



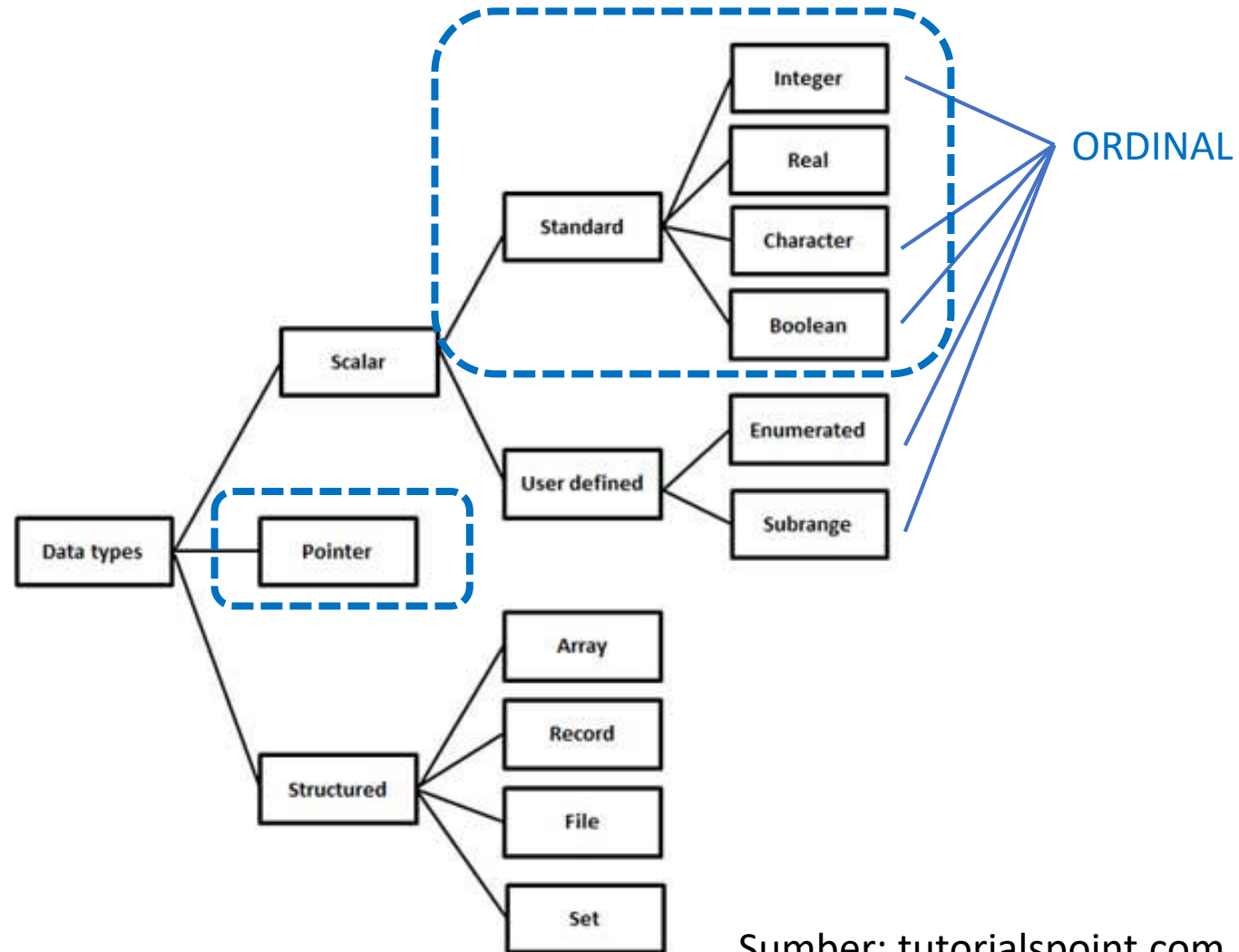
POLITEKNIK STATISTIKA STIS

For Better Official Statistics

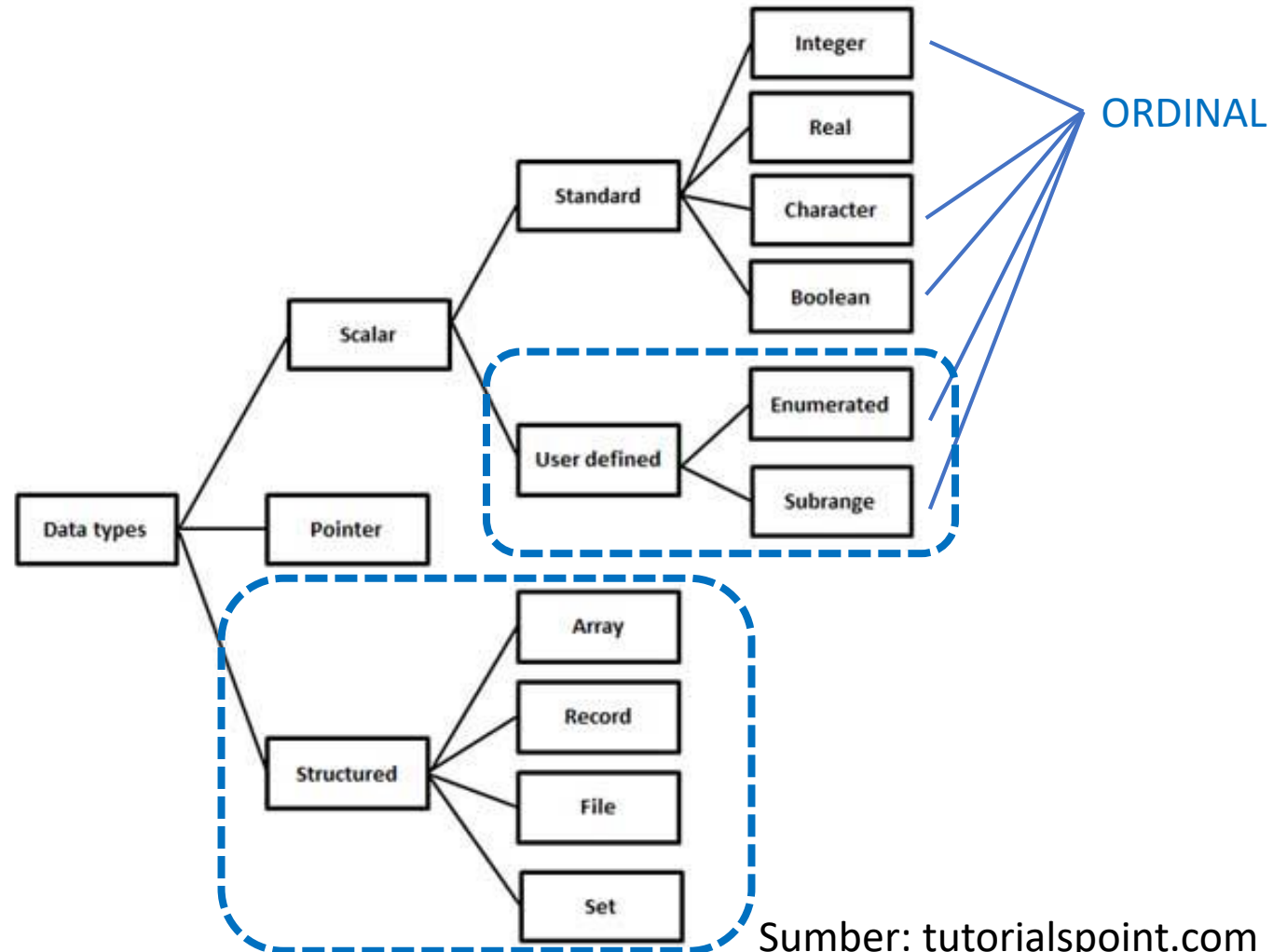
Enumerated & Subrange

Nori Wilantika, S.S.T., M.T.I.





Sumber: tutorialspoint.com



- Tipe data sederhana terbagi menjadi 2 bagian:
 - tipe data standar
 - tipe data yang ditentukan oleh user (user-defined)
- Tipe data standard termasuk di dalamnya INTEGER, CHAR, dan BOOLEAN yang menjadi pembahasan di pertemuan-pertemuan sebelumnya.
- Sedangkan *User-Defined type* terdiri dari enumerated (terbilang) dan subrange (jangkauan).
- Untuk tipe data lainnya yang masuk dalam kategori tipe data terstruktur seperti array akan dibahas pada pertemuan selanjutnya.



ENUMERATED



- Enumerations adalah daftar nilai yang didefinisikan oleh pemrogram sesuai dengan kebutuhannya.
- Contoh:
 - ‘**warna**’ yang berisi: merah, kuning, biru dan ungu.
 - ‘**nama_hari**’ yang berisi nama-nama hari seperti *senin, selasa, rabu*, dst.
- Contoh penggunaannya pada variabel dalam Pascal:

```
VAR  
    BULAN: (JAN, FEB, MAR, APR);  
           0     1     2     3
```
- Nilai yang ada di dalam daftar direpresentasikan dalam *small integer* 0 sampai n-1.

VAR

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

Agar enumerations tersebut menjadi sebuah tipe data baru di pascal, kita harus menggunakan perintah **type**, baru kemudian mendeklarasikannya menggunakan perintah **var**.

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

Membuat tipe enum
"nama_hari"

```
1 | var
2 |     a,b: nama_hari;
```

Menggunakan "nama_hari"
pada deklarasi variabel

Dengan menggunakan *enumerated data types* atau **tipe data enum** atau **tipe data terbilang**, pemrogram dapat mendefinisikan suatu tipe data yang nilainya belum ada sebelumnya.

- Enum termasuk tipe data ordinal. Kita bisa menggunakan operasi ORD, PRED dan SUCC terhadap elemen yang ada di dalam enum.

Operation	Name
Equity	=
Inequity	$\langle \rangle$
Less than	<
Less than, equal to	\leq
Greater than	>
Greater than, equal to	\geq
Predecessor	pred
Successor	succ
Ordinal Number	ord


```
Program Ordnum;
```

```
Var
```

```
    Brand : (Lenovo, HP, Asus, Acer); {ada 4 item}
```

```
Begin
```

```
    Writeln(ORD(Lenovo)); {output 0}
```

```
    Writeln(ORD(HP)); {output 1}
```

```
    Writeln(ORD(Asus)); {output 2}
```

```
    Writeln(ORD(Acer)); {output 3}
```

```
    Writeln(PRED(Asus)); {output HP}
```

```
    Writeln(SUCC(Asus)); {output Acer}
```

```
End.
```

```
PROGRAM EnumLoop (OUTPUT) ;  
VAR  
    Month : (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec) ;  
BEGIN  
    WRITELN ;  
    FOR Month := Jan TO Dec DO  
        WRITE (ORD (Month) , ' ' )  
    END .
```

variabel enum bisa dijadikan ketukan dalam perulangan, karena variabel ini termasuk dalam variabel ordinal

0 1 2 3 4 5 6 7 8 9 10 11

```
Program ColoursWithCountries;
uses Wincrt;
```

```
TYPE
```

```
    Colours = (Red,Blue,Black,Orange,Green,White);
    Countries = (France,Italy,Greece,Spain);
```

```
VAR
```

```
    Colour : Colours;
    Country : Countries;
    i : Integer;
```

```
Begin (* main *)
```

```
    For Colour := White downto Orange do
```

```
        For Country := Italy to Spain do
```

```
            Begin
```

```
                i := ORD(Colour) + ORD(Country);
                writeln(i,succ(i):4)
```

```
            End;
```

```
End.
```

OUTPUT

White	Italy	i=5+1	6	7
	Greece	i=5+2	7	8
	Spain	i=5+3	8	9
Green	Italy	i=4+1	5	6
	Greece	i=4+2	6	7
	Spain	i=4+3	7	8
Orange	Italy	i=3+1	4	5
	Greece	i=3+2	5	6
	Spain	i=3+3	6	7

```
program tipe_enumerated;  
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.
```

Variabel dengan tipe data enum hanya bisa diisi dengan nilai yang ada di dalam enumeration yang telah didefinisikan.

```
program tipe_enumerated;  
uses crt;  
type  
    angka= (1,2,3,4,5);  
var  
    a,b: angka;  
begin  
    clrscr;  
  
    a:= 1;  
    writeln('a: ',a);  
  
    readln;  
end.
```

Error diatas terjadi karena kita memberikan nilai angka (**integer**): 1,2,3,4,5 untuk tipe data enum. Jika ingin membuat batasan nilai seperti ini, bisa menggunakan tipe data subrange.



SUBRANGE



- Subrange dapat digunakan untuk mengambil sebagian nilai dari tipe data yang sudah didefinisikan sebelumnya.
- Subrange dapat menghindarkan suatu nilai variabel melebihi batas yang ingin kita tentukan (*out of range*). Nilai yang berada di luar range tidak diperbolehkan.
- Keuntungan lainnya yaitu dapat menghemat penggunaan memori dan program terdokumentasi lebih baik.

- Contoh penggunaannya pada variabel dalam Pascal:

VAR

Month : 1..12; {*INTEGER*}

Uppercase: 'A'..'Z'; {*CHAR*}

Lowercase: 'a'..'z';


```
PROGRAM Subrange1 (INPUT,OUTPUT) ;
VAR
    MonthNumber :1..12;

BEGIN
    WRITE ('Please enter the number of the month: ');
    READLN (MonthNumber) ;

    CASE MonthNumber OF
        12, 1, 2 :WRITELN('This is wintertime. ');
        3, 4, 5 :WRITELN('This is springtime. ');
        6, 7, 8 :WRITELN('This is summertime. ');
        9, 10, 11 :WRITELN('This is autumn. ')
    END
END.
```

- Untuk membuat tipe data subrange, caranya hampir sama dengan tipe data enumerated, dimana kita harus mendefinisikannya di bagian **type**, baru kemudian menggunakannya pada deklarasi variabel di bagian **var**.
- Tipe data subrange juga merupakan tipe data ordinal.

```
program tipe_subrange;
uses crt;
type
    satuan= 1..9;
var
    a,b: satuan;
begin
    clrscr;

    a:= 2;
    writeln('a: ',a);

    b:= 7;
    writeln('b: ',b);

    readln;
end.
```

```
program totalmark;
type
    mark=1..100;
var
    s1,s2,s3:mark;
    total:integer;
begin
    writeln('Enter 3 marks');
    readln(s1,s2,s3);
    total:=s1+s2+s3;
    writeln('Total marks:',total);
end.
```

```
Program DisplayMonths;
uses wincrt;
CONST
    MaxMonth = 12;
TYPE
    Month = 1..MaxMonth;
VAR
    m : Month;

Begin (* main *)
For m := 1 to MaxMonth do
    Case m of
        1 : writeln('January');
        2 : writeln('February');
        3 : writeln('March');
        ...
        12: writeln('December');
    End
End.
```

- Variabel dengan tipe data subrange dapat menjadi subset dari variable dengan tipe enumerated.

VAR

months : (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec);

Summer : Apr .. Aug;

Winter : Oct .. Dec;

- Kita bisa membuat tipe data subrange yang isinya merupakan subset dari tipe data enumerated.

type

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

hari_kerja= senin..jumat;

weekend= sabtu..minggu;

var

a:hari_kerja;

b:weekend;

```
Program MyClass;  
TYPE  
    Grades = (A,B,C,D,E,F);  
    PassMark = A..D;  
VAR  
    Pass : PassMark;  
Begin (* main *)  
    For Pass := A to D do  
        Case Pass of  
            A : writeln('Excellent');  
            B : writeln('Very Good');  
            C : writeln('Good');  
            D : writeln('Pass')  
        End  
    End.  
End.
```

- Dalam subrange, elemen pertama (batas bawah) harus lebih kecil.
- Subrange dapat menjadi subset dari enumeration. Tetapi enumeration tidak dapat digunakan oleh enumeration lainnya.
- Elemen pada enumeration tidak dapat digunakan sebagai identifier untuk variabel lainnya. Hal ini seperti mendeklarasikan variabel yang sama dua kali dalam satu program.



TYPE



- Bagian TYPE digunakan untuk mendeklarasikan tipe data baru yang ditentukan oleh user (user-defined) atau untuk mengganti nama pada tipe yang sudah ada.

- Sintaks dari deklarasi type adalah:

TYPE

type-name = type-definition;

- Pada beberapa compiler Pascal, penulisan type mengikuti urutan berikut:

LABEL section

CONST section

TYPE section

VAR section

- Beberapa compiler tidak mewajibkan urutan tersebut, namun type harus mendahului var.

- Pascal memungkinkan kita mengganti suatu tipe data dengan nama baru, misalnya tipe data INTEGER berikut:

TYPE

BilBulat=Integer;

- Pada deklarasi ini kita mengganti tipe data **Integer** dengan nama baru **BilBulat**. Kita bisa menggunakan tipe data ini pada bagian VAR.

VAR

Nilai, Umur, Tinggi: BilBulat;

- Enumerated dan subrange dapat dideklarasikan di bagian TYPE, seperti pada contoh berikut:

TYPE

Day = (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday);

WorkingDay = Monday..Friday;

- Workingday bertipe subrange yang merupakan subset dari Day yang memiliki tipe enumerated



LATIHAN



1. Deklarasi mana yang valid apabila semua dibuat dalam satu program?

```
TYPE
{1} Football = (Saints, Cowboys);
{2} Games = (Football, Baseball, Basketball)
{3} Week = (Mon, Tue, Wed, Thu, Fri, Sat, Sun);
{4} Weekend = Sat..Sun;
{5} Compiler = (C, Pascal, Fortran, Ada, Basic);
VAR
{6} WholeWeek : Week;
{7} WorkingDay : (Mon, Tue, Wed, Fri); sudah ada
{8} Weekday : Mon..Fri;
{9} SW : (Compiler, OperatingSystem,
    ApplicationProgram);
{10} DpTools : (Hardware, Software, PeopleWare);
{11} DpTool : (HW, SW, PW);
{12} C : (TurboC, QuickC);
{13} Margin : -10..+10;
```

2. Apakah hasil dari pernyataan writeln berikut:

TYPE

FamilyMember = (Sam, Camelia, Hazem, Craig, Sally);

BEGIN

{a} WRITELN(Sam < Sally);

{b} WRITELN(Sam > Sally);

{c} WRITELN(Camelia > Craig);

{d} WRITELN(ORD(Craig)-1 = ORD(Hazem));

{e} WRITELN(ORD(Sam)+4 = ORD(Sally));

END.

3. Manakah yang tidak benar dalam mendeklarasikan TYPE

TYPE

a) Range1 = 1..100 DIV 2; ✓

b) Range2 = 1..7; ✓

c) Range3 = 1.0..7.0; ✗

d) Range4 = 1..(100/2); ✗

e) TVSystem = (SECAM, PAL, NTSC);

f) ProgrammingLanguage=(C, C++, CSharp, Pascal, Fortran, COBOL);

g) int = INTEGER;

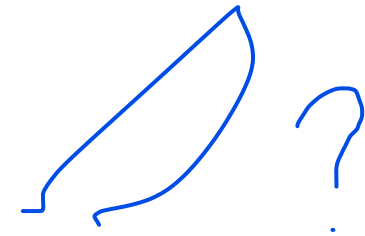
h) float = REAL j

4. Misalkan terdapat pendeklarasian sebagai berikut

```
TYPE
    ColorComp = (Cyan, Magenta, Yellow);
VAR
    MyColor, YourColor :ColorComp;
```

Tentukan mana instruksi yang tidak benar

- a) MyColor := Yellow; ✓
- b) YourColor := Cyan + 1; ✗
- c) WRITELN(ORD(Cyan), ' ', ORD(MyColor)); ✓
- d) WRITELN(Magenta); ✓





POLITEKNIK STATISTIKA STIS

For Better Official Statistics

TERIMA KASIH

