

- PROC SORT
- Menggabungkan Data set (SET dan MERGE)
- Pernyataan WHERE
- Pernyataan Do-loop

# PROC SORT

```
PROC SORT DATA = SAS-data-set <option>;  
BY <DESCENDING> var1 <DESCENDING> var2 ...;
```

Mengurutkan data berdasarkan *by-variable*.

Option:

OUT=*nama-data-set* : data set keluaran dari PROC SORT.  
NODUPRECS atau NODUP : menghapus data yang sama untuk semua peubah.  
NODUPKEY : menghapus data yang sama berdasarkan *by-variable*.  
DUPOUT=*nama-data-set* : nama data set untuk menyimpan data yang duplikat.  
DATECOPY : Sort SAS data set tanpa mengganti tanggal *created* dan  
*modified*.

dll.

Lihat di Help untuk lebih lengkapnya.

## Contoh:

```
PROC SORT DATA=Class OUT=Class_sort1();
```

```
  BY UTS;
```

```
RUN;
```

```
PROC SORT DATA=Class OUT=Class_sort2;
```

```
  BY UTS UAS;
```

```
RUN;
```

# Pernyataan SET

```
DATA SAS-data-set;  
    SET data1<option> data2<option> ...;  
    pernyataan-SAS;  
RUN;
```

Atau

```
DATA SAS-data-set;  
    SET data1<option>;  
    pernyataan-SAS;  
    SET data2<option>;  
    pernyataan-SAS;  
RUN;
```

Option : keep, drop, rename, firstobs, obs, dll.

## Contoh:

```
DATA first;  
INPUT X1 X2 X3;  
CARDS;  
1 2 3  
4 5 6  
7 8 9  
;  
RUN;
```

```
DATA second;  
INPUT X1 X2 X3;  
CARDS;  
11 22 33  
44 55 66  
77 88 99  
;  
RUN;
```

```
DATA combine;  
SET first second;  
RUN;
```

```
DATA combinel;  
SET first;  
SET second;  
RUN;
```

```
DATA combine2;  
SET first;  
SET second(rename=(X1=V1 X2=V2 X3=V3));  
RUN;
```

# Pernyataan MERGE

```
DATA SAS-data-set;  
    MERGE data1(IN=nama-in1) data2(IN=nama-var2) ...;  
    <BY var1 var2 ...;>  
    <IF nama-in1 <AND (OR) nama-in2 ...;>;>  
RUN;
```

Menggabungkan dua SAS data set atau lebih.

Ada di dalam tahapan DATA.

**Contoh:**

**DATA** dataX;

**INPUT** Year VarX \$ @@;

**DATALINES**;

1991 X1 1992 X2 1993 X3 1994 X4 1995 X5

;

**RUN**;

**DATA** dataY;

**INPUT** Year VarY \$ @@;

**DATALINES**;

1991 Y1 1991 Y2 1993 Y3 1994 Y4 1995 Y5

;

**RUN**;

**DATA** combinedXY;

SET dataX;

SET dataY;

**RUN**;

**DATA** combinedXY;

**MERGE** dataX dataY;

**RUN**;

**DATA** combinedXY;

**MERGE** dataX dataY;

**BY** Year;

**RUN**;

# Pernyataan WHERE

**WHERE** *kondisi*;

Pernyataan WHERE dapat digunakan untuk menyatakan kondisi tertentu seperti pernyataan IF.

WHERE dapat digunakan pada tahapan DATA atau PROC, sedangkan IF hanya pada tahapan DATA.



**Contoh:**

```
DATA dataX;  
INPUT Year VarX $ @@;  
DATALINES;  
1991 X1 1992 X2 1993 X3 1994 X4 1995 X5  
;  
RUN;
```

```
DATA dataY;  
INPUT Year VarY $ @@;  
DATALINES;  
1991 Y1 1991 Y2 1993 Y3 1994 Y4 1995 Y5  
;  
RUN;
```

```
PROC PRINT DATA=combinedXY;  
    WHERE Year > 1992;  
RUN;
```

```
DATA combinedXY;  
    MERGE dataX dataY;  
    BY Year;  
    WHERE Year > 1992;  
RUN;
```

# Pernyataan DO-loop

```
DATA SAS-data-set;  
    pernyataan-SAS;  
    DO indexvar=spesifikasi1 <,spesifikasi2, ...>;  
        pernyataan-SAS;  
    END;  
RUN;
```

```
DATA SAS-data-set;  
    pernyataan-SAS;  
    DO indexvar=start TO stop <BY increment>;  
        pernyataan-SAS;  
    END;  
RUN;
```

# Pernyataan DO Iteratif

```
do Month='JAN', 'FEB', 'MAR';
```

3 times.

```
do Fib=1, 2, 3, 5, 8, 13, 21;
```

7 times.

```
do i=Var1, Var2, Var3;
```

3 times.

```
do j=BeginDate to Today() by 7;
```

Unknown. The number of iterations depends

on the values of **BeginDate** and **Today()**.

```
do k=Test1-Test50;
```

1 time. A single value of **k** is determined

by subtracting **Test50** from **Test1**.

# Pernyataan DO Iteratif

```
do i=1 to 12;
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13
```

Out of range

```
do j=2 to 10 by
```

```
2 4 6 8 10 12
```

Out of range

```
do k=14 to 2 by -2;
```

```
14 12 10 8 6 4 2 0
```

Out of range

```
do m=3.6 to 3.8 by .05;
```

```
3.60 3.65 3.70 3.75 3.80 3.85
```

Out of range

# Pernyataan DO WHILE

- Pernyataan DO WHILE menjalankan pernyataan yang ada dalam DO loop ketika ekspresi bernilai BENAR.

```
DO WHILE (ekspresi);  
    <pernyataan-SAS>  
END;
```

*ekspresi* lebih dulu dievaluasi sebelum menjalankan pernyataan.

- Pernyataan dalam loop tidak akan dijalankan jika ekspresi bernilai SALAH.

# Pernyataan DO UNTIL

- Pernyataan DO UNTIL menjalankan pernyataan yang ada dalam DO loop hingga ekspresi bernilai BENAR.

```
DO UNTIL (ekspresi);  
    <pernyataan-SAS>  
END;
```

*ekspresi* dievaluasi diakhir setelah menjalankan pernyataan.

- Pernyataan dalam loop akan dijalankan minimal satu kali hingga ekspresi bernilai BENAR.

## Contoh:

```
DATA data1;  
Var = 1;  
rata = 7;  
Seed = 0;  
DO x = 1 TO 10;  
y = sqrt(var)*rannor(seed)+rata;  
END;  
RUN;  
  
PROC PRINT DATA=data1;  
RUN;
```

```
DATA data1;  
Var = 1;  
rata = 7;  
Seed = 0;  
DO x = 1 TO 10;  
y = sqrt(var)*rannor(seed)+rata;  
OUTPUT;  
END;  
RUN;  
  
PROC PRINT DATA=data1;  
RUN;  
  
PROC PLOT DATA=data1;  
PLOT y*x;  
PLOT y*x="*";  
RUN;
```

**Contoh:**

```
DATA data1;
var = 1;
rata = 7;
seed = 0;
x = 1.1;
DO WHILE(x <= 1);
y = sqrt(var)*rannor(seed)+rata;
x = x + 0.1;
OUTPUT;
END;
RUN;

PROC PRINT DATA=data1;
RUN;
```

```
DATA data1;
var = 1;
rata = 7;
seed = 0;
x = 1.1;
DO UNTIL(x > 1);
y = sqrt(var)*rannor(seed)+rata;
x = x + 0.1;
OUTPUT;
END;
RUN;

PROC PRINT DATA=data1;
RUN;
```



**Terima Kasih**