

MODUL PRAKTIKUM 4

EKSTRAKSI TRANSFORMASI LOAD DATA-2

MEMBACA XML DAN MEMBUAT TEXT FILE FIXED
PRAKTIKUM DATA WAREHOUSE DAN BUSINESS INTELLIGENT

DASAR TEORI

Extensible Markup Language (XML) adalah bahasa markup yang diciptakan oleh konsorsium World Wide Web (W3C). Bahasa ini berfungsi untuk menyederhanakan proses penyimpanan dan pengiriman data antarserver. Setiap server bisa memiliki sistem yang berbeda-beda. Sehingga, pertukaran data antarserver yang sistemnya tidak cocok akan memakan banyak waktu. Anda perlu mengubah format data agar cocok dengan server tujuan. Data yang diubah pun berisiko hilang.

Sebagai solusinya, XML menyimpan data dalam format teks yang sederhana. Sehingga Anda tidak perlu mengubah format data sama sekali. Bahkan format datanya pun akan tetap sama walaupun ditransfer ke server lain.

Oleh karena itu, XML juga mudah untuk diperbarui ke operating system baru atau browser baru. Bahasanya pun mudah untuk dibaca manusia, komputer, hingga teknologi pengenalan suara.

Perbedaan antara XML dengan HTML adalah:

XML	HTML
Berfungsi untuk menyimpan dan mengirimkan data.	Berfungsi untuk menampilkan data.
Bebas menggunakan tag yang Anda buat sendiri.	Harus menggunakan tag-tag tertentu untuk menjalankan fungsinya.
Dapat membedakan huruf kapital dengan huruf kecil (case sensitive).	Tidak dapat membedakan huruf kapital dengan huruf kecil (case insensitive).
Terdiri dari data struktural.	Tidak terdiri dari data struktural.
Harus ada tag penutup di dalamnya.	Tidak perlu menggunakan tag penutup.

Contoh struktur XML:

Strukturnya cukup sederhana, karena hanya terdiri dari tiga segmen, yaitu:

1. Deklarasi – Bagian ini menunjukkan versi XML yang digunakan.
2. Atribut – Bagian ini berisi keterangan objek. Contohnya seperti nama, judul, jabatan, atau sejenisnya.
3. Elemen – Bagian ini berisi tag yang mendeskripsikan objek. Strukturnya terdiri dari tiga bagian, yaitu tag pembuka, isi, dan tag penutup.

Berikut adalah ilustrasi penggunaan ketiganya:

```
<?xml version="1.0"?> --> Deklarasi
<productlisting title="Daftar Produk PT.A"> --> Atribut
  <product>
    <nama>Produk Pertama</nama> --> Elemen
    <deskripsi>Meningkatkan daya tahan tubuh</deskripsi>
    <harga>Rp. 20.000</harga>
    <ongkir>Rp. 5000</ongkir>
  </product>
</productlisting>
```

Seperti yang Anda lihat, deklarasinya menunjukkan bahwa versi yang digunakan adalah 1.0. Lalu, atribut yang digunakan dalam data product listing adalah Daftar Produk PT. A.

Data di atas terdiri dari beberapa elemen, yaitu nama, deskripsi, harga, dan ongkir. Dan setiap elemen terdiri dari tiga bagian. Contohnya <nama> (tag pembuka), Produk Pertama (isi), dan </nama> (tag penutup).

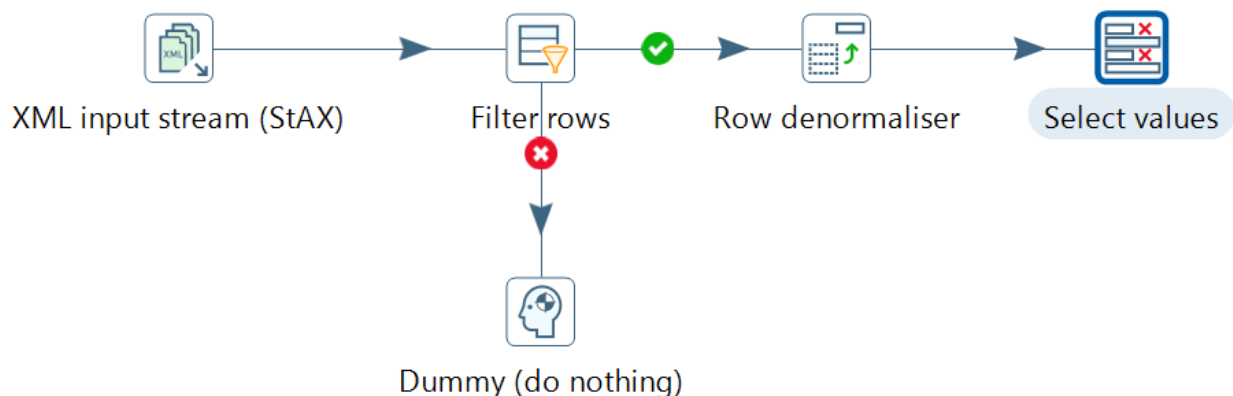
Ada beberapa contoh penerapan XML dalam pemrograman yaitu:

- XML digunakan untuk membawa data pada API seperti SOAP.
- XML digunakan untuk membentuk struktur form pada pemrograman desktop seperti delphi, lazarus, vb, gtk, dll.
- XML digunakan untuk membuat layout aplikasi android.
- XML bertugas untuk membentuk struktur proyek pada java yang menggunakan Maven.
- XML digunakan sebagai format file SVG, DOCX, dsb.
- dan masih banyak lagi contoh penerapan XML.

Untuk membaca file XML pada pentaho kita bisa menggunakan komponen XML input Stream (StAX). Pada pratikum ini kita juga belajar membuat file text dengan memanfaatkan komponen pentaho Text file output.

1) Membaca File XML

1) Buatlah desain transformasi seperti gambar dibawah ini:



2) Pada Step **XML input stream**, untuk keterangan lebih lanjut tentang step ini bisa dilihat di link :

[https://help.pentaho.com/Documentation/9.1/Products/XML_Input_Stream_\(StAX\)](https://help.pentaho.com/Documentation/9.1/Products/XML_Input_Stream_(StAX))

- Gunakan file XML yang sudah disediakan dengan nama “Provinsi Lainnya.xml”
- Browse dan pilih file tersebut pada Step ini.

```
<data>
  <row>
    <provinsi>Bali</provinsi>
    <ibukota>Denpasar</ibukota>
    <populasi>491500</populasi>
  </row>
  <row>
    <provinsi>Nusa Tenggara Barat</provinsi>
    <ibukota>Mataram</ibukota>
    <populasi>356141</populasi>
  </row>
  <row>
    <provinsi>Nusa Tenggara Timur</provinsi>
    <ibukota>Kupang</ibukota>
    <populasi>269680</populasi>
  </row>
  <row>
    <provinsi>Maluku</provinsi>
    <ibukota>Ambon</ibukota>
    <populasi>428585</populasi>
  </row>
  <row>
    <provinsi>Maluku Utara</provinsi>
    <ibukota>Sofifi</ibukota>
    <populasi>163467</populasi>
  </row>
  <row>
    <provinsi>Papua</provinsi>
    <ibukota>Jayapura</ibukota>
    <populasi>197396</populasi>
  </row>
  <row>
    <provinsi>Papua Barat</provinsi>
    <ibukota>Manokwari</ibukota>
    <populasi>207398</populasi>
  </row>
</data>
```

XML input stream (StAX)

Step name: XML input stream (StAX)
Filename: \${Internal.Entry.Current.Directory}/File Pendukung Latihan/XML/Prop... Browse...
Source is from a previous step ☐
Source field name:
Add filename to result? ☐
Skip (Elements/Attributes): 0 ⬆ ⬇ ⬇ ⬆
Limit (Elements/Attributes): 0 ⬆ ⬇ ⬇ ⬆
Default String Length: 1024 ⬆ ⬇ ⬇ ⬆
Encoding: UTF-8 ⬆ ⬇ ⬇ ⬆
Add Namespace information? ☐
Trim strings? ☒
Include filename in output? ☐ Fieldname: xml_filename
Row number in output? ☐ Fieldname: xml_row_number
XML data type (numeric) in output? ☐ Fieldname: xml_data_type_numeric
XML data type (description) in output? ☒ Fieldname: xml_data_type_description
XML location line in output? ☐ Fieldname: xml_location_line
XML location column in output? ☐ Fieldname: xml_location_column
XML element ID in output? ☒ Fieldname: xml_element_id
XML parent element ID in output? ☒ Fieldname: xml_parent_element_id
XML element level in output? ☒ Fieldname: xml_element_level
XML path in output? ☒ Fieldname: xml_path
XML parent path in output? ☒ Fieldname: xml_parent_path
XML data name in output? ☒ Fieldname: xml_data_name
XML data value in output? ☒ Fieldname: xml_data_value

Help
OK
Preview
Cancel

- Klik tombol Preview, maka akan diperlihatkan isi dari file XML yang kita gunakan.

Examine preview data
— □ ×

Rows of step: XML input stream (StAX) (81 rows)

#	xml_data_type_description	xml_element_id	xml_parent_element_id	xml_element_level	xml_pa
1	START_DOCUMENT	0	<null>	0	
2	START_ELEMENT	1	0	1	/data/
3	START_ELEMENT	2	1	2	/data/
4	START_ELEMENT	3	2	3	/data/
5	CHARACTERS	3	2	3	/data/
6	END_ELEMENT	3	2	3	/data/
7	START_ELEMENT	4	2	3	/data/
8	CHARACTERS	4	2	3	/data/
9	END_ELEMENT	4	2	3	/data/
1..	START_ELEMENT	5	2	3	/data/
1..	CHARACTERS	5	2	3	/data/
1..	END_ELEMENT	5	2	3	/data/
1..	END_ELEMENT	2	1	2	/data/
1..	START_ELEMENT	6	1	2	/data/
1..	START_ELEMENT	7	6	3	/data/
1..	CHARACTERS	7	6	3	/data/
1..	END_ELEMENT	7	6	3	/data/
1..	START_ELEMENT	8	6	3	/data/
1..	CHARACTERS	8	6	3	/data/
2..	END_ELEMENT	8	6	3	/data/
2..	START_ELEMENT	9	6	3	/data/
2..	CHARACTERS	9	6	3	/data/
2..	END_ELEMENT	9	6	3	/data/
2..	END_ELEMENT	6	1	2	/data/
2..	START_ELEMENT	10	1	2	/data/
2..	START_ELEMENT	11	10	3	/data/

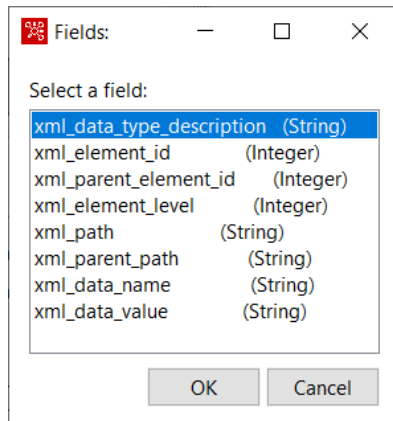
< >

Close
Show Log

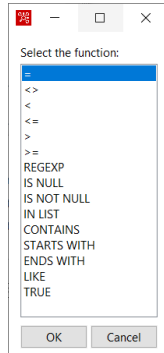
Pada kolom xml_data_type_description bisa dilihat ada bermacam-macam elemen. Nah untuk praktikum ini, kita akan memfilter yang CHARACTERS nya saja.

3) Pada Step **Filter rows**, untuk keterangan lebih lanjut tentang step ini bisa dilihat di link : <https://wiki.pentaho.com/display/EAI/Filter+rows>

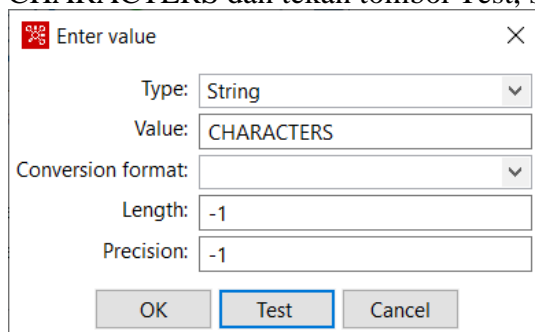
- Pada Filter rows terdapat beberapa field yang kita lihat pada gambar.
- Pada kotak bagian paling kiri yang lebih panjang (<field>), pilih xml_data_type_description.



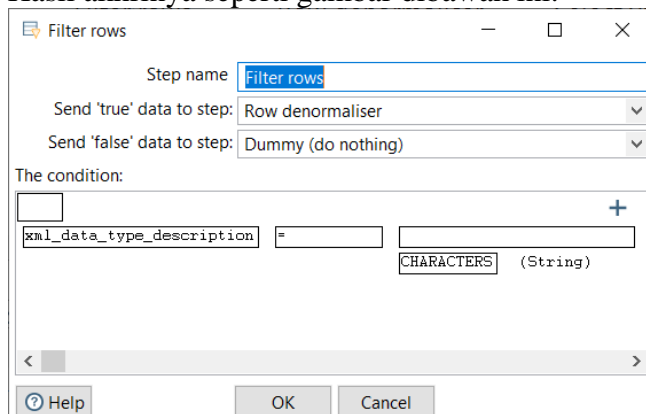
- Pada kotak bagian tengah pilih “=”.



- Pada kotak bagian paling kanan yang dibawah(<value>), inputkan pada field value: CHARACTERS dan tekan tombol Test, setelah berhasil tekan tombol OK.



- Hasil akhirnya seperti gambar dibawah ini:



- 4) Pada Step **Dummy**, untuk keterangan lebih lanjut tentang step ini bisa dilihat di link : [https://wiki.pentaho.com/display/EAI/Dummy+\(do+nothing\)](https://wiki.pentaho.com/display/EAI/Dummy+(do+nothing)) Step Dummy tidak melakukan apapun. Fungsi utamanya adalah menjadi placeholder untuk tujuan pengujian. Misalnya, untuk melakukan transformasi, Anda memerlukan setidaknya dua langkah yang terhubung satu sama lain. Jika Anda ingin menguji langkah input file, Anda dapat menghubungkannya ke Dummy step.
- 5) Pada Step **Row denormaliser**, digunakan untuk meningkatkan performa dengan cara mengurangi data yang sama atau mengelompokkan data. Fungsi lainnya untuk mengkonversi tipe data seperti yang kita lakukan pada praktikum ini. Untuk keterangan lebih lanjut tentang step ini bisa dilihat di link: https://help.pentaho.com/Documentation/8.2/Products/Data_Integration/Transformation_Scripting/Row_Denormaliser

Row denormaliser

Step name: Row denormaliser

The key field: xml_data_name

The fields that make up the grouping:

#	Group field
1	xml_parent_element_id

Get Fields

Target fields:

#	Target fieldname	Value fieldname	Key value	Type	Format
1	Provinsi	xml_data_value	provinsi	String	
2	Ibukota	xml_data_value	ibukota	String	
3	Populasi	xml_data_value	populasi	Integer	

Get lookup fields

Help OK Cancel

- The key Field yang digunakan adalah menurut xml_data_name karena berisi nama-nama dari data. Contohnya: Sumatera Barat, nama datanya adalah provinsi. Padang, nama datanya adalah ibukota.
- Group field nya adalah xml_parent_element_id karena masing-masing field berada pada parent yang berbeda.
- Target fieldname isi dengan nama data yang baru yaitu Provinsi, Ibukota dan Populasi. Pada key valuenya kita inputkan provinsi, ibukota, dan populasi yang ada pada file xml kita.

6) Pada Step **Select values**,

Select values

Step name:

Select & Alter Remove Meta-data

Fields:

#	Fieldname	Rename to	Length	Precision
1	Provinsi			
2	Ibukota			
3	Populasi			

Get fields to select
Edit Mapping

Include unspecified fields, ordered ☐

Help OK Cancel

7) Run Transformation, pada saat kita selesai Run transformasi maka muncul bagian Execution Result. Pilih bagian tab **Preview data** untuk melihat bagaimana data yang masuk atau mengalir kepada setiap Step di transformasi ini.

a. XML input stream

Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data

First rows Last rows Off

#	xml_data_type_description	xml_element_id	xml_parent_element_id	xml_element_level	xml_path	xml_parent_path	xml_data_name	xml_data_value
1	START_DOCUMENT	0	<null>	0		<null>		<null>
2	START_ELEMENT	1	0	1	/data		data	<null>
3	START_ELEMENT	2	1	2	/data/row	/data	row	<null>
4	START_ELEMENT	3	2	3	/data/row/provinsi	/data/row	provinsi	<null>
5	CHARACTERS	3	2	3	/data/row/provinsi	/data/row	provinsi	Bali
6	END_ELEMENT	3	2	3	/data/row/provinsi	/data/row	provinsi	<null>
7	START_ELEMENT	4	2	3	/data/row/ibukota	/data/row	ibukota	<null>
8	CHARACTERS	4	2	3	/data/row/ibukota	/data/row	ibukota	Denpasar
9	END_ELEMENT	4	2	3	/data/row/ibukota	/data/row	ibukota	<null>
1..	START_ELEMENT	5	2	3	/data/row/populasi	/data/row	populasi	<null>
1..	CHARACTERS	5	2	3	/data/row/populasi	/data/row	populasi	491500
1..	END_ELEMENT	5	2	3	/data/row/populasi	/data/row	populasi	<null>
1..	END_ELEMENT	2	1	2	/data/row	/data	row	<null>
1..	START_ELEMENT	6	1	2	/data/row	/data	row	<null>
1..	START_ELEMENT	7	6	3	/data/row/provinsi	/data/row	provinsi	<null>
1..	CHARACTERS	7	6	3	/data/row/provinsi	/data/row	provinsi	Nusa Tenggara Barat
1..	END_ELEMENT	7	6	3	/data/row/provinsi	/data/row	provinsi	<null>
1..	START_ELEMENT	8	6	3	/data/row/ibukota	/data/row	ibukota	<null>
1..	CHARACTERS	8	6	3	/data/row/ibukota	/data/row	ibukota	Mataram
2..	END_ELEMENT	8	6	3	/data/row/ibukota	/data/row	ibukota	<null>

b. Filter rows, masih sama dengan data di XML input stream.

c. Row denormaliser, data yang masuk ke step ini datanya sudah di filter pada step Filter rows dan dengan penambahan 3 field baru.

Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data

First rows Last rows Off

#	xml_data_type_description	xml_element_id	xml_parent_element_id	xml_element_level	xml_path	xml_parent_path	Provinsi	Ibukota	Populasi
1	CHARACTERS	5	2	3	/data/row/populasi	/data/row	Bali	Denpasar	491500
2	CHARACTERS	9	6	3	/data/row/populasi	/data/row	Nusa Tenggara Barat	Mataram	356141
3	CHARACTERS	13	10	3	/data/row/populasi	/data/row	Nusa Tenggara Timur	Kupang	269680
4	CHARACTERS	17	14	3	/data/row/populasi	/data/row	Maluku	Ambon	428585
5	CHARACTERS	21	18	3	/data/row/populasi	/data/row	Maluku Utara	Sofifi	163467
6	CHARACTERS	25	22	3	/data/row/populasi	/data/row	Papua	Jayapura	197396
7	CHARACTERS	29	26	3	/data/row/populasi	/data/row	Papua Barat	Manokwari	207398

- d. Select values, data berhasil di filter dan diubah formatnya dari XML dengan Row denormaliser ke bentuk yang lebih mudah dibaca.

Execution Results

Logging Execution History Step Metrics Perfo			
● First rows ○ Last rows ○ Off			
#	Provinsi	Ibukota	Populasi
1	Bali	Denpasar	491500
2	Nusa Tenggara Barat	Mataram	356141
3	Nusa Tenggara Timur	Kupang	269680
4	Maluku	Ambon	428585
5	Maluku Utara	Sofifi	163467
6	Papua	Jayapura	197396
7	Papua Barat	Manokwari	207398

TUGAS:

JELASKAN KEMBALI PROSES YANG TERJADI PADA ROW DENORMALISER!

2) Menulis File Text Fixed 1

Menulis ke file dalam format fixed width. Pada praktikum kali ini, file yang ditulis harus memiliki header.

- 1) Pada Step **Generate rows**, untuk keterangan lebih lanjut tentang step ini bisa dilihat di link: <https://wiki.pentaho.com/display/EAI/Generate+Rows> . Sebagai baris data masukan, kita akan membentuk 5 baris data. Untuk membentuk 5 baris data, kita gunakan step **Generate rows**. Bawa step Generate rows yang ada di grup input ke ruang kerja, dan sunting properties-nya. Pada **Limit** isi dengan 5, pada grid **Fields**, tambahkan sebuah baris sebagai berikut: Name: Urutan, Type: Integer, dan Value: 2.

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	Urutan	Integer							2	N

Jika anda tekan tombol preview, anda akan melihat 5 baris berisi angka 2. Tutup windows properties step **Generate rows**.

- 2) Pada Step **Generate random value**, untuk keterangan lebih lanjut tentang step ini bisa dilihat di link: <https://wiki.pentaho.com/display/EAI/Generate+Random+Value> . Untuk

mengisi baris data, akan kita isi dengan bilangan acak. Bawa step **Generate random value** ke ruang kerja. Buat hop antara step **Generate Rows** dan step **Generate random value**.



Generate rows Generate random value

Sunting pada properties step **Generate random value**. Sisipkan 2 baris row metadata berikut:

Step name:

Fields:

#	Name	Type
1	Teks Acak	Random string
2	Bilangan Acak	Random integer

Buttons: ? Help, OK, Cancel

Tipe field **Bilangan Acak** adalah integer, dan kita akan mengubahnya menjadi String. Bawa step **Select values** yang ada didalam grup **Transform** ke ruang kerja, dan buat hop antara step **Generate random value** ke step **Select Values**.



- 3) Pada Step **Select values**, sunting properties pindah ke tab Meta-data. Tekan tombol kolom Fieldname, dan pada kolom Fieldname akan muncul ikon ujung panah dan pilih **Bilangan Acak**. Pada kolom Type pilih String.

Step name:

Tab: Select & Alter, Remove, Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length
1	Bilangan Acak		String	

Buttons: ? Help, OK, Cancel

Karena data akan ditulis dalam bentuk fixed width, maka semua field harus kita set ukurannya, menambahkan spasi sebagai pad, dan menentukan field akan rata kiri atau rata kanan. Untuk semua kebutuhan tersebut, kita dapat menggunakan step **String Operations**.

- 4) Pada Step **String operations**, untuk keterangan lebih lanjut tentang step ini bisa dilihat di link: https://help.pentaho.com/Documentation/9.1/Products/String_Operations. Bawa step **String Operations** ke ruang kerja. Dan buat hop anatra step **Select values** dan step **String operations**.



Generate rows Generate random value Select values String operations

Sunting properties step **String operations**. Ubah nilai kolom pada masing-masing baris seperti pada baris sebagai berikut:

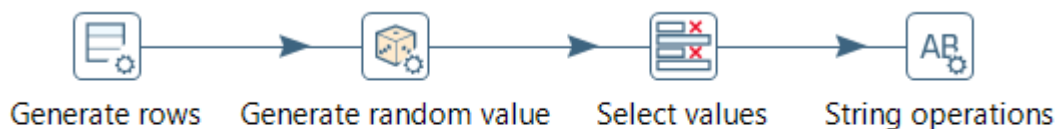
#	In stream field	Padding	Pad char	Pad Length
1	Teks Acak	Right	[spasi]	30
2	Bilangan acak	Left	[spasi]	20

Kolom lainnya biarkan apa adanya. Khusus untuk kolom **Pad char**, masukkan karakter **spasi dengan menggunakan tombol spacebar** pada keyboard.

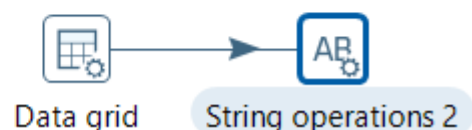
#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape
1	Teks Acak				right		30		
2	Bilangan Acak				left		20		
3									

Dengan nilai properties seperti diatas, ukuran field **Teks Acak** akan dibuat tepat 30 karakter, dan jika ukuran teks pada field tersebut kurang dari 30 karakter, maka akan dikenakan menjadi 30 karakter dengan cara menambah jumlah karakter spasi pada bagian kanan teks, sehingga teks akan rata kiri. Hal yang sama terjadi pada field **Bilangan Acak**, bedanya spasi akan ditambahkan pada bagian kiri teks, sehingga teks akan rata kanan.

- 5) Bawa step **Data Grid** dan step **String operations 2** ke ruang kerja. Buat hop dari step Data grid ke string operations 2. Step Data Grid digunakan untuk membentuk header seperti yang diharapkan.



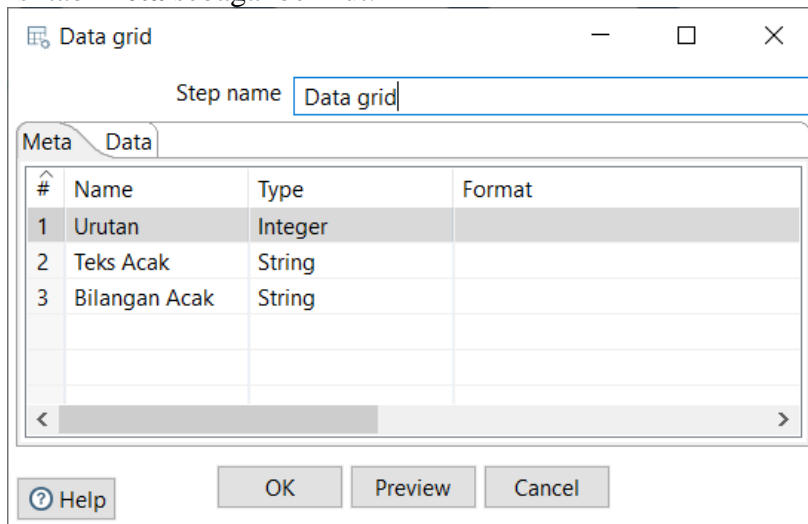
Generate rows Generate random value Select values String operations



Data grid String operations 2

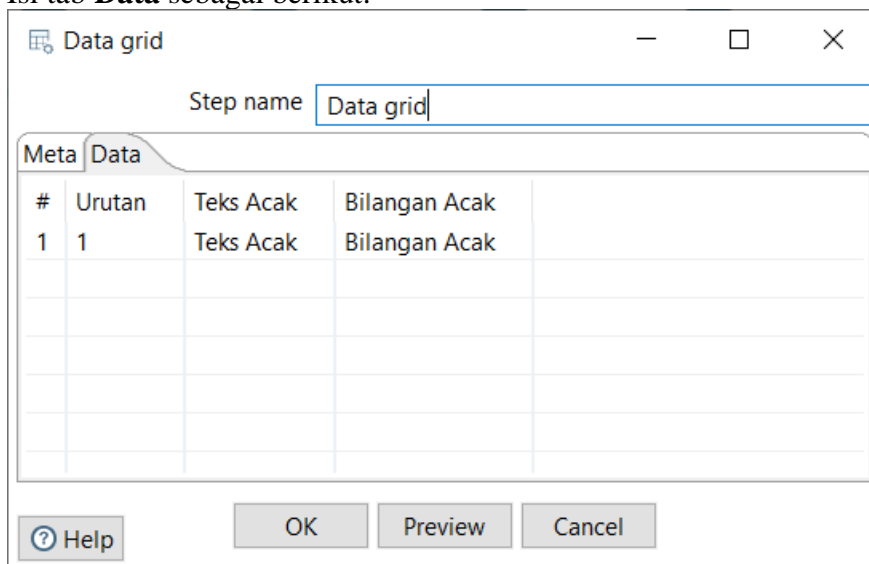
- 6) Pada Step **Data Grid**, untuk keterangan lebih lanjut tentang step ini bisa dilihat di link : <https://wiki.pentaho.com/pages/viewpage.action?pageId=23533997> . Sunting properties step **Data Grid**.

Isi tab **Meta** sebagai berikut:



#	Name	Type	Format
1	Urutan	Integer	
2	Teks Acak	String	
3	Bilangan Acak	String	

Isi tab **Data** sebagai berikut:



#	Urutan	Teks Acak	Bilangan Acak
1	1	Teks Acak	Bilangan Acak

Dikarenakan Step berjalan serentak secara paralel saat transformation dijalankan. Oleh karena itu, step yang bagian bawah menghasilkan header yang terletak di footer. Untuk hal ini kita lakukan proses Sort Rows. Kita akan mengurutkan data-data tersebut sehingga yang footer bisa jadi header.

Dimana pada **Data Grid** ditambahkan field **Urutan dengan nilai 1** mirip seperti pada **Generate rows** yang berisi nilai 2, nah urutan ini berguna untuk meletakkan data pada **Data Grid** yang dibuat agar naik keatas.

- 7) Pada Step **String operations 2**, tekan tombol **Get Fields**. Setelah field muncul, ubah nilai kolom pada masing-masing baris seperti berikut:

#	In stream field	Padding	Pad char	Pad Length
1	Teks Acak	Right	[spasi]	30
2	Bilangan acak	Left	[spasi]	20

Kolom lainnya biarkan apa adanya. Khusus untuk kolom **Pad Char**, masukkan karakter spasi dengan menggunakan tombol spacebar pada keyboard.

String operations

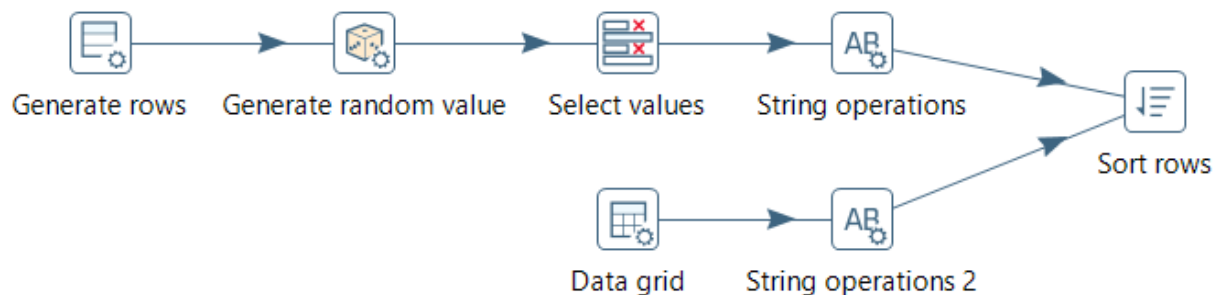
Step name: String operations 2

The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	Teks Acak		none	none	right		30	N	None	none	none
2	Bilangan Acak		none	none	left		20	N	None	none	none

Buttons: Help, OK, Get fields, Cancel

- 8) Bawa step **Sort Rows** ke ruang kerja. Buat hop untuk menghubungkan step **String operation** ke step **Sort Rows** dan step **String operation 2** ke **Sort Rows**.



Sunting properties step **Sort rows**. Pada textbox **Fieldname**, pilih field **Urutan** dan pada kolom **Ascending** pilih **Y**.

Sort rows

Step name: Sort rows

Sort directory: %%java.io.tmpdir%% [Browse...]

TMP-file prefix: out

Sort size (rows in): 1000000

Free memory threshold: []

Compress TMP Files? ☐

Only pass unique rows? ☐

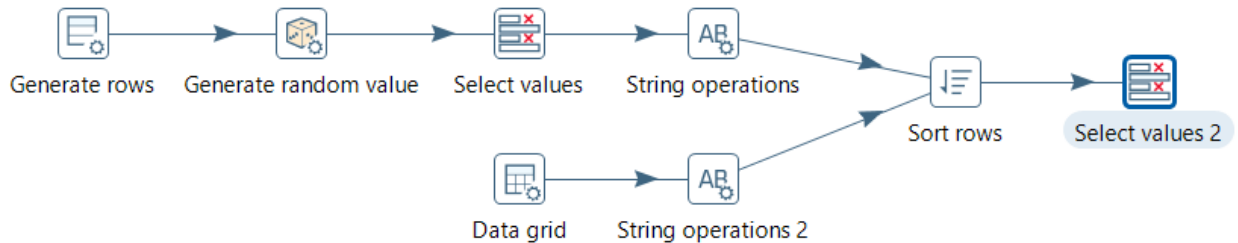
Fields:

#	Fieldname	Ascending	Case sensitive compare?	Sort
1	Urutan	Y	N	

Buttons: Help, OK, Cancel, Get Fields

Field **Urutan** hanyalah field bantuan agar data tersusun sesuai urutan pada field **Urutan** ini. Field ini tidak diperlukan untuk disimpan pada file output kita, sehingga kita butuh step **Select Values 2** untuk mengambil semua fields kecuali **Urutan**.

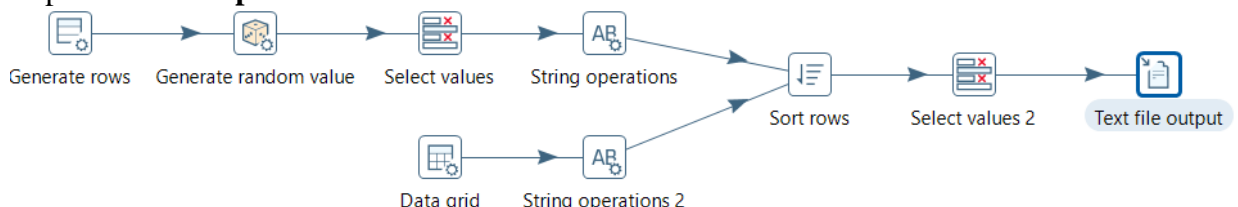
- 9) Bawa step **Select values** ke ruang kerja. Dan hubungkan hop Sort rows ke Select values 2.



Pada Step **Select values 2**, sunting properties pada **Select & Alter**, pilih fields **Teks Acak** dan **Bilangan Acak**.

#	Fieldname	Rename to	Length	Precision
1	Teks Acak			
2	Bilangan Acak			

- 10) Bawa Step **Text file output** ke ruang kerja dan hubungkan hop antara **Select values 2** ke step **Text file output**.



Pada Step **Text file output**, untuk keterangan lebih lanjut tentang step ini bisa dilihat di link:

https://help.pentaho.com/Documentation/8.2/Products/Data_Integration/Transformation_Step_Reference/Text_File_Output

Sunting properties step pada Text file output,

Pada tab **File**, tentukan lokasi dan nama file yang akan kita buat disimpan. Properties lain biarkan apa adanya, termasuk property **extension** yang berisi **txt**.

Pilih tab Content pastikan property sebagai berikut:

- **Append:** pastikan properties ini tidak dipilih.
- **Separator:** kosongkan kolom ini, kita tidak memerlukan separator.
- **Enclosure:** kosongkan kolom ini, kita tidak memerlukan karakter pengapit string.
- **Header:** Header tidak dipilih (tidak diberi tanda) karena kita membuat sendiri header kita.
- **Footer:** tidak dipilih.

Text file output

Step name: Text file output

File Content Fields

Append ☐

Separator Insert TAB

Enclosure

Force the enclosure around fields? ☐

Disable the enclosure fix? ☐

Header ☐

Footer ☐

Format: CR+LF terminated (Windows, DOS)

Compression: None

Encoding:

Right pad fields ☐

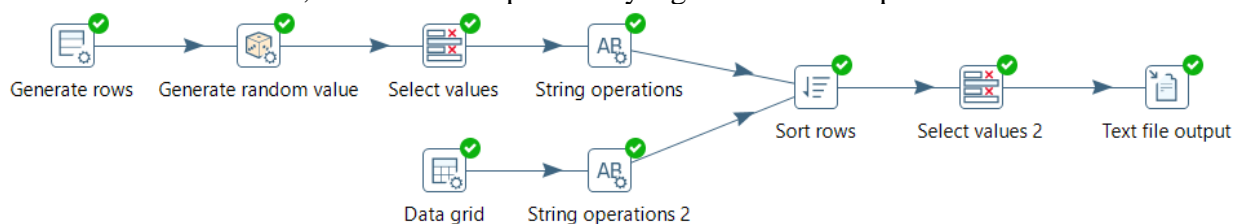
Fast data dump (no formatting) ☐

Split every ... rows:

Add Ending line of file:

Help OK Cancel

11) Jalankan transformation, dan lihat hasil pada file yang sudah kita simpan.



File txt yang kita buat:

Tulisan Acak.txt - Notepad

File Edit Format View Help

Teks Acak	Bilangan Acak
5qqmsfsp3jo58	790253656
1as0j580kogj	-1935460703
7qlfv7u0c39et	-1361271955
7mfokv697srit	-1681391143
3ui55m2dv334v	-2106699221

Pada saat kita selesai Run transformasi maka muncul bagian Execution Result. Pilih bagian tab **Preview data** untuk melihat bagaimana data yang masuk atau mengalir kepada setiap Step di transformasi ini.

a. **Step Generate rows,**

#	Urutan
1	2
2	2
3	2
4	2
5	2

b. **Generate random values,**

#	Urutan	Teks Acak	Bilangan Acak
1	2	2674554ok4vam	-1866317545
2	2	3occe6vlecmd	816037674
3	2	je7lqglhjlct	-556638308
4	2	2nhqlifaogjb7	-179674853
5	2	5j0mgv4l9tlpn	343274772

c. **String Operations,**

#	Urutan	Teks Acak	Bilangan Acak
1	2	2674554ok4vam	-1866317545
2	2	3occe6vlecmd	816037674
3	2	je7lqglhjlct	-556638308
4	2	2nhqlifaogjb7	-179674853
5	2	5j0mgv4l9tlpn	343274772

d. **Data Grid,**

#	Urutan	Teks Acak	Bilangan Acak
1	1	Teks Acak	Bilangan Acak

e. **String Operations 2,**

#	Urutan	Teks Acak	Bilangan Acak
1	1	Teks Acak	Bilangan Acak

f. **Sort rows,**

#	Urutan	Teks Acak	Bilangan Acak
1	1	Teks Acak	Bilangan Acak
2	2	2674554ok4vam	-1866317545
3	2	3occe6vlecmd	816037674
4	2	je7lqglhjlct	-556638308
5	2	2nhqlifaogjb7	-179674853
6	2	5j0mgv4l9tlpn	343274772

g. **Select Values 2,**

#	Teks Acak	Bilangan Acak
1	Teks Acak	Bilangan Acak
2	2674554ok4vam	-1866317545
3	3occe6vlecmd	816037674
4	je7lqglhjlct	-556638308
5	2nhqlifaogjb7	-179674853
6	5j0mgv4l9tlpn	343274772

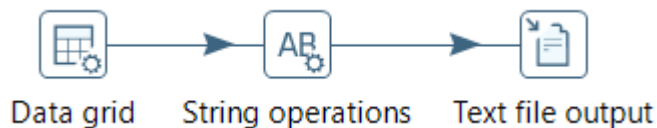
3) Menulis File Text Fixed 2

Praktikum ini tidak jauh berbeda dengan praktikum sebelumnya, yaitu membentuk file fixed width dengan header, namun dengan beberapa step yang berbeda dengan praktikum sebelumnya.

Fokus utama adalah menggabungkan Job dan Transformation. Akan dibuat 2 buah transformation, satu untuk membentuk header file dan satunya lagi untuk membentuk datanya. Kedua transformation tersebut akan dijalankan didalam sebuah job.

TRANSFORMASI 1 UNTUK HEADER

- 1) Buat sebuah transformasi baru dan simpan lah dengan nama **Header.ktr**
- 2) Bawa step **Data Grid**, step **String operations**, dan step **Text file output** ke ruang kerja. Buat hop untuk menghubungkan ketiganya.



3) Pada Step **Data Grid**.

Berikut adalah properties untuk step Data Grid pada tab **Meta**.

The screenshot shows the 'Data grid' step configuration window. The 'Step name' is 'Data grid'. The 'Meta' tab is selected, displaying a table with the following data:

#	Name	Type	Format	Length	Precision
1	Usia	String			
2	Kelompok Usia	String			

At the bottom of the window, there are buttons for 'Help', 'OK', 'Preview', and 'Cancel'.

Berikut adalah properties untuk step Data Grid pada tab **Data**.

Step name: Data grid

#	Usia	Kelompok Usia
1	Usia	Kelompok Usia

Buttons: Help, Preview, Cancel

4) Properties untuk step **String operations**.

#	In stream field	Padding	Pad char	Pad Length
1	Usia	right	[spasi]	10
2	Kelompok Usia	right	[spasi]	20

Kolom lainnya biarkan apa adanya. Khusus untuk kolom **Pad Char**, masukkan karakter spasi dengan menggunakan tombol spacebar pada keyboard.

Step name: String operations

The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length
1	Usia				right		10
2	Kelompok Usia				right		20

Buttons: Help, OK, Get fields, Cancel

5) Properties untuk step **Text field output**.

Pada tab **File**, **Filename** simpan dengan nama **Kelompok Usia**. **Extension** : txt.

Step name: Text file output

File Content Fields

Filename: \${Internal.Entry.Current.Directory}/Kelompok Usia

Pass output to servlet: ☐

Create Parent folder: ☒

Do not create file at start: ☐

Accept file name from field?: ☐

File name field: [dropdown]

Extension: txt

Include stepnr in filename?: ☐

Include partition nr in filename?: ☐

Include date in filename?: ☐

Include time in filename?: ☐

Specify Date time format: [dropdown]

Date time format: [dropdown]

Show filename(s):...

Add filenames to result: ☒

Buttons: Help, OK, Cancel

Pada tab **Content**:

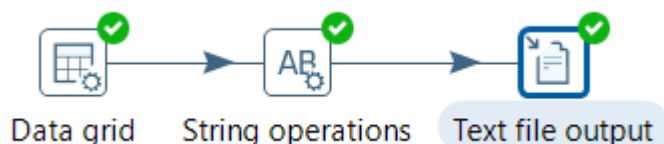
- Append: tidak dipilih, tujuannya agar setiap transformasi dijalankan, file akan dibuat dari awal, file lama akan ditimpa.
- Separator: kosongkan.
- Enclosure: kosongkan.
- Header: tidak dipilih.
- Footer: tidak dipilih.

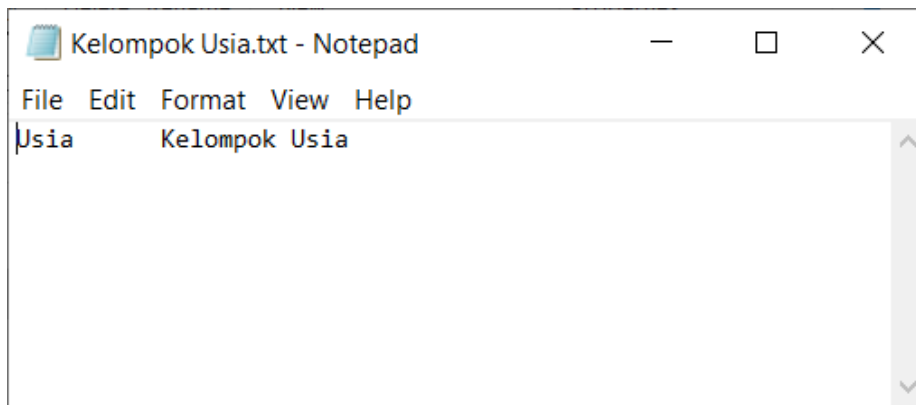
Pada tab **Fields**, cukup tekan tombol **Get Fields** dan biarkan hasil yang muncul apa adanya.

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	N
1	Usia	String		10					none	
2	Kelompok Usia	String		20					none	

At the bottom of the table, there are buttons for 'Get Fields' and 'Minimal width'.

- 6) Jalankan transformation. Jika tidak ada masalah, seharusnya terbentuk file sesuai pada step Text field output. Yang isinya seperti berikut:





TRANSFORMASI 2 UNTUK DATA

- 1) Buatlah sebuah transformasi baru, dan simpan dengan nama **Data.ktr**
- 2) Bawa Step **Generate rows** ke ruang kerja dimana kita akan membentuk 100 baris data dengan set **Limit 100**.

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1										

- 3) Bawa Step **Modified JavaScript value** ke ruang kerja dan hubungkan hop dengan **Generate rows**, untuk keterangan lebih lanjut tentang step ini bisa dilihat di link: [https://help.pentaho.com/Documentation/8.2/Products/Data Integration/Transformation Step Reference/Modified Java Script Value](https://help.pentaho.com/Documentation/8.2/Products/Data%20Integration/Transformation%20Step%20Reference/Modified%20Java%20Script%20Value) .



Sunting properties dari step ini, pada text box **Script 1**, masukkan kode JavaScript sebagai berikut:

```
var usia;
usia=Math.floor((Math.random()*80)+1);
```

Tujuan dari script ini adalah untuk membuat memberi nilai pada variabel usia dengan nilai random dari 1 hingga 80.

Pada grid Fields, sisipkan baris sebagai berikut:

- **Fieldname:** usia
- **Type:** Integer

Modified JavaScript value

Step name: Modified JavaScript value

Java script functions :

- > Transform Scripts
- > Transform Constants
- > Transform Functions
- + Input fields
- ▼ Output fields
 - Please use the 'Rep

Java script :

```
Script 1
//Script here
var usia;
usia=Math.floor(Math.random()*80)+1);
```

Position: 3, 38

Compatibility mode? ☐ Optimization level 9

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	usia		Integer			

Help OK Cancel Get variables Test script

- 4) Bawa Step **Number range** ke ruang kerja dan hubungkan dengan hop step Modified JavaScript value , untuk keterangan lebih lanjut tentang step ini bisa dilihat di link: <https://wiki.pentaho.com/display/EAI/Number+range>



Sunting properties **Number range**. Pada select box **Input field**, pilih field **usia**. Isi text box **Output field** dengan *kelompok*. Untuk panel **Ranges**, sispkan daftar jarak nilai sebagai berikut:

Number range

Step name: Number range

Input field: usia

Output field: kelompok

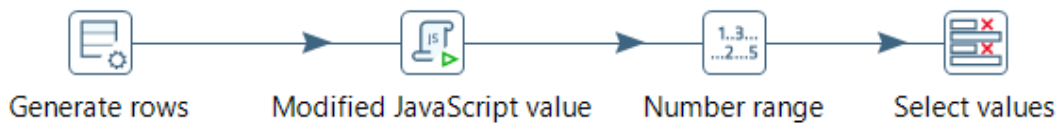
Default value(if no range) unknown

Ranges (min <= x < max):

#	Lower Bound	Upper Bound	Value
1		6	Balita
2	6	12	Kanak-kanak
3	12	17	Remaja
4	17	24	Pemuda
5	24	55	Dewasa
6	55		Tua

Help OK Cancel

- 5) Bawa Step **Select values** ke ruang kerja. Buat hop dari step **Number range** ke step **Select values**.



Pindah ke tab **Meta-data**. Sisipkan baris meta-data sebagai berikut:

Step name: Select values

Select & Alter Remove Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Prec
1	usia		String	10	
2	kelompok		String	20	

Get fields to change

Help OK Cancel

- 6) Bawa Step **String operations**, buat hop dari step **Select values** ke step **String operations**.



Sunting properties step **String operations**. Berikut properties yang harus ditetapkan:

#	In stream field	Padding	Pad char	Pad Length
1	usia	right	[spasi]	10
2	kelompok	right	[spasi]	20

Kolom lainnya biarkan apa adanya. Khusus untuk kolom **Pad Char**, masukkan karakter spasi dengan menggunakan tombol spacebar pada keyboard.

String operations

Step name: String operations

The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escap
1	usia				right		10		
2	kelompok				right		20		

Help OK Get fields Cancel

- 7) Bawa Step **Text file output** ke ruang kerja. Buat hop dari step **String operations** ke step **Text file output**.



Generate rows Modified JavaScript value Number range Select values String operations Text file output

Pada tab **File**:

- Filename: simpan pada lokasi yang sama dengan Transformasi Header dengan nama yang sama **Kelompok Usia**
- Extension: txt

Text file output

Step name: Text file output

File Content Fields

Filename: Browse...

Pass output to servlet ☐

Create Parent folder ☒

Do not create file at start ☐

Accept file name from field? ☐

File name field:

Extension:

Include stepnr in filename? ☐

Include partition nr in filename? ☐

Include date in filename? ☐

Include time in filename? ☐

Specify Date time format ☐

Date time format:

Show filename(s)...

Add filenames to result ☒

Help OK Cancel

Pada tab **Content**:

- Append: Pilih properties ini, agar jika file telah ada, maka isi file akan ditambahkan.
- Separator: kosongkan.
- Enclosure: kosongkan.
- Header: tidak dipilih.
- Footer: tidak dipilih.

Text file output

Step name: Text file output

File Content Fields

Append ☒

Separator: Insert TAB

Enclosure:

Force the enclosure around fields? ☐

Disable the enclosure fix? ☐

Header ☐

Footer ☐

Format:

Compression:

Encoding:

Right pad fields ☐

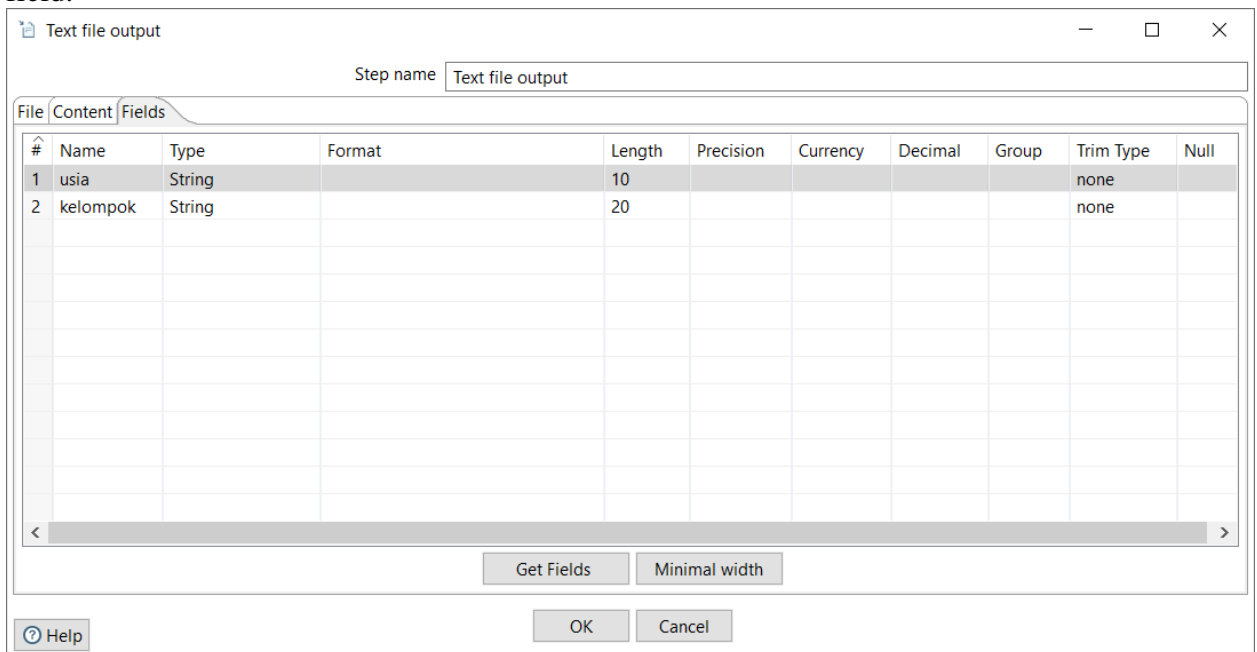
Fast data dump (no formatting) ☐

Split every ... rows:

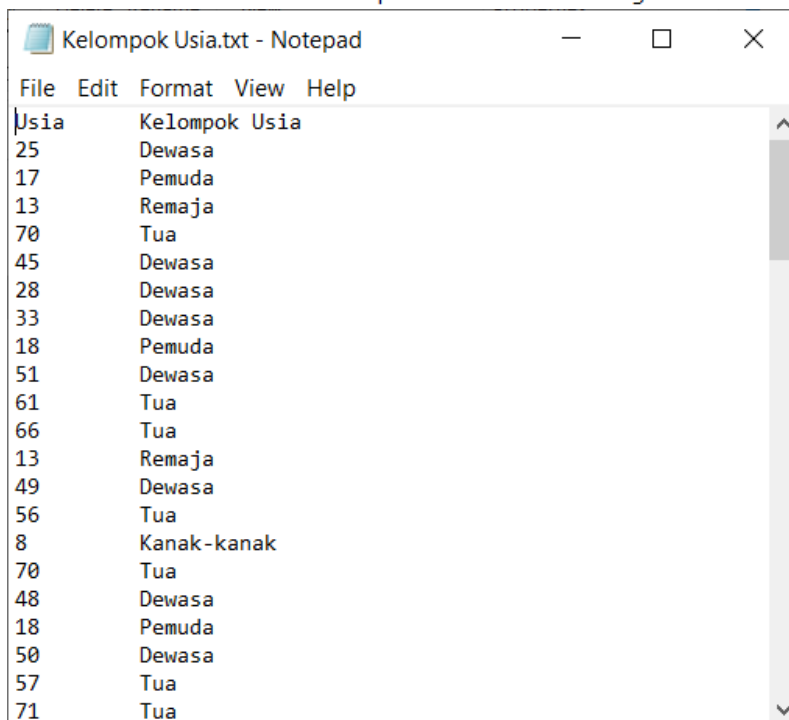
Add Ending line of file ☒

Help OK Cancel

Pada tab **Fields**. Seperti biasa, tekan tombol **Get Fields** untuk mendapatkan field secara otomatis. Jika semua field sudah didapat, pastikan kolom **Format** kosong untuk semua field.



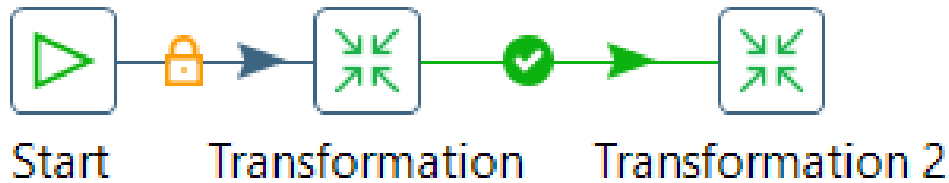
- 8) Jalankan transformation, dan jika tidak ada masalah, coba lihat isi file Kelompok Usia.txt. jika didalam sudah terbentuk data, berarti langkah pembuatan transformation sudah selesai.



Dari hasil yang kita dapatkan, bahwa hasil dari transformasi Header dan Data gabungan menjadi satu dikarenakan pada Step **Text file output di transformasi Data** bagian Content yang Append kita centang . Akan tetapi kita harus membuka satu persatu transformasi tersebut dan menjalankannya satu-satu. Oleh karena itu, kita akan menggunakan Job agar kedua transformasi bisa dijalankan secara berurutan.

JOB UNTUK MENGGABUNGKAN KEDUA TRANSFORMASI

- 1) Buatlah sebuah job. Simpan dengan nama yang diinginkan.
- 2) Bawa job entry **START** ke ruang kerja. Lalu bawa 2 buah job entry **Transformation** ke ruang kerja. Buat hop dari job entry **START** ke job entry **Transformation**. Buat hop dari job entry **Transformation** ke job entry **Transformation 2**.

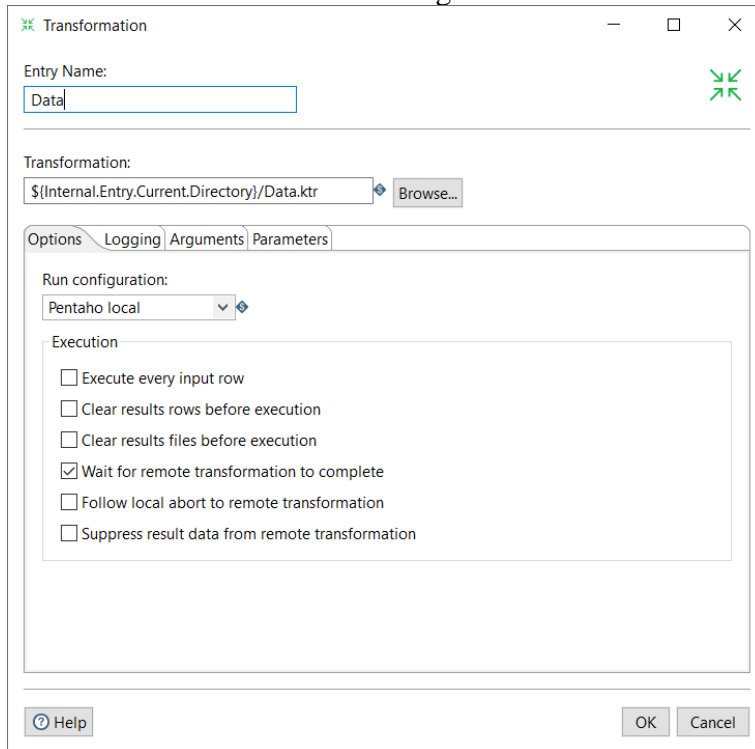


- 3) Sunting properties job entry **Transformation**. Pada text box **Entry Name**, ganti nama Transformation menjadi *Header*. Pada text box **Transformation** filename, tekan tombol **Browse Transformation** untuk membuka window **Open**. Pada window **Open**, pilih file **Header.ktr**, dan tekan tombol **Open** anda akan melihat isi dari text box Transformation filename menjadi:

The screenshot shows the 'Transformation' properties window. The 'Entry Name' field is set to 'Header'. The 'Transformation' field shows the file path 'D:/PNP/DWBI/Header.ktr'. Below this, there are tabs for 'Options', 'Logging', 'Arguments', and 'Parameters'. The 'Options' tab is selected, showing 'Run configuration' set to 'Pentaho local'. Under the 'Execution' section, several checkboxes are visible: 'Execute every input row', 'Clear results rows before execution', 'Clear results files before execution', 'Wait for remote transformation to complete' (which is checked), 'Follow local abort to remote transformation', and 'Suppress result data from remote transformation'. At the bottom, there are 'Help', 'OK', and 'Cancel' buttons.

Tutup window properties job entry Transformation. Job entry Transformation telah berubah nama menjadi job entry Header.

- 4) Sunting job entry Transformation 2. Ganti nama Transformation 2 menjadi **Data**, dan pilih file transformation **Data.ktr** sebagai **Transformation filename**.

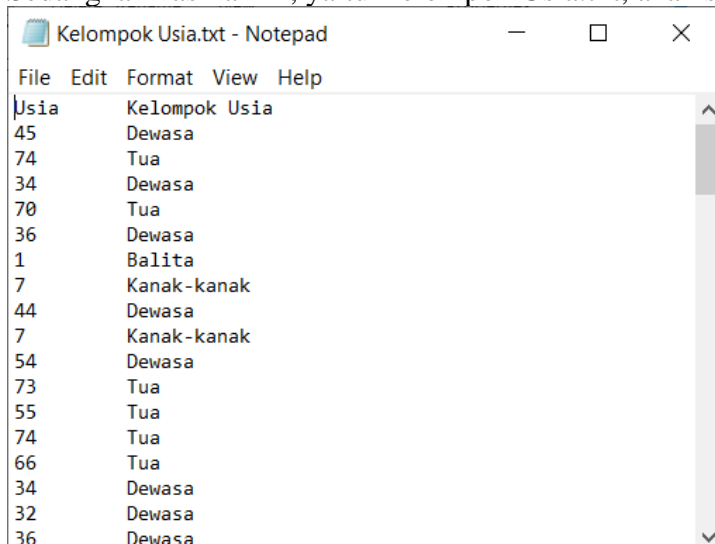


- 5) Jalankan job dengan menggunakan tombol Run this job. Tekan tombol Launch pada window Execute a job. Jika semua berjalan dengan baik, laporan Execute Results pada tab Job Metrics akan seperti gambar dibawah ini:

Execution Results

Job / Job Entry	Comment	Result	Reason	Filename	Nr	Log date
Job: Job ETL	Start of job execution		start			2021/04/07 15:...
Start	Start of job execution		start			2021/04/07 15:...
Start	Job execution finished	Success			0	2021/04/07 15:...
Header	Start of job execution		Followed unconditional link	D:/PNP/DWBI/Header.ktr		2021/04/07 15:...
Header	Job execution finished	Success		D:/PNP/DWBI/Header.ktr	1	2021/04/07 15:...
Data	Start of job execution		Followed link after success	file:///D:/PNP/DWBI/Data.ktr		2021/04/07 15:...
Data	Job execution finished	Success		file:///D:/PNP/DWBI/Data.ktr	2	2021/04/07 15:...
Job: Job ETL	Job execution finished	Success	finished		2	2021/04/07 15:...

Sedangkan hasil akhir, yaitu Kelompok Usia.txt, akan seperti gambar dibawah ini.



4) Tugas

1. Lakukan semua percobaan yang telah dijelaskan pada modul diatas dan buat laporannya.
2. Dalam laporan sesuaikan tampilan screenshot dari hasil praktikum.
3. Kerjakan perintah yang di tugaskan (**tulisan tugas yang di cetak tebal**).
4. Berikan analisa dan kesimpulan setiap percobaan yang telah dilakukan.

FORMAT LAPORAN, terdiri dari:

- A. Cover
- B. Dasar Teori
- C. Langkah dan Hasil Percobaan
- D. Analisa dan Kesimpulan Percobaan
- E. Referensi