



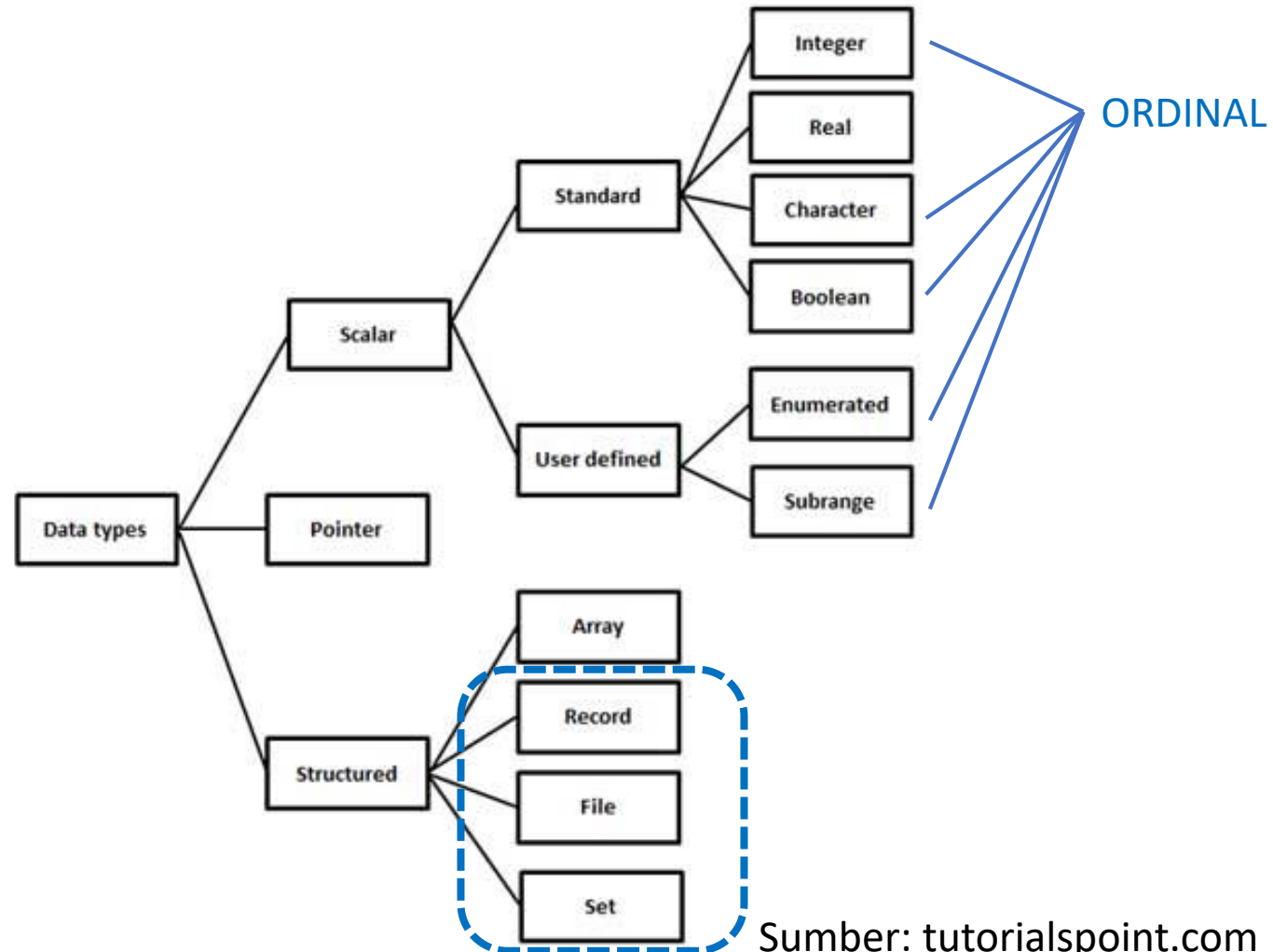
POLITEKNIK STATISTIKA STIS

For Better Official Statistics

RECORD, SET, FILE

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







RECORD



- Pascal menyediakan suatu tipe data **terstruktur** yang disebut dengan Record.
- Tidak seperti Array dimana semua elemennya harus bertipe sama, isi Record boleh berbeda-beda tipe data antara satu dengan lainnya.
- Masing-masing item data pada record disebut dengan Field. Jadi Record terdiri dari kumpulan Field yang dapat berbeda tipe.
- Deklarasi record:

```
Var  
    nama_record : Record  
        namafield-1 : Type  
        namafield-2 : Type  
        . . .  
        namafield-n : Type  
End;
```

```
Type  
    nama_record = Record  
        namafield-1 : Type  
        namafield-2 : Type  
        . . .  
        namafield-n : Type  
End;
```

```
Var  
    nama_var: nama_record;
```

```
var
pegawai      : record
    kd_peg    : string[5];
    nama      : string[15];
    alamat    : string[20];
    gaji      : longint;
end;
```

```
type
    data_pegawai = record
        id_peg    : string[5];
        nama      : string[15];
        alamat    : string[20];
        gaji      : longint;
    end;
var
    pegawai : data_pegawai;
```


Setiap field dalam record dapat diakses menggunakan nama record dan nama field yang dipisahkan oleh titik.

```
program tipe_record;
uses crt;
var
  siswa: record
    nama: string[20];
    umur: integer;
    sekolah: string[30];
    kota: string[20];
  end;
begin
  clrscr;

  siswa.nama:= 'Anto';
  siswa.umur:= 17;
  siswa.sekolah:= 'SMA 1 Durian Runtuh';
  siswa.kota:= 'Pelembang';

  writeln('Nama : ',siswa.nama);
  writeln('Umur : ',siswa.umur);
  writeln('Sekolah : ',siswa.sekolah);
  writeln('Kota : ',siswa.kota);

  readln;
end.
```

Output:

Nama : Anto
Umur : 17
Sekolah : SMA 1 Durian Runtuh
Kota : Pelembang

type

```
data_pegawai = record
    id_peg    : string[5];
    nama      : string[15];
    alamat    : string[20];
    gaji      : longint;
```

end;

var

```
pegawai : data_pegawai;
```

begin

```
pegawai.id_peg := '001';
pegawai.nama   := 'Andi Sujarwo';
pegawai.alamat:= 'Jl. Kyi Telingsing No 56 Kudus';
pegawai.gaji:=1500000;
writeln('Kode Pegawai  :',pegawai.id_peg);
writeln('Nama      :',pegawai.nama);
writeln('Alamat    :',pegawai.alamat);
writeln('Gaji      :',pegawai.gaji);
readln;
```

end.

Output:

```
Kode Pegawai  :001
Nama :Andi Sujarwo
Alamat :Jl. Kyi Telingsing N
Gaji  :1500000
```

- Penggunaan Field didalam Record seperti contoh sebelumnya dapat menyebabkan statement menjadi panjang, sebagai berikut : `pegawai.id_peg`.
- Akan lebih mudah dan lebih pendek bila pengenalan Record tidak selalu harus ditulis, sehingga statement diatas dapat berbentuk : `id_peg`.
- Hal ini dapat dilakukan dengan menggunakan statement `With` yang menyebutkan identifier Recordnya di awal, sehingga penggunaan Field-field selanjutnya sudah tidak perlu menyebutkan pengenalan Recordnya kembali.
- statement `With` digunakan untuk mempersingkat penulisan dalam pembacaan field.
- Penulisan :

with *namaRecord* **do**

Program pada contoh sebelumnya untuk menyimpan data pegawai bila diubah menggunakan statement With menjadi seperti berikut ini:

```
begin
with pegawai do
begin
id_peg := '001';
nama := 'Andi Sujarwo';
alamat:= 'Jl. Kyi Telingsing No 56 Kudus';
gaji:=1500000;
writeln('Kode Pegawai  :',id_peg);
writeln('Nama  :',nama);
writeln('Alamat  :',alamat);
writeln('Gaji  :',gaji);
readln;
end;
end.
```

Dalam contoh sebelumnya penggunaan tipe data record digunakan untuk menyimpan satu record.

```
program exRecords;
type
Books = record
    title: string[50];
    author: string[50];
    subject: string[100];
    book_id: longint;
end;

var
    (* Declare Book1 and Book2 of type Books *)
    Book1, Book2: Books;

begin
    (* book 1 specification *)
    Book1.title := 'C Programming';
    Book1.author := 'Nuha Ali ';
    Book1.subject := 'C Programming Tutorial';
    Book1.book_id := 6495407;

    (* book 2 specification *)
    Book2.title := 'Pascal Programming';
    Book2.author := 'Zara Ali';
    Book2.subject := 'Pascal Programming Tutorial';
    Book2.book_id := 6495700;
```

```
(* print Book1 info *)
writeln('Book 1 title : ', Book1.title);
writeln('Book 1 author : ', Book1.author);
writeln('Book 1 subject : ', Book1.subject);
writeln('Book 1 book_id : ', Book1.book_id);
writeln;

(* print Book2 info *)
writeln('Book 2 title : ', Book2.title);
writeln('Book 2 author : ', Book2.author);
writeln('Book 2 subject : ', Book2.subject);
writeln('Book 2 book_id : ', Book2.book_id);
end.
```

Output:

```
Book 1 title : C Programming
Book 1 author : Nuha Ali
Book 1 subject : C Programming Tutorial
Book 1 book_id : 6495407
```

```
Book 2 title : Pascal Programming
Book 2 author : Zara Ali
Book 2 subject : Pascal Programming Tutorial
Book 2 book_id : 6495700
```

Untuk dapat menyimpan sejumlah record maka dapat digunakan array yang bertipe record yang sudah didefinisikan.

```
program exRecords;
uses crt;
type
Books = record
    title: string[50];
    author: string[50];
    subject: string[100];
    book_id: longint;
end;

var
    Book: array [1..10] of Books;

begin
    clrscr;
    (* book 1 specification *)
    Book[1].title := 'C Programming';
    Book[1].author := 'Nuha Ali';
    Book[1].subject := 'C Programming Tutorial';
    Book[1].book_id := 6495407;

    (* book 2 specification *)
    Book[2].title := 'Pascal Programming';
    Book[2].author := 'Zara Ali';
    Book[2].subject := 'Pascal Programming Tutorial';
    Book[2].book_id := 6495700;
```

```
(* print Book1 info *)
writeln ('Book 1 title : ', Book[1].title);
writeln('Book 1 author : ', Book[1].author);
writeln('Book 1 subject : ', Book[1].subject);
writeln('Book 1 book_id : ', Book[1].book_id);
writeln;

(* print Book2 info *)
writeln ('Book 2 title : ', Book[2].title);
writeln('Book 2 author : ', Book[2].author);
writeln('Book 2 subject : ', Book[2].subject);
writeln('Book 2 book_id : ', Book[2].book_id);

end.
```

```
program tipe_record_array;
uses crt;
```

```
type
  rec_siswa =
    record
      nama: string[20];
      umur: integer;
      sekolah: string[30];
      kota: string[20];
    end;
```

```
var
  siswa: array [0..5] of rec_siswa;
```

```
begin
  clrscr;
```

```
  siswa[0].nama:= 'Anto';
  siswa[0].umur:= 17;
  siswa[0].sekolah:= 'SMA 1 Durian Runtuh';
  siswa[0].kota:= 'Pelembang';
```

```
  writeln('Nama : ',siswa[0].nama);
  writeln('Umur : ',siswa[0].umur);
  writeln('Sekolah : ',siswa[0].sekolah);
  writeln('Kota : ',siswa[0].kota);
```

```
  writeln;
```

```
  siswa[1].nama:= 'Santi';
  siswa[1].umur:= 18;
  siswa[1].sekolah:= 'SMA 1 Merdeka';
  siswa[1].kota:= 'Makasar';
```

```
  writeln('Nama : ',siswa[1].nama);
  writeln('Umur : ',siswa[1].umur);
  writeln('Sekolah : ',siswa[1].sekolah);
  writeln('Kota : ',siswa[1].kota);
```

```
  readln;
end.
```

Output:

```
Nama : Anto
Umur : 17
Sekolah : SMA 1 Durian Runtuh
Kota : Pelembang
```

```
Nama : Santi
Umur : 18
Sekolah : SMA 1 Merdeka
Kota : Makasar
```

- Jika dalam suatu record terdapat beberapa field yang sama tipenya, dapat digunakan array.
- Contoh terdapat data barang yang mempunyai struktur.
 - Nama barang -> bertipe String
 - Jumlah unit barang ke 1 -> bertipe Byte
 - Jumlah unit barang ke 2 -> bertipe Byte
 - Jumlah unit barang ke 3 -> bertipe Byte


```
uses crt;
Type barang= RECORD
    nmbrg:string[20];
    jmlbrg:array[1..3]of byte;
    hrgbrg:real;
    total:real;
end;

var jual:barang;
    tbarang, i:integer;

Begin
clrscr;
write('Nama Barang :');readln(jual.nmbrg);
for i:=1to 3 do
begin
write('Jumlah barang ',i,' : ');readln(jual.jmlbrg[i]);
tbarang:=tbarang+jual.jmlbrg[i];
end;
write('Harga barang :');readln(jual.hrgbrg);
jual.total:=tbarang*jual.hrgbrg;
writeln('Total Harga Barang = ', jual.total:10:2);
end.
```

- Dalam Pascal, tipe data record dapat didefinisikan juga sebagai field dari suatu record.
- Artinya suatu record dapat juga mempunyai field yang merupakan record (*Nesting Record*).
- Contoh, sebuah data pegawai mempunyai struktur sebagai berikut:

Nama Pegawai	
Mulai Masuk	Tanggal
	Bulan
	Tahun
Alamat Pegawai	Jalan
	Kota
Gaji Pegawai	Gaji Pokok
	Lembur
	Tunjangan

```
type
  datatype = record
    tgl : 1..31;
    bln : 1..12;
    thn : integer;
  end;

  addresstype = record
    jalan : string[20];
    kota : string[10];
  end;

  salarytype = record
    pokok,tunjangan,lembur : real;
  end;

  datapegawai = record
    nama : string[20];
    tglmasuk : datatype;
    alamat : addresstype;
    gaji : salarytype;
  end;

var pegawai:array[1..10] of datapegawai;
```

TYPE

```
AddressRecord = RECORD
    Street :STRING[18];
    City :STRING[15];
    State :STRING[2];
    Zip :String[5];
END;
```

```
EmployeeRecord = RECORD
    Name :STRING[25];
    AddressRec :AddressRecord;
    Phone :STRING[12];
    Rate :REAL;
    MaritalStatus :CHAR;
END;
```

VAR

```
EmployeeRec :EmployeeRecord;
```

Begin

```
clrscr;
EmployeeRec.Name := 'Jean L. Krauss';
EmployeeRec.AddressRec.Street := '15 Darell Street';
EmployeeRec.AddressRec.City := 'Bloomington';
EmployeeRec.AddressRec.State := 'IL';
```

```
EmployeeRec.AddressRec.Zip := '60108';
EmployeeRec.Phone := '312-987-5432';
EmployeeRec.Rate := 27.5;
EmployeeRec.MaritalStatus := 'M';
```

```
Writeln('Employee Name: ', EmployeeRec.Name);
Write('Address: ', EmployeeRec.AddressRec.Street);
Write(', ', EmployeeRec.AddressRec.City);
Write(', ', EmployeeRec.AddressRec.State);
Writeln(', ', EmployeeRec.AddressRec.Zip);
Writeln('Telephone #: ', EmployeeRec.Phone);
Writeln('Hourly Rate: $', EmployeeRec.Rate:0:2);
Writeln('Marital Status: ', EmployeeRec.MaritalStatus);
end.
```

Output:

```
Employee Name: Jean L. Krauss
Address: 15 Darell Street, Bloomington, IL, 60108
Telephone #: 312-987-5432
Hourly Rate: $27.50
Marital Status: M
```

- Mentransfer record sebagai argumen subprogram dilakukan dengan cara yang sama saat mentransfer variabel lainnya.
- Di dalam subprogram, mengakses record dilakukan dengan cara yang sama seperti pada contoh-contoh sebelumnya


```
program exRecords;
uses crt;
type
Books = record
    title: string[50];
    author: string[50];
    subject: string[100];
    book_id: longint;
end;

var
(* Declare Book1 and Book2 of type Books *)
Book1, Book2: Books;

(* procedure declaration *)
procedure printBook( var book: Books );
begin
    (* print Book info *)
    writeln( 'Book title : ', book.title);
    writeln( 'Book author : ', book.author);
    writeln( 'Book subject : ', book.subject);
    writeln( 'Book book_id : ', book.book_id);
end;
```

```
begin
    clrscr;
    (* book 1 specification *)
    Book1.title := 'C Programming';
    Book1.author := 'Nuha Ali ';
    Book1.subject := 'C Programming Tutorial';
    Book1.book_id := 6495407;

    (* book 2 specification *)
    Book2.title := 'Pascal Programming';
    Book2.author := 'Zara Ali';
    Book2.subject := 'Pascal Programming Tutorial';
    Book2.book_id := 6495700;

    (* print Book1 info *)
    printbook(Book1);
    writeln;

    (* print Book2 info *)
    printbook(Book2);
end.
```

- Apakah output yang dihasilkan program berikut?

```
Uses Wincrt;
Type mhs = record
    NIM : String[4];
    Nama : String[20];
    Prodi : String[20];
    IP : Real;
End;

Var data : mhs;
Begin
    With data do
        Begin
            Write('NIM : ');Readln(NIM);
            Write('Nama : ');Readln(Nama);
            Write('Program Studi : ');Readln(Prodi);
            Write('IP : ');Readln(IP);
        End;
        Writeln;
        Writeln;
        Writeln('NIM : ',data.NIM);
        Writeln('Nama : ',data.Nama);
        Writeln('Program Studi : ',data.Prodi);
        Writeln('IP : ',data.IP:2:2);
    End.
```



SET



- Tipe data set adalah tipe data yang terdiri dari kelompok data yang sama.
- Tipe data ini sedikit mirip dengan tipe data enumeration, dimana kita bisa menentukan sendiri isi dari tipe data set.
- Secara khusus, tipe data set ini sangat mirip dengan konsep **himpunan** di dalam matematika.
- Contoh deklarasi set yang valid:

type

```
Days = (mon, tue, wed, thu, fri, sat, sun);  
Letters = set of char;  
DaySet = set of days;  
Alphabets = set of 'A' .. 'Z';  
studentAge = set of 13..20;
```

var

```
himpunan1: set of 1..9;  
himpunan2: set of 99..999;  
himpunan3: set of 'a'..'z';
```

Untuk mengisi anggota himpunan, dilakukan dengan menggunakan tanda kurung siku.

$S1 := ['a', 'b', 'c'];$

$S2 := ['c', 'd', 'e'];$

Operator	Deskripsi	Hasil
+	Union 2 Set	$['a', 'b', 'c', 'd', 'e']$
-	Difference 2 Set	$['a', 'b']$
*	Intersection 2 Set	$['c']$
><	Symmetric difference 2 Set	$['a', 'b', 'd', 'e']$
=	Memeriksa equality 2 Set	False
<>	Memeriksa non-equality 2 Set	True
<=	Contains	False
Include	Include sebuah elemen pada set	Include ($S1, 'd'$) -> $['a', 'b', 'c', 'd']$
Exclude	Exclude sebuah elemen pada set	Exclude ($S2, 'd'$) -> $['c', 'e']$
In	Memeriksa keberadaan sebuah elemen pada sebuah set	'e' in $S2$ -> True


```
program tipe_set;
uses crt;
var
    himpunan1: set of char;
begin
    clrscr;
    himpunan1 := ['a', 'i', 'u', 'e', 'o'];

    if 'e' in himpunan1 then
        writeln('e ada di himpunan1');

    readln;
end.
```

```
program tipe_set;
uses crt;
var
    i:char;
    himpunan1: set of char;
    himpunan2: set of char;
    himpunan3: set of char;
begin
    clrscr;
    himpunan1 := ['a', 'i', 'u', 'e', 'o'];
    himpunan2 := ['x', 'y', 'z'];

    himpunan3 := himpunan1 + himpunan2;

    for i in himpunan3 do
        writeln(i);

    readln;
end.
```



FILE



- File umumnya didefinisikan sebagai kumpulan item/komponen/elemen yang saling berkaitan, yang disimpan di disk atau media penyimpanan eksternal lainnya dan disusun secara berurutan seperti yang ditunjukkan di bawah ini:

Item-1	Item-2	Item-3	Item-4	...	EOF
--------	--------	--------	--------	-----	-----

- File diakses untuk melakukan salah satu dari operasi berikut:
 - Membaca dari file (input)
 - Menulis ke file (output)

- Pascal standar menyediakan dua jenis file yaitu file teks dan file non teks .
- Pada kedua jenis file tersebut, seluruh elemennya adalah karakter.
- Perbedaan antara file karakter biasa dan file teks adalah bahwa file teks dibagi menjadi beberapa baris, masing-masing diakhiri oleh penanda akhir baris khusus, yang secara otomatis dimasukkan oleh sistem.
- File teks terdiri dari baris-baris karakter yang berurutan dimana setiap baris dipisahkan oleh tanda *end-of-line* (EOLN). File diakhiri dengan tanda *end-of-file* (EOF).
- Contoh:

This is a text file. (EOLN)
Each line is composed of successive characters. (EOLN)
Lines are separated by end-of-line marks. (EOLN)
The file is terminated by an end-of-file mark. (EOLN)
(EOF)

Type

```
rfile = file of real;  
ifile = file of integer;  
bfile = file of boolean;  
datafile = file of record  
arrfile = file of array[1..4] of integer;
```

Deklarasi File Non Teks
type-identifier = FILE OF component-type;

var

```
marks: arrfile;  
studentdata: datafile;  
rainfalldata: rfile;  
tempdata: ifile;  
choices: bfile;  
myfile: text;
```

Deklarasi File Teks
file-name : text;

Deklarasi file teks

```
program exText;  
var  
    filename: string;  
    myfile: text;
```

Membuka file
untuk ditulis

```
begin  
    writeln('Enter the file name: ');  
    readln(filename);
```

Menetapkan nama
ke file

```
    assign(myfile, filename);  
    rewrite(myfile);
```

```
    writeln(myfile, 'Note to Students: ');  
    writeln(myfile, 'For details information on Pascal Programming');  
    writeln(myfile, 'Read: Pascal Programming Tutorial');  
    writeln('Completed writing');
```

```
    close(myfile);  
end.
```

Menutup file

Output:

```
Enter the file name:  
coba.txt  
Completed writing
```

```
program exReadText;
uses crt;
var
    myfile: text;
    info: string;

begin
    clrscr;
    (* let us read from this file *)
    assign(myfile, 'coba.txt');
    reset(myfile);
    while not eof(myfile) do
        begin
            readln(myfile, info);
            writeln(info);
        end;
    close(myfile);
end.
```

Membuka file
untuk dibaca

Output:

Note to Students:
For details information on Pascal Programming
Read: Pascal Programming Tutorial

Pada versi pascal tertentu yang mendukung tipe data STRING, memungkinkan untuk membaca file teks per baris kalimat pada file tersebut. Setiap STRING memiliki panjang maks 80 karakter.

Append sebuah file berarti menulis ke file yang sudah ada dan sudah memiliki beberapa data tanpa menimpa data yang sudah ada tersebut.

```
program exReadText;  
uses crt;  
var  
    myfile: text;  
  
begin  
    assign(myfile, 'coba.txt');  
    append(myfile);  
  
    writeln(myfile, 'Contact Details');  
    writeln(myfile, 'info@stis.ac.id');  
    writeln('Completed appending');  
  
    close(myfile);  
end.
```

Membuka file
dalam mode append

Terlebih dahulu buatlah sebuah file dengan nama `test.txt`. Isilah file tersebut dengan kalimat:
`This is a test text file`. Simpan file di folder yang sama dengan file program.

```
PROGRAM ReadMultipleStrings1 (OUTPUT,F);  
VAR  
F :TEXT;  
Str1,Str2,Str3 :STRING[5];  
BEGIN  
ASSIGN(F,'test.txt');  
RESET(F);  
READLN(F,Str1,Str2,Str3);  
WRITELN('Str1= ', Str1);  
WRITELN('Str2= ', Str2);  
WRITELN('Str3= ', Str3);  
CLOSE(F);  
WRITELN('Press ENTER to continue..');  
READLN  
END.
```

Apakah output yang dihasilkan program? Simpan ulang (save as) program dengan mengubah deklarasi `Str1, Str2, Str3` dengan menghapus panjang stringnya menjadi seperti berikut ini.

```
Str1,Str2,Str3 :STRING;
```

Amati dan pahami output yang dihasilkan.



POLITEKNIK STATISTIKA STIS

For Better Official Statistics

TERIMA KASIH

