

- FORMAT dan PROC FORMAT,
- Pernyataan LABEL,
- Pernyataan RETAIN,
- Pernyataan IF-THEN,
- Functions

FORMAT dan PROC FORMAT

Pernyataan FORMAT

FORMAT *var1* format1 *var2* format2 < *varn* formatn>;

dapat digunakan di tahapan DATA ataupun tahapan PROC PRINT.

Beberapa contoh Format yang sudah ada di SAS:

DOLLAR<w>.<d>	: dollar12.2
DATE<w>.	: date9.
COMMA<w>.<d>	: comma12.
\$CHAR<w>.	: \$char3.

dan lain-lain.

Contoh:

DATA employee;

INPUT ID \$ gender \$ name \$ hiredate \$9. salary division \$;

CARDS;

K01	M	John	20JAN2001	1200	1
K02	M	Steve	09SEP2004	1100	1
K03	F	Rebeca	25NOV2002	1150	2

;

RUN;

PROC PRINT DATA=employee;

FORMAT salary **dollar12.2;**

RUN;

Contoh:

DATA employee;

INPUT ID \$ gender \$ name \$ hiredate DATE9. salary division \$;

CARDS;

K01	M	John	20JAN2001	1200	1
K02	M	Steve	09SEP2004	1100	1
K03	F	Rebeca	25NOV2002	1150	2

;

RUN;

PROC PRINT DATA=employee;

FORMAT hiredate **mmddy10.** salary **dollar12.2;**

RUN;

FORMAT dan PROC FORMAT

PROC FORMAT;

VALUE <\$>*namaformat* <">*val1*<"> = <">*format1*<"> ... ;

RUN;

Tanda \$ digunakan untuk peubah karakter, sedangkan peubah numerik tidak perlu.

Contoh:

PROC FORMAT;

VALUE \$jk "M" = "Male" "F" = "Female"; /*karakter*/

VALUE div 1 = "HRD" 2 = "Accounting"; /*numerik*/

RUN;

proc print data=employee;

format gender \$jk.

hiredate **mmddy10.**

salary **comma12.2**

division **div.;**

run;

LABEL

Pernyataan LABEL

```
LABEL var1 = label1 ... <var-n = label-n>;
```

Penyataan LABEL dapat ditulis pada tahapan PROC PRINT atau DATA.

```
PROC PRINT DATA = employee LABEL;
```

```
    LABEL ID = "Employee ID" gender = "Jenis Kelamin";
```

```
RUN;
```

RETAIN

Pernyataan RETAIN

RETAIN *variable(s)*;

Digunakan untuk mempertahankan nilai suatu peubah dari satu iterasi ke iterasi berikutnya. Setiap iterasi SAS akan membuat nilai peubah menjadi missing.

Contoh:

DATA salary;

SET employee;

 kumulatif = kumulatif + salary;

RUN;

DATA salary;

SET employee;

 kumulatif = 0;

 kumulatif = kumulatif + salary;

RUN;

DATA salary;

SET employee;

RETAIN kumulatif 0;

 kumulatif = kumulatif + salary;

RUN;

IF-THEN-ELSE

Pernyataan IF-THEN-ELSE

IF *kondisi* **THEN** *ekspresi*;

IF *kondisi* **THEN** *ekspresi1*;

ELSE *ekspresi2*;

- *kondisi* adalah ekspresi logika
- operator perbandingan: EQ (=), NE (<>), GT(>), LT (<), GE (>=), LE (<=)
- operator logika: AND (& atau &&) dan OR (| atau ||)

IF-THEN DO-ELSE

Pernyataan IF-THEN DO-ELSE

IF *kondisi* **THEN DO;**

ekspresi1;

 ekspresi2;

END;

IF *kondisi* **THEN DO;**

ekspresi1;

 ekspresi2;

ELSE DO;

ekspresi1;

 Ekspresi2;

END;

Contoh:

DATA nilai;

SET class;

IF nilai \geq **75** **THEN** mutu = "A";

ELSE IF nilai $<$ **75** **AND** nilai \geq **60** **THEN** mutu = "B";

ELSE mutu = "C";

run;

```
DATA nilai;  
  SET class;  
  LENGTH lulus $ 5;  
  IF nilai >= 75 THEN DO;  
    mutu = "A";  
    lulus = "Ya";  
  END;  
  ELSE IF nilai < 75 AND nilai >= 60 THEN DO;  
    mutu = "B";  
    lulus = "Ya";  
  END;  
  ELSE DO;  
    mutu = "C";  
    lulus = "Tidak";  
  END;  
RUN;
```

SAS FUNCTION

Beberapa Fungsi dalam SAS

Numeric:

INT (<i>arg</i>)	: Mengambil nilai integer saja dari argumen.
LOG (<i>arg</i>)	: Logaritma natural (ln).
LOG10 (<i>arg</i>)	: Logaritma basis 10.
MAX (<i>arg, arg, ...</i>)	: Nilai maksimal yang non-missing.
MIN (<i>arg, arg, ...</i>)	: Nilai terkecil yang non-missing.
MEAN (<i>arg, arg, ...</i>)	: Rata-rata dari nilai yang non-missing.
SUM (<i>arg, arg, ...</i>)	: Menjumlahkan nilai yang non-missing.
ROUND (<i>arg, round-off-unit</i>)	: Pembulatan ke round-off-unit terdekat.

Pernyataan *Assignment* [1]

Bentuk umum:

NamaPeubah = ekspresi_numerik;

- Ekspresi_numerik terdiri dari:
 1. Konstanta numerik
 2. Peubah numerik
 3. Operator numerik
 4. Functions: SQRT, ABS, MIN, MAX, SIN, dll

Contoh:

Rata_rata = mean(var1, var2, var3);

Beberapa Fungsi dalam SAS

Character:

- LEFT**(*arg*) : Left aligns a character expression.
- RIGHT**(*arg*) : Membuat karakter rata kanan.
- LENGTH**(*arg*) : Mengembalikan panjang suatu karakter argumen tanpa menghitung trailing blank (missing values panjangnya 1).
- SUBSTR**(*arg, position*<*length*>) : Ekstrak substring from an argument starting at 'position' for 'n' characters or until end if no 'n'.
- SCAN**(*arg, n*<*delimiter*>) : Mengambil kata ke-*n* yang dipisah oleh delimiter.
- UPCASE**(*arg*) : Mengubah karater menjadi uppercase.
- LOWCASE**(*arg*) : Mengubah karater menjadi lowercase.
- TRIM**(*arg*) : Membuang trailing blanks dari ekspresi karakter.

Pernyataan *Assignment* [3]

Bentuk umum:

NamaPeubah = ekspresi_string;

- Ekspresi_string terdiri dari:
 1. Konstanta karakter atau string
 2. Peubah string
 3. Operator string: ||
 4. Functions: TRIM, LENGTH, dsb.

Contoh:

```
DATA string_ekstrak;  
    string = "Hari Senin minggu kedua";  
    Text1 = SUBSTR(string, 1, 10);  
    text2 = string;  
    SUBSTR(text2, 6, 5) = "Jumat";  
RUN;
```

Slide Terakhir

Terima Kasih