='. Didalam pascal, tanda ini digunakan untuk operasi perbandingan, bukan pengisian ke dalam sebuah variabel (operasi *assignment*).

Dengan kata lain, "**a:= 1=1**" berarti kita mengisi nilai variabel **a** dengan hasil dari perbandingan "**1=1**". Jadi apakah **1=1**? Yup **benar**, yang di dalam bahasa inggris menjadi: **true**. Oleh karena itulah variabel **a** akan diisi dengan nilai boolean **true**. Pada contoh kedua, apakah "**9<8**"? **salah!**, karena 9 bukan lebih kecil dari delapan, sehingga hasilnya: **false**.

Berikutnya, apakah "**999<>998**"? Tanda "<>" di dalam pascal bisa dibaca sebagai "*tidak sama dengan*", sehingga hasilnya adalah: **true**.

Contoh terakhir, apakah '**dekaoktv**' = '**dekaoktv.com**'? **salah!**, karena untuk '*benar*', kedua string harus mengandung karakter yang sama persis. Sehingga hasilnya adalah **false**.

---

**Tipe data boolean** yang dipelajari disini akan lebih jelas penggunaannya jika anda telah mempelajari struktur logika percabangan kode program seperti **IF**. Saya akan menjelaskan struktur alur kode program ini setelah pembahasan mengenai tipe data dan operator pascal.

Berikutnya kita akan masuk ke **tipe data enumerated** di dalam pascal.

# Pengertian Tipe Data Boolean Pascal

**Tipe data boolean** sebenarnya sangat sederhana. Tipe data ini hanya bisa diisi dengan salah satu dari 2 nilai: **TRUE** atau **FALSE**. Tipe data **boolean** banyak digunakan untuk percabangan kode program, atau untuk memutuskan apa yang mesti dijalankan ketika sebuah kondisi terjadi.

Sebagai contoh, kita bisa membuat kode program untuk menentukan apakah sebuah angka genap atau ganjil berdasarkan input dari pengguna. Untuk keperluan ini kita harus mengecek terlebih dahulu apakah angka itu bisa dibagi 2 (untuk angka genap), atau tidak bisa dibagi 2 (untuk angka ganjil). Tipe data boolean bisa digunakan untuk menampung kondisi seperti ini, benar atau salah (*True* atau *False*).

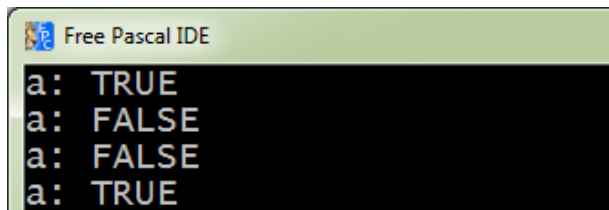
Contoh penggunaan tipe data boolean akan lebih jelas jika anda telah mempelajari kondisi percabangan program seperti IF (yang akan saya jelaskan pada tutorial terpisah). Untuk sementara anda hanya perlu memahami bahwa tipe data boolean adalah tipe data yang hanya memiliki 2 isi: **TRUE** atau **FALSE**.

---

## Cara Penggunaan Tipe Data Boolean Pascal

Langsung saja kita lihat contoh program untuk mendefenisikan dan mengisi variabel dengan tipe data **boolean** di pascal:

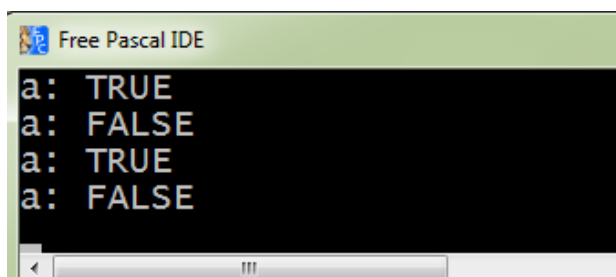
```
1
2 program tipe_boolean;
3 uses crt;
4 var
5   a: boolean;
6 begin
7   clrscr;
8
9   a:= true;
10  writeln('a: ',a);
11
12  a:= false;
13  writeln('a: ',a);
14
15  a:= not true;
16  writeln('a: ',a);
17
18  a:= not false;
19  writeln('a: ',a);
20  readln;
21 end.
```



```
a: TRUE
a: FALSE
a: FALSE
a: TRUE
```

Dalam contoh diatas, saya membuat variabel **a** sebagai **boolean**, kemudian mengisinya dengan **true** dan **false**. Pada contoh ketiga dan keempat, saya menggunakan perintah **not** untuk membalik nilai boolean. Perintah '**not true**' akan menghasilkan **false**, sedangkan '**not false**' akan menghasilkan **true**. Perhatikan bahwa nilai untuk boolean ini (**true** atau **false**) harus ditulis **tanpa tanda kutip**. Jika anda menambahkan tanda kutip, ini menjadi **string**. Dengan kata lain '**true**' tidak sama dengan **true** (perhatikan tanda kutipnya). Selain menulis langsung nilai boolean, kita juga bisa menginput hasilnya dari sebuah operasi perbandingan. Berikut contohnya:

```
1
2 program tipe_boolean;
3 uses crt;
4 var
5   a: boolean;
6 begin
7   clrscr;
8
9   a:= 1=1;
10  writeln('a: ',a);
11
12  a:= 9<8;
13  writeln('a: ',a);
14
15  a:= 999<>998;
16  writeln('a: ',a);
17
18  a:= 'dekaoktv' = 'dekaoktv.com';
19  writeln('a: ',a);
20  readln;
21 end.
```



```
a: TRUE
a: FALSE
a: TRUE
a: FALSE
```

Salah satu konsep yang harus anda pahami di dalam pemrograman adalah penggunaan tanda '='. Didalam pascal, tanda ini digunakan untuk operasi perbandingan, bukan pengisian ke dalam sebuah variabel (operasi *assignment*).



Dengan kata lain, "**a:= 1=1**" berarti kita mengisi nilai variabel **a** dengan hasil dari perbandingan "**1=1**". Jadi apakah **1=1**? Yup **benar**, yang di dalam bahasa inggris menjadi: **true**. Oleh karena itulah variabel **a** akan diisi dengan nilai boolean **true**. Pada contoh kedua, apakah "**9<8**"? **salah!**, karena 9 bukan lebih kecil dari delapan, sehingga hasilnya: **false**.

Berikutnya, apakah "**999<>998**"? Tanda "<>" di dalam pascal bisa dibaca sebagai "*tidak sama dengan*", sehingga hasilnya adalah: **true**.

Contoh terakhir, apakah '**dekaoktv**' = '**dekaoktv.com**'? **salah!**, karena untuk '*benar*', kedua string harus mengandung karakter yang sama persis. Sehingga hasilnya adalah **false**.

---

**Tipe data boolean** yang dipelajari disini akan lebih jelas penggunaannya jika anda telah mempelajari struktur logika percabangan kode program seperti **IF**. Saya akan menjelaskan struktur alur kode program ini setelah pembahasan mengenai tipe data dan operator pascal.

Berikutnya kita akan masuk ke **tipe data enumerated** di dalam pascal.

# Pengertian Tipe Data Enumerated Pascal

**Tipe data enumerated** adalah tipe data khusus yang isinya bisa kita tentukan sendiri. Sebagai contoh, kita bisa membuat tipe *enumerated* '**warna**' yang berisi: *merah, kuning, biru* dan *ungu*. Atau tipe *enumerated* '**nama\_hari**' yang berisi nama-nama hari seperti *senin, Selasa, Rabu*, dst.

Tipe data *enumerated* sering juga disebut **enumeration** atau **enum** saja.

---

## Cara Penggunaan Tipe Data Enumerated Pascal

Untuk membuat tipe data **enumerated** di pascal, kita harus menggunakan perintah **type**, baru kemudian mendeklarasikannya menggunakan perintah **var**. Sebagai contoh, jika saya ingin membuat tipe enum "**nama\_hari**" yang berisi nama-nama hari, harus mendefinisikannya sebagai berikut:

```
1 type
```

```
2   nama_hari= (senin,selasa,rabu,kamis,jumat,sabtu,minggu);
```

Setelah itu, baru diberikan ke dalam variabel:

```
1 var
```

```
2   a,b: nama_hari;
```

Sekarang variabel **a** dan **b** berisi tipe enum **nama\_hari**. Berikut contoh kode program lengkapnya:

```
1
2 program tipe_enumerated;
3 uses crt;
4 type
5   nama_hari= (senin,selasa,rabu,kamis,jumat,sabtu,minggu);
6 var
7   a,b: nama_hari;
8 begin
9   clrscr;
10
11   a:= senin;
12   writeln('a: ',a);
13
14   b:= minggu;
15   writeln('b: ',b);
16   readln;
17 end.
```

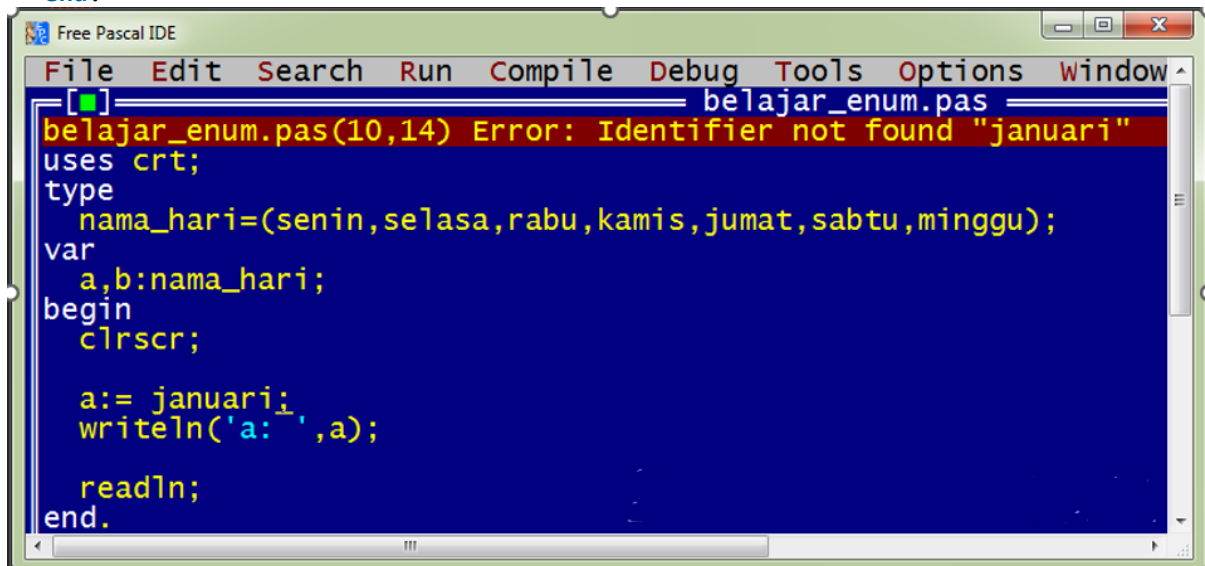


```
a: senin
b: minggu
```

Perhatikan urutan pendefinisian tipe enum '**nama\_hari**', dimana kita harus membuat isinya terlebih dahulu (menggunakan *keyword* **type**), baru kemudian diberikan ke dalam variabel (menggunakan *keyword* **var**).

Variabel **a** dan **b** dalam contoh diatas hanya bisa diisi dengan nilai yang ada di dalam tipe data enum **nama\_hari**. Jika saya memberikan nama lain, compiler pascal akan menghasilkan **error**, seperti contoh berikut:

```
1
2 program tipe_enumerated;
3 uses crt;
4 type
5     nama_hari= (senin,selasa,rabu,kamis,jumat,sabtu,minggu);
6 var
7     a,b: nama_hari;
8 begin
9     clrscr;
10
11     a:= januari;
12     writeln('a: ',a);
13     readln;
14 end.
```



```
File Edit Search Run Compile Debug Tools Options Window
belajar_enum.pas
belajar_enum.pas(10,14) Error: Identifier not found "januari"
uses crt;
type
    nama_hari=(senin,selasa,rabu,kamis,jumat,sabtu,minggu);
var
    a,b:nama_hari;
begin
    clrscr;

    a:= januari;
    writeln('a: ',a);

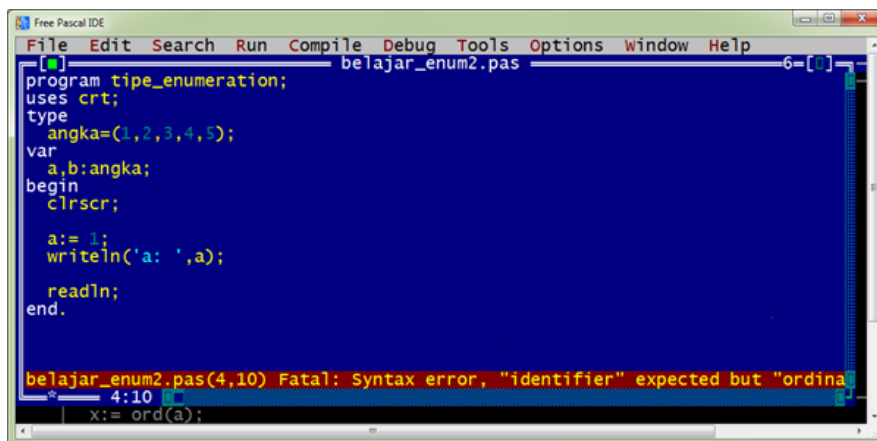
    readln;
end.
```

Pada contoh diatas, saya mencoba memberikan nilai '**januari**' ke dalam variabel **a**. Hasilnya, compiler *FreePascal* akan menghasilkan **error** karena '**januari**' tidak ada di dalam nama hari.

Selain itu perhatikan bahwa walaupun nama-nama hari diatas adalah kumpulan karakter (*string*), kita tidak perlu memberikan tanda kutip.

Salah satu sifat tipe data enumerated di pascal, kita tidak bisa membuat nilai angka sebagai bagian dari tipe data **enumerated**. Contoh berikut akan menghasilkan error:

```
1
2 program tipe_enumerated;
3 uses crt;
4 type
5   angka= (1,2,3,4,5);
6 var
7   a,b: angka;
8 begin
9   clrscr;
10  a:= 1;
11  writeln('a: ',a);
12
13  readln;
14 end.
```



Error diatas terjadi karena saya memberikan nilai angka (integer): 1,2,3,4,5 untuk tipe data enum 'angka'. Jika kita ingin membuat batasan seperti ini bisa menggunakan tipe data **subrange** (akan dijelaskan dalam tutorial setelah ini). Di dalam sebuah kode program pascal, kita juga bisa membuat lebih dari 1 tipe enum, seperti contoh berikut:

```
1 program tipe_enumerated;
2 uses crt;
3 type
4   warna= (merah,kuning,biru,ungu,jingga);
5   angka= (satu,dua,tiga,empat,lima);
6 var
7   a: warna;
8   b: angka;
9 begin
10  clrscr;
11
12  a:= kuning;
13  writeln('a: ',a);
```

```
13
14 b:= tiga;
15 writeln('b: ',b);
16
17 readln;
18 end.
19
```



Pada contoh diatas, saya membuat 2 buah tipe data **enumerated** yang berisi nama-nama warna dan nama angka, kemudian menampilkan isinya.

---