



STK 572

Manajemen Data Statistik

Tim Dosen:

Dr. Farit Muhammad Affendi

Dr. Agus M Soleh

2016

INFORMAT - FORMAT

Dr. Agus M Soleh

agusms@apps.ipb.ac.id



INFORMAT

- Informats are typically used to read or input data from external files called flat files (text files, ASCII files, or sequential files)
- The informat instructs SAS on how to read data into SAS variables SAS informats are typically grouped into three categories:
- character, numeric, and date/time



INFORMAT

Character Informats:	\$INFORMAT <i>w.</i>
Numeric Informats:	INFORMAT <i>w.d</i>
Date/Time Informats:	INFORMAT <i>w.</i>



Ilustrasi

- Data tersimpan:

ID	Transaction Date	Transaction Amount
124325	08/10/2003	1250.03
7	08/11/2003	12500.02
114565	08/11/2003	5.11



Ilustrasi

```
filename transact 'C:\BBU FORMAT\DATA\TRANS1.DAT';

data transact;
  infile transact;
  input @1 id $6. @10 tran_date mmddy10. @25 amount ;
run;

proc print data=transact;
run;
```

Starting Column

VARIABLE

INFORMAT

Obs	id	tran_date	amount
1	124325	15927	1250.03
2	7	15928	12500.02
3	114565	15928	5.11



Ilustrasi 2

- Data tersimpan:

124325	08/10/2003	\$1,250.03
7	08/11/2003	\$12,500.02
114565	08/11/2003	5.11



Ilustrasi 2

```
filename transact 'C:\BBU FORMAT\DATA\TRANS1.DAT';

data transact;
  infile transact;
  input @1 id          $6.
        @10 tran_date mmdyy10.
        @25 amount    comma10.2
        ;
run;

proc print data=transact;
run;
```

Obs	id	tran_date	amount
1	124325	15927	1250.03
2	7	15928	12500.02
3	114565	15928	5.11



Ilustrasi 3

```
filename transact 'C:\BBU FORMAT\DATA\TRANS1.DAT';

data transact;
  infile transact;
  input @1 id          $CHAR6.
        @10 tran_date mmddy10.
        @25 amount    comma10.2
        ;
run;

proc print data=transact;
run;
```

Obs	id	tran_date	amount
1	124325	15927	1250.03
2	7	15928	12500.02
3	114565	15928	5.11



FORMAT

- Format are instructions for outputting data
- Enhance the readability of reports by formatting the data values

Salary Report						
Obs	Emp ID	Last Name	First Name	Job Code		Annual Salary
1	0031	GOLDENBERG	DESIREE	PILOT		\$50,221.62
2	0040	WILLIAMS	ARLENE M.	FLTAT		\$23,666.12
3	0071	PERRY	ROBERT A.	FLTAT		\$21,957.71
4	0082	MCGWIER-WATTS	CHRISTINA	PILOT		\$96,387.39
5	0091	SCOTT	HARVEY F.	FLTAT		\$32,278.40

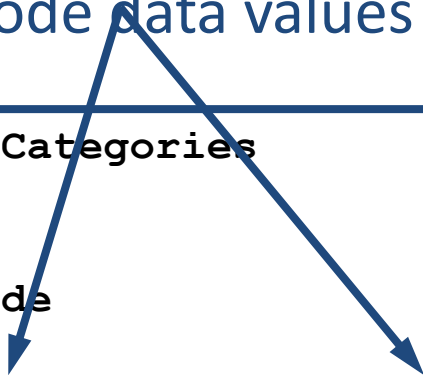


Using User-defined Formats

- Create custom formats to **recode data values** in a report.

Salary Report in Categories

Emp ID	Last Name	First Name	JobCode	Annual Salary
0031	GOLDENBERG	DESIREE	Pilot	More than 50,000
0040	WILLIAMS	ARLENE M.	Flight Attendant	Less than 25,000
0071	PERRY	ROBERT A.	Flight Attendant	Less than 25,000
0082	MCGWIER-WATTS	CHRISTINA	Pilot	More than 50,000
0091	SCOTT	HARVEY F.	Flight Attendant	25,000 to 50,000
0106	THACKER	DAVID S.	Flight Attendant	Less than 25,000
0355	BELL	THOMAS B.	Pilot	More than 50,000
0366	GLENN	MARTHA S.	Pilot	More than 50,000



Creating User-defined Formats (Permanent **Value** label assignment)

- Similar to the LABEL statement, SAS also provides the FORMAT procedure, which enables you to define custom format as a **PERMANENT** value label assignment, defined before PROC PRINT statements,.
- To create and use your own formats,
 1. use the **FORMAT procedure** to **create** the format
 2. **apply** the format to specific variable(s) by using a **FORMAT statement**.

Creating User-defined Formats

- General form of a PROC FORMAT step:

```
PROC FORMAT <options>;  
  VALUE format-name range1='label'  
                    range2='label'  
                    ... ;  
RUN;
```

Useful *Options* are

- **LIBRARY = libref** specify the libref for a SAS data library for a permanent catalog in which user-defined formats are stored.
- **FMTLIB** prints the contents of a format catalog.



How to create User-defined Formats

Rules for Format-name

- names the format you are creating
- cannot be more than 8 characters
- for character values, must have a dollar sign (\$) as the first character, a letter or underscore as the second character, and no more than 6 additional characters, numbers, and underscores
- for numeric values, must have a letter or underscore as the first character and no more than 7 additional characters, numbers, and underscores
- cannot end in a number
- cannot be the name of a SAS format
- does not end with a period in the VALUE statement.

Creating User-defined Formats

- *Labels*
 - can be up to **32,767 characters** in length
 - are typically enclosed in **quotes**, although it is not required.

Range(s)

- can be single values
- ranges of values.



Creating User-defined Formats

- Assign labels to single numbers.

```
proc format;  
  value gender 1='Female'  
              2='Male'  
              other='Miscoded';  
run;
```

Formatted
value

Numeric
format
name

Numeric data
value

Keyword



Creating User-defined Formats

- Assign labels to ranges of numbers.

```
proc format;  
  value boardfmt low-49='Below'  
                50-99='Average'  
                100-high='Above Average';  
run;
```

Keyword

Numeric data ranges



Creating User-defined Formats

- Assign labels to character values and ranges of character values.

Character
format name

```
proc format;  
  value $grade_fm 'A'='Good'  
                 'B'-'D'='Fair'  
                 'F'='Poor'  
                 'I','U'='See Instructor'  
  other='Miscoded';  
run;
```

Character
value range

Discrete
character
values

Keyword

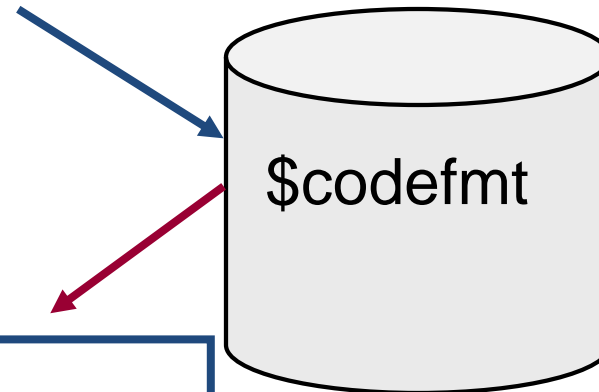
Creating User-defined Formats

Step 1: Create the format.

```
proc format;  
    value $codefmt 'FLTAT'='Flight Attendant'  
                  'PILOT'='Pilot';  
run;
```

Step 2: Apply the format.

```
proc print  
data=mylib.empdata;  
    format JobCode $codefmt.;  
run;
```



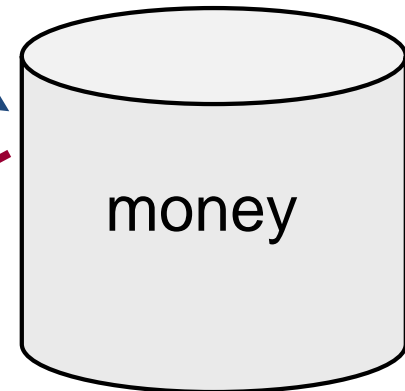
Creating User-defined Formats

Step 1: Create the format.

```
proc format;  
  value money low-<25000 ='Less than 25,000'  
              25000-50000='25,000 to 50,000'  
              50000<-high='More than 50,000';  
run;
```

Step 2: Apply the format.

```
proc print  
data=mylib.empdata;  
  format Salary money.;  
run;
```



Creating User-defined Formats

You can use multiple VALUE statements in a single PROC FORMAT step.

```
proc format;  
  value $codefmt 'FLTAT'='Flight Attendant'  
                'PILOT'='Pilot';  
  value money low-<25000 ='Less than 25,000'  
              25000-50000='25,000 to 50,000'  
              50000<-high='More than 50,000';  
run;
```



Applying User-defined Formats

```
proc print data=mylib.empdata split=' ' noobs;  
  label LastName='Last Name'  
        FirstName='First Name'  
        Salary='Annual Salary';  
  format Jobcode $codefmt. Salary money.;  
  title1 'Salary Report in Categories';  
run;
```

Salary Report in Categories

Emp ID	Last Name	First Name	JobCode	Annual Salary
0031	GOLDENBERG	DESIREE	Pilot	More than 50,000
0040	WILLIAMS	ARLENE M.	Flight Attendant	Less than 25,000
0071	PERRY	ROBERT A.	Flight Attendant	Less than 25,000
0082	MCGWIER-WATTS	CHRISTINA	Pilot	More than 50,000
0091	SCOTT	HARVEY F.	Flight Attendant	25,000 to 50,000
0106	THACKER	DAVID S.	Flight Attendant	Less than 25,000
0355	BELL	THOMAS B.	Pilot	More than 50,000
0366	GLENN	MARTHA S.	Pilot	More than 50,000

Display the list of your formats

After you create your own user-defined formats and saved in your permanent library, named 'library', you can ask SAS to list the formats you have created:

```
PROC FORMAT lib = library fmlib;  
Run;
```

Fmlib is an option in the PROC FORMAT to list your formats in the LIBRARY named : library.formats catalog.

NOTE: user-defined format is saved as a Format Catalog.
The file name is **LIBREF.formats**



A SAS Program example: Permanent User-defined Format

```
libname library 'C:\Math660\fmtfolder';  
proc format lib = library fmtlib;  
value $jobfmt    'PILOT'='Pilot'  
                'FLTAT' = 'Flight Attendant';  
value salfmt     Low-30000 = 'Low'  
                30001 - 60000 = 'Moderate'  
                60000 - High = 'High';  
  
run;  
Data emp_salary; set mylib.empdata;  
format jobcode $jobfmt.  salary salfmt.;  
run;  
proc print ;  run;
```

NOTE: The above program indicates the Libref is **LIBRARY** for format catalog.

Output from **fmtlib** option in PROC FORMAT

Numeric Variable: Salary

FORMAT NAME: SALFMT **LENGTH: 8** NUMBER OF VALUES: 3
MIN LENGTH: 1 MAX LENGTH: 40 **DEFAULT LENGTH 8** FUZZ: STD

START	END	LABEL (VER. V7 V8 26SEP2010:21:06:03),
LOW	30000	Low
30001	60000	Moderate
60000	HIGH	High

NOTE: The default Length is ALWAYS 8 for numeric variables



User Defined Format for Character Variable: Jobcode

FORMAT NAME: \$JOBfmt **LENGTH: 16** NUMBER OF VALUES: 2
MIN LENGTH: 1 MAX LENGTH: 40 **DEFAULT LENGTH 16** FUZZ: 0

START	END	LABEL (VER. V7 V8 26SEP2010:21:06:03)
-------	-----	---------------------------------------

FLTAT	FLTAT	Flight Attendant
PILOT	PILOT	Pilot

NOTE: The default Length is the length of the largest format value.
For this example, there are 16 characters for value: Flight Attendant



How does SAS handle the Format behind the scene?

- Once user-defined format is created, these formats must be stored somewhere. Therefore, the user needs to define the location to store the format catalog in SAS library. The library can be either temporary or permanent.
- Temporary library is the WORK library.
- Permanent library to store Format Catalog is defined using the Libname statement.
- The recommended permanent Library Reference Name is 'LIBRARY'.
- If the user decides to use LIBREF different from LIBRARY, then, one needs to be aware that SAS system needs to be told where to look for the specific LIBREF using a system OPTIONS statement.

Temporary storing Format Catalog

In PROC FORMAT procedure, if there is no LIBRARY defined for storing the format catalog, SAS automatically creates one in the WORK library as a temporary format catalog. The name is **WORK.Formats**

The syntax is

```
PROC FORMAT;
```

```
    VALUE format-name
```

```
        range1 = 'label1' range2='label2 .....;
```

```
    VALUE format-name
```

```
        range1 = 'label1' range2='label2 .....;
```

```
RUN;
```

Permanent Library for Format Catalog

If you want to create a permanent library for your format catalog, the syntax is:

```
LIBNAME library 'physical location' ;
```

```
PROC FORMAT LIBRARY = library;
```

```
    VALUE format-name
```

```
        range1 = 'label1' range2='label2 .....;
```

```
RUN;
```

NOTE: The name for the libref to store format catalog is called *library*. The permanent catalog is called: ***library.formats***

The libref 'LIBRARY' is recommended by SAS

NOTE: Can we use a different LIBREF from 'LIBRARY' as our permanent format catalog?

ANSWER is YES, but, tricky!

How does SAS search for the user-defined formats to be applied in your SAS program?

1. By default, SAS always search for formats in the WORK library, which is named as **WORK.FORMATS** , then, search for a permanent format library with the libref named **LIBRARY** . The permanent format catalog is named **LIBRARY.FORMATS**. Then, look for other LIBREF, which is NOT automatically done!
2. For the reason in (1), the library name for storing permanent format must be named: LIBRARY, Otherwise, you must tell SAS system to look for your specific LIBREF. This is accomplished by providing the format search sequence in SAS system OPTIONS statement, which will be discussed next.

Define format catalog search sequence for the libref differs from LIBRARY

As mentioned, SAS search format catalog based on the following order:

(1) **WORK** library, (2) the library named **LIBRARY**

For this reason, the LIBNAME statement for specifying the storage location of formats must use LIBRARY as the libref.

However, users can provide the format search sequence in the SAS program to instruct SAS to locate the library with name different from LIBRARY using the following SAS System OPTIONS statement:

OPTIONS FMTSEARCH = (libref1 libref2);

SAS will search the format catalog in the order the libref's are listed.

Example for assigning different Libref for format catalog

```
libname fmtemp2 'C:\Math660\fmt2';  
options fmtsearch = (fmtemp2);  
proc format lib = fmtemp2 fmtlib;  
value $jfmt 'PILOT'='Pilot'  
          'FLTAT' = 'Flight Attendant';  
value sfmt Low-30000 = 'Low'  
          30001 - 60000 = 'Moderate'  
          60000 - High = 'High';  
  
run;  
Data emp_salary; set mylib.empdata;  
format jobcode $jfmt. salary sfmt.; run;  
proc print data=emp_salary; run;
```

In this example,
A permanent format catalog:
fmtemp2.formats
is created.
The OPTIONS statement directs
SAS to look for
The format catalog :
fmtemp2.formats



Selesai