# Manajemen Data (*Data Wrangling*) di SAS

Outline dalam praktikum ini sebagai berikut:
1. Sorting Data
2. Subsetting dan Splitting Data
3. Merging Data

---

## Dataset

Dataset yang digunakan dalam praktikum ini dapat didownload pada link berikut:

https://drive.google.com/file/d/15g43VNur0rYh3qg31hEb6WvGmzrp4rNw/view?usp=sharing

Dataset memuat sebanyak 115 pengamatan dengan 15 peubah dengan deskripsi masing-masing peubah sebagai berikut:

- Ids: nomor identitas siswa
- bday: tanggal lahir siswa
- Rank: status siswa terdiri dari 4 jenis, freshman (1), sophomores (2), junior (3), senior (4)
- Major: jurusan siswa
- Gender: jenis kelamin siswa, perempuan (0), laki – laki (1)
- Athlete: apakah siswa seorang athlete, tidak (0), iya (1)
- Height: tinggi siswa
- Weight: Berat badan siswa
- English: nilai bahasa Inggris
- Reading: nilai dalam reading
- Math: nilai matematika
- Writing: nilai dalam writing
- State: tempat tinggal terdiri dari 2 jenis, in state dan out state
- SleepTime: durasi tidur (jam)
- StudyTime: durasi belajar (jam)

## Membangun Gugus Data SAS

Sebelum menggunakan dataset, upload dataset ke dalam penyimpanan Files (Home) SAS Studio. Selanjutnya panggil dataset ke dalam SAS system dengan pernyataan data, infile, input.

```
DATA sample1;
infile '/home/u49735076/data_sample1.txt';
input ids bday Rank Major $ Gender Athlete Weight Height English
Reading Math Writing State $ SleepTime StudyTime;
informat bday MMDDYY10.;
format bday MMDDYY10.;
run;
```

Lokasi dan nama file dalam pernyataan Infile sesuaikan dengan lokasi dan nama file pada masing – masing pengguna. Jika dalam membangun gugus data SAS ingin menggunakan penyimpanan permanen maka gunakan pernyataan **libname**.

## Sorting Data

Sorting data merupakan prosedur untuk mengubah struktur data dengan mengurutkan data berdasarkan suatu peubah atau beberapa peubah. Dalam SAS untuk mengurutkan data dapat dilakukan menggunakan perintah **PROC SORT**. Bentuk umum prosedur sort adalah:

```
PROC SORT <options>;
BY var;
RUN;
```

options dapat diisi dengan pernyataan Data diikuti nama gugus data SAS atau pernyataan Out diikuti nama gugus data baru hasil sorting. Secara *default* prosedur SORT mengurutkan data dengan urutan *ascending* (dari terkecil ke terbesar). Untuk mengurutkan data dari terbesar ke terkecil (*descending*) bisa dilakukan dengan menambahkan perintah DESCENDING di depan nama variabel.

### Contoh penggunaan prosedur Sort:

Misalkan ingin mengurutkan data berdasarkan nilai matematika dari nilai terbesar ke nilai terkecil. Syntax yang digunakan adalah:

```
proc sort data=sample1 out=sample_sorted;
by descending Math;
proc print data=sample_sorted;
run;
```

Atau misalkan ingin mengurutkan data sampel pertama berdasarkan beberapa peubah yaitu durasi tidur paling sedikit dan durasi belajar paling lama. Syntax yang digunakan adalah:

```
proc sort data=sample1 out=sample_duration;
by SleepTime descending StudyTime;
proc print data=sample_duration;
run;
```

## Subsetting dan Splitting Data

Subsetting adalah mengambil sebagian data yang memenuhi suatu kriteria yang diberikan dari keseluruhan data yang ada. Subsetting data dapat juga dianggap sebagai filtering data. Subsetting menghasilkan suatu dataset baru. Sementara splitting adalah membagi data menjadi dua atau lebih dataset. Antara subsetting dan splitting keduanya menggunakan *conditional logic*. Bentuk umum prosedur subsetting sebagai berikut:

```
DATA new_dataset_name <options>;
SET old_dataset_name <options>;
IF condition THEN OUTPUT;
RUN;
```

Bentuk di atas merupakan prosedur subsetting data dengan kriteria subset inklusi, yaitu subset dengan tetap mempertahankan data yang memenuhi kriteria yang ditentukan.

```
DATA new_dataset_name <options>;
SET old_dataset_name <options>;
IF condition THEN DELETE;
RUN;
```

Sementara bentuk di atas merupakan prosedur subsetting data dengan kriteria subset eksklusi, yaitu subset dengan menghapus data yang memenuhi kriteria yang ditentukan. Selain subsetting berdasarkan kolom (peubah), prosedur subsetting juga bisa dilakukan berdasarkan baris. Bentuk umum prosedur subsetting berdasarkan baris adalah:

```
DATA new_dataset_name <options>;
SET old_dataset_name (firstobs=i obs=j);
RUN;
```

Prosedur splitting data merupakan perluasan dari subsetting data dengan kriteria subset inklusi yang memiliki *keyword* **THEN OUTPUT**. Bentuk lain prosedur splitting adalah menggunakan prosedur **IF ELSE**.

```
DATA new_dataset1 <options> new_dataset2 <options>;
SET old_dataset_name <options>;
IF condition for new_dataset_1 THEN OUTPUT new_dataset1;
IF condition for new_dataset_2 THEN OUTPUT new_dataset2;
RUN;
```

```
DATA new_dataset1 <options> new_dataset2 <options>;
SET old_dataset_name <options>;
IF condition for new_dataset_1 THEN OUTPUT new_dataset1;
ELSE OUTPUT new_dataset2;
RUN;
```

**Contoh penggunaan prosedur Subsetting dan Splitting:**

Misalkan ingin mengambil data siswa yang berjenis kelamin perempuan. Syntax yang bisa digunakan adalah:

```
data subset;
set sample1;
if (gender=0) then output;
proc print data=subset;
run;
```

Atau misalkan ingin mengambil data nilai reading dan nilai writing siswa dimana data yang diambil adalah data dengan nilai reading lebih besar sama dengan 75 dan nilai nilai writing kurang dari sama dengan 75. Syntax yang bisa digunakan adalah:

```
data subset2 (keep= reading writing);
set sample1;
if (Reading GE 75 AND Writing LE 75) then output;
proc print data=subset2;
run;
```

Dari dataset di atas misalkan data ingin dibagi menjadi 2 data yaitu data siswa yang lulus dan data siswa yang tidak lulus. Siswa dinyatakan lulus jika nilai matematika lebih besar sama dengan 70 atau nilai bahasa Inggris lebih besar sama dengan 80, selain itu maka siswa dinyatakan tidak lulus. Data yang diambil adalah data identitas siswa, nilai matematika dan nilai bahasa Inggris. Syntax yang digunakan adalah:

```
data split_lulus (keep= ids Math English) split_tidak (KEEP= ids Math english);
set sample1;
if (Math GE 70 or English GE 80) then output split_lulus;
else output split_tidak;
proc print data= split_lulus;
title 'Daftar Siswa Lulus';
proc print data= split_tidak;
title 'Daftar Siswa Tidak Lulus';
run;
```

## Merging Data

Secara umum merging atau menggabungkan dua data ada dua bentuk, yaitu:

- Appending atau disebut stacking: menggabungkan 2 data dengan menempatkan data kedua di bawah data pertama. Digunakan ketika dua atau lebih data memiliki struktur yang sama.
- Match – merging: menggabungkan data sedemikian sehingga data satu dan data yang lainnya sesuai (*match*) berdasarkan suatu peubah identitas unik pada kedua data. Digunakan ketika antar data memiliki informasi yang relevan (informasi berbeda tetapi subject sama).

Bentuk umum prosedur appending sebagai berikut:

```
DATA new_dataset_name <options>;

SET dataset_1 <options> dataset_2 <options>;

RUN;
```

Beberapa hal yang perlu diperhatikan ketika menggabungkan data secara appending adalah:

- Jika data memiliki peubah yang berbeda maka data hasil gabungan akan menyimpan semua nama peubah dengan memberi *misiing value* pada peubah yang tidak ada di data lain
- Jika data memiliki nama peubah yang sama tetapi format, label, atau length berbeda, data gabungan akan menggunakan definisi dari data yang ditulis pertama dalam prosedur SET.
- Jika data memiliki nama peubah yang sama tetapi tipe data berbeda maka syntax tidak akan tereksekusi dan tidak ada data gabungan yang terbentuk

Bentuk umum prosedur Match – merging sebagai berikut:

```
DATA new_dataset_name <options>;

MERGE dataset_1 <options> dataset_2 <options>;

BY var;

RUN;
```

Ketika melakukan match – merging setiap data terlebih dahulu diurutkan (SORT) berdasarkan peubah yang ada dalam pernyataan BY.

### Contoh penggunaan prosedur Merging Data:

Misalkan data siswa lulus dan tidak lulus sebelumnya ingin digabungkan menjadi satu data. Syntax yang digunakan adalah:

```
data siswa_stack;
set split_lulus split_tidak;
proc print data=siswa_stack;
run;
```

Untuk contoh match – merging terlebih dahulu gunakan syntax berikut untuk mendapatkan 2 data yang memiliki satu peubah unik yaitu *ids*:

```
data sub1 (keep=ids gender athlete);
set sample1 (obs=10);
DATA sub2 (keep=ids weight height);
set sample1 (obs=10);
proc print data=sub1;
title 'data sub 1';
proc print data=sub2;
title 'data sub 2';
run;
```

Misalkan ingin menggabungkan kedua data di atas, syntax yang bisa digunakan adalah:

```
data match;
merge sub1 sub2;
by ids;
proc print data=match;
run;
```

## Latihan

1. Urutkan dataset sample di atas berdasarkan **usia** siswa dengan urutan siswa yang lebih tua berada di atas
2. Buatlah peubah baru bernama **bahasa** yang merupakan hasil rata – rata nilai bahasa Inggris, Reading dan Writing, dan peubah **keterangan** yang bernilai lulus jika nilai matematika lebih dari sama dengan 70 dan nilai bahasa lebih dari sama dengan 75, selain itu keterangan tidak lulus. Simpan data dengan peubah baru tersebut ke dalam data baru bernama **Latihan** dengan menghilangkan peubah nilai bahasa Inggris, Reading dan Writing.
3. Selanjutnya pisahkan data siswa yang lulus dan tidak lulus berdasarkan keterangan.