1.4. Literasi Data Menggunakan KNIME

Bagian ini akan menjelaskan penggunaan beberapa Nodes dasar di KNIME secara praktik. Karena langsung diterapkan terhadap data, maka praktek ini sekaligus menjadi sarana literasi data karena membahas jenis data, cara memanipulasi data termasuk pemilihan kolom fitur dan baris, serta memvisualisasi data. Literasi dibagi dalam dua bagian besar: terhadap data tabular dan tekstual.

Literasi Data dalam Data Tabular

Literasi pertama adalah mengenai data yang berbentuk Tabular, yaitu yang sudah disusun dalam kolom fitur. Literasi mencakup penerapan beberapa Node terhadap data tabular, yang pada prinsipnya adalah melakukan pemilihan baris, kolom, dan memanipulasi isi di dalam sel (irisan antara baris dan kolom) tabel.

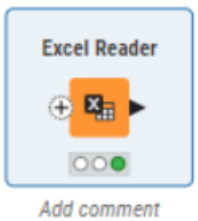
Tabel data untuk praktek KNIME diberikan dalam Tabel 1.1 yang disimpan dengan nama lulusan(1).xlsx.

Tabel 1.1. Isi file Excel "lulusan1.xlsx"

No	Mahasiswa	IPK (X1)	Lama Studi (tahun) (X2)	Lama Menganggur (bulan) (X3)
1	A	3	4,1	5,1
2	B	3,2	4	5,6
3	C	2,8	4,6	7
4	D	2,9	4	8
5	E	2,5	5	9
6	F	3,7	4,1	4,8
7	G	3,5	3,8	3,3
8	H	3,2	4,3	3,5
9	I	2,9	4,6	4
10	J	2,7	4,4	4,3
11	K	3,8	3,8	4
12	L	3,4	5	5,2

Workflow 1: Membaca Data Excel

Workflow pertama bertujuan untuk membuka data dalam file Excel dimaksud, dan melihat statistiknya. Workflow dan node yang digunakan diberikan dalam Gambar 1.32



Gambar 1.32 Node Excel Reader

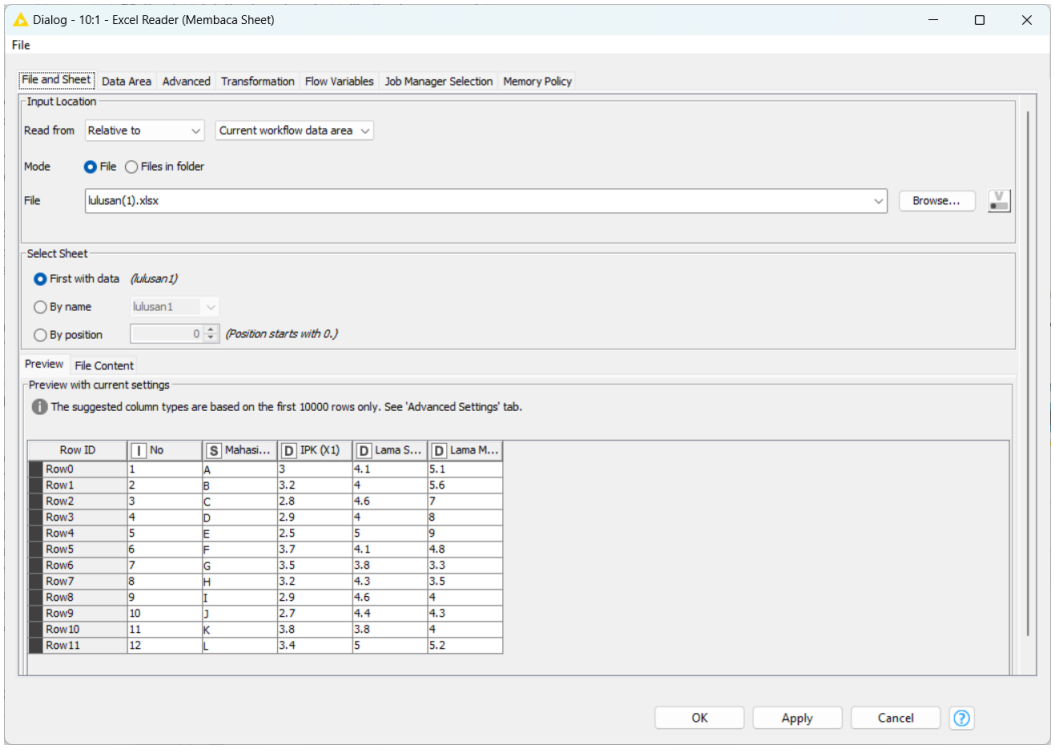
Karena ini adalah workflow pertama, maka langkah yang umumnya dilakukan untuk membuat workflow baru adalah:

1. Dari Home:
   - Di opsi "Recent", klik button "+ Create new workflow", atau Di Local space, klik button kuning bertanda plus untuk membuat workflow baru.
   - Kemudian akan muncul form dalam Gambar 1.33, isikan nama workflow yang diinginkan; lalu klik button Create

Gambar 1.33 Form Pengisian Nama Workflow

2. Kanvas kosong akan terbuka, dan siap digunakan untuk membuat workflow dalam Gambar 1.32 dengan cara menyeret-dan-menjatuhkan (drag-and-drop) node Excel Reader dari panel Node ke Kanvas.

3. Dobel klik Node Excel Reader untuk membuka form dialog setingan parameter node dimaksud, seperti dalam Gambar 1.34.



Gambar 1.34 Form Dialog Excel Reader

4. Pilih file “lulusan(1).xlsx dengan opsi:
   – Pilih Read from “Relative to” untuk memilih file yang berada di dalam folder “data”, di dalam folder workspace ini
   – Pilih Read from “Local file system” untuk file yang berada di sebuah folder di dalam sistem berkas komputer User.
   – Opsi lainya belum dijelaskan di sini dan dapat diselidiki sendiri oleh Pembaca

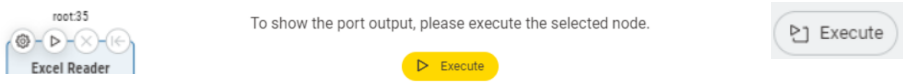
5. Pilih sheet yang akan dibaca karena sebuah file Excel dapat terdiri dari banyak sheet, dengan opsi:
   – First with data: sheet pertama yang berisi data
   – By name: jika Anda ingin memasukkan nama sheet-nya
   – By Position: nomer sheet yang dimulai dari nomer nol

6. Lihat isi sheet:
   – Preview: untuk melihat beberapa baris pertama
   – File Content: untuk melihat isi seluruh data

7. Klik OK atau Apply untuk menerapkan konfigurasi node

Mengeksekusi Node dan melihat output Node, dilakukan dengan cara yang umumnya sebagai berikut:

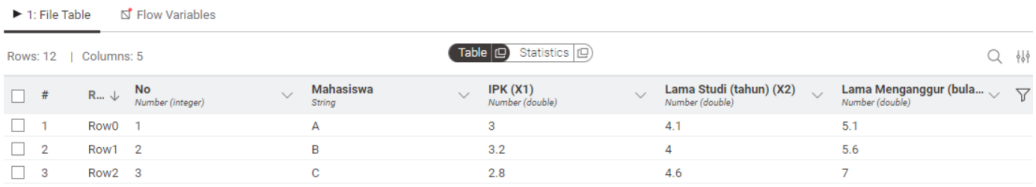
– Klik button Execution, yang berada di Node yang bersangkutan (button bersimbol *play*), button Execute di area Output, atau di balok navigasi di bawah tab workflow yang sedang terbuka. Ilustrasi masing-masing button Execute dijelaskan berturut-turut dari kiri ke kanan di Gambar 1.35.



Gambar 1.35. Ragam Opsi untuk Eksekusi Node

– Lihat hasilnya di area Output, yang terdiri dari:

   – Table, berisi konten dari data yang dibaca seperti dalam Gambar 1.36,



#	R...	No (Number (integer))	Mahasiswa (String)	IPK (X1) (Number (double))	Lama Studi (tahun) (X2) (Number (double))	Lama Menganggur (bula... (Number (double))
1	Row0	1	A	3	4.1	5.1
2	Row1	2	B	3.2	4	5.6
3	Row2	3	C	2.8	4.6	7

Gambar 1.36

   – Statistics, yang mencerminkan karakteristik data yang terdiri atas Jenis data, jumlah nilai yang hilang (missing values), nilai unik (unique values), nilai minimum dan maksimum, nilai kuartil (Q1, Q2, dan Q3), nilai tengah (mean)

dan mean multak; seperti dalam Gambar 1.37. Jenis data yang terbaca adalah Number integer dan double yang mewakili bilangan bulan dan real, serta String untuk jenis data alfabet, termasuk nomer dan tanda baca.

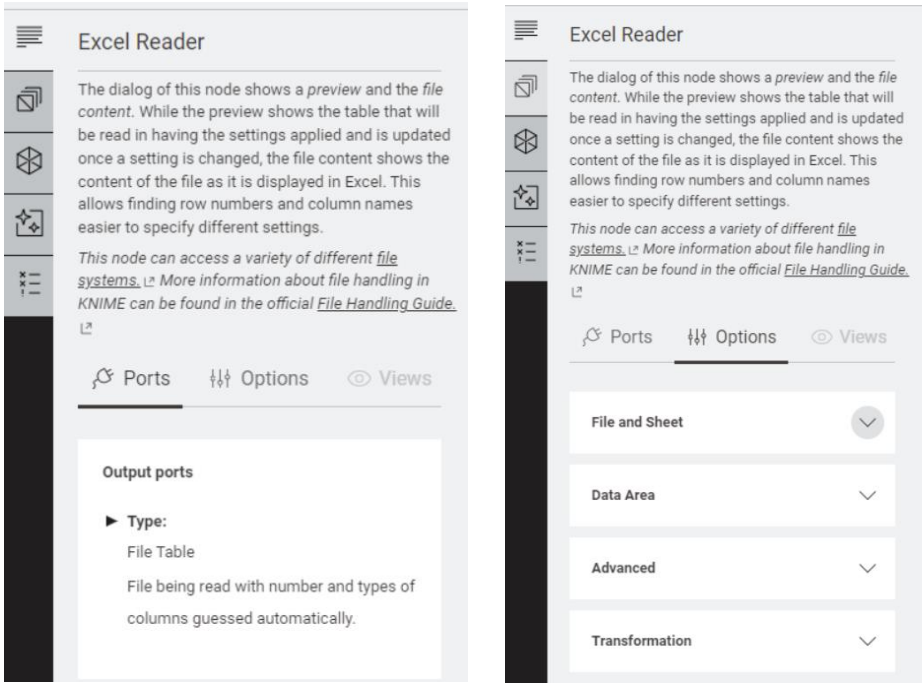
Rows: 5 | Columns: 14 — Table | Statistics

Name	Type	# Missing val...	# Unique valu...	Minimum	Maximum	25% Quantile	50% Quantile ...	75% Quantile	Mean	Mean Absolu...
No	Number (integer)	0	12	1	12	3.25	6.5	9.75	6.5	3
Mahasiswa	String	0	12	?	?	?	?	?	?	?
IPK (X1)	Number (double)	0	10	2.5	3.8	2.825	3.1	3.475	3.133	0.333
Lama Studi (t...	Number (double)	0	7	3.8	5	4	4.2	4.6	4.308	0.343
Lama Menga...	Number (double)	0	11	3.3	9	4	4.95	6.65	5.317	1.389

Gambar 1.37 Statistik dan Karakteristik Data

Melihat informasi Node, dapat dilakukan melalui panel side-bar kiri, paling atas seperti dalam Gambar 1.38. Di panel informasi ini juga terdapat informasi:

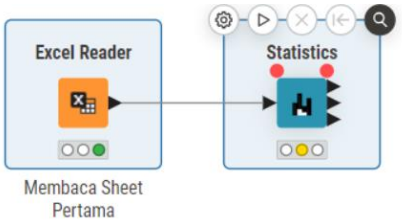
- Port termasuk jenis port yang melekat di Node (input, output dan lainnya),
- Options yang meliputi opsi yang tersedia di Node, dan
- Views bagi beberapa node yang dapat menayangkan luaran secara visual yaitu yang diakses melalui button kaca pembesar pada Node.



Gambar 1.38 Informasi Node di Panel side-bar kiri

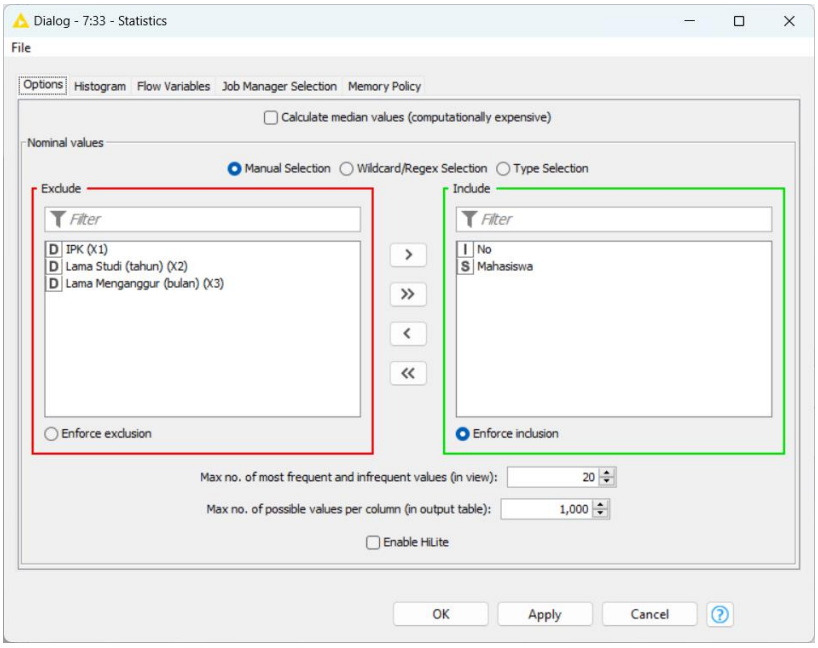
Workflow 2: Melihat Statistik Data

Selain dengan fasilitas Statistics di area output node, KNIME menyediakan node Statistics untuk melihat statistik dari data. Workflow dan node yang digunakan diberikan dalam Gambar 1.39.



Gambar 1.38 Workflow Melihat Satistik data dengan Node Statistics

Dobel klik Node Statistics untuk membuka form dialog setingan parameter node dimaksud, seperti dalam Gambar 1.39.



Gambar 1.39 Form Dialog Node Statictics

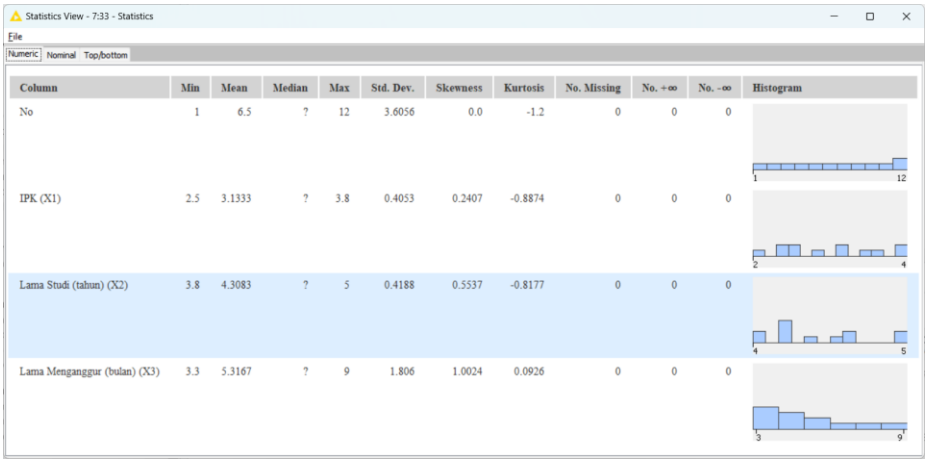
Pengaturan yang dapat diberikan antara lain:

- Include: memilih fitur dengan nilai nominal (kategorikal) yang akan diikutkan dalam perhitungan statistiknya. Contohnya: Mahasiswa dan No.
- Exclude: kebalikan dari opsi Include, yaitu memilih fitur yang bukan nominal (yaitu numerik) di box Exclude ini
- Pengaturan lainnya bisa dipraktekan sendiri oleh Pembelajar

Setelah diklik Apply atau OK, maka node Statistics ini juga bisa dieksekusi sekaligus divisualisasi hasilnya melalui button kaca pembesar di sudut kanan atas dari Node (setelah Node di mouse-hover), seperti pada Gambar 1.38.

Hasilnya diberikan pada Gambar 1.40, 1.41 dan 1.42 dengan penjelasan singkat berturut-turut sebagai berikut:

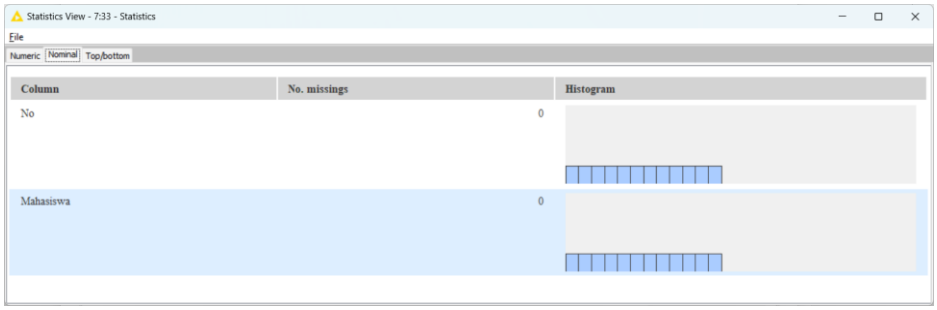
- Tab Numeric: menayangkan pemandangan statistik (*Statistics view*) dari fitur bertipe angka, termasuk nilai Minimum, maksimum, median, mean, standar deviasi, kemiringan (skewness), kurtosis, jumlah fitur dengan *missing value* (nilai yang hilang) (No. Missing), dan lainnya, serta Histogram.



Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
No	1	6.5	?	12	3.6056	0.0	-1.2	0	0	0	
IPK (X1)	2.5	3.1333	?	3.8	0.4053	0.2407	-0.8874	0	0	0	
Lama Studi (tahun) (X2)	3.8	4.3083	?	5	0.4188	0.5537	-0.8177	0	0	0	
Lama Menganggur (bulan) (X3)	3.3	5.3167	?	9	1.806	1.0024	0.0926	0	0	0	

Gambar 1.40 Tab Numeric di Statistic View

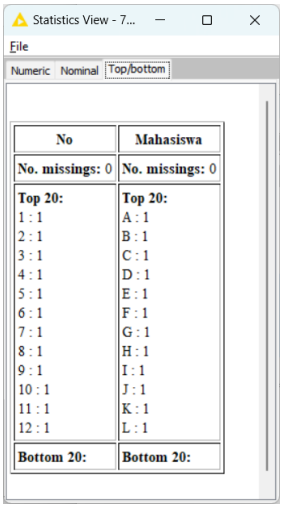
– Tab Nominal: menayangkan jumlah fitur dengan nilai yang hilang dan Histogram.



Column	No. missings	Histogram
No	0	
Mahasiswa	0	

Gambar 1.41 Tab Nominal di Statistic View

– Tab Top/Buttom: menayangkan fitur-fitur dengan frequensi tertinggi dan terendah (most frequent dan infrequent)



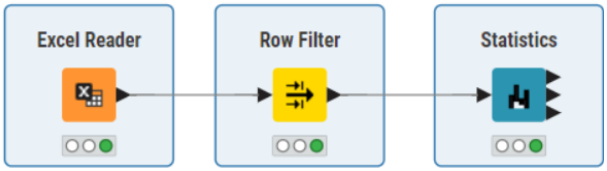
No	Mahasiswa
No. missings: 0	**No. missings:** 0
Top 20: 1 : 1 2 : 1 3 : 1 4 : 1 5 : 1 6 : 1 7 : 1 8 : 1 9 : 1 10 : 1 11 : 1 12 : 1	**Top 20:** A : 1 B : 1 C : 1 D : 1 E : 1 F : 1 G : 1 H : 1 I : 1 J : 1 K : 1 L : 1
Bottom 20:	**Bottom 20:**

Gambar 1.42 Tab Top/Bottom menayangkan Most Frequent/Infrequent Fitur

Workflow 3: Menyaring Baris Data

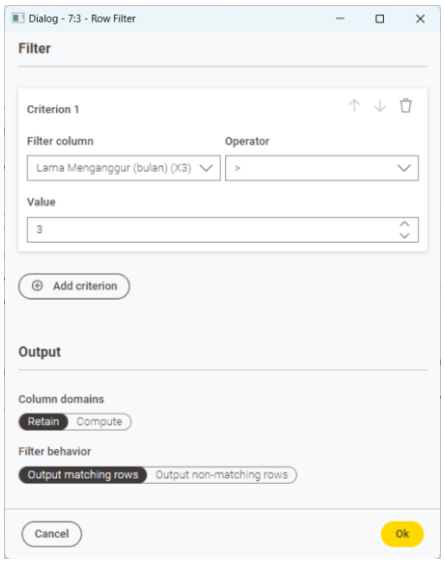
Node berikutnya adalah Row Filter yang berfungsi untuk memilih baris Data dengan workflow yang ditunjukkan dalam Gambar 1.43. User diminta menentukan kriteria pada

kolom tertentu dalam memilih baris; artinya, hanya baris dengan kolom-kolom yang memenuhi kriteria yang akan dipilih.



Gambar 1.43 Worfklow untuk Menyaring Baris data

Form Dialog node Row Filter diberikan dalam Gambar 1.44 dengan penjelasan sebagai berikut,



Gambar 1.44 Form Dialog node Row Filter

- Masukkan kriteria pada kolom tertentu yang ditentukan melalui penggunaan operator dan nilainya. Di Gambar 1.44 ditunjukkan bahwa hanya baris yang mana nilai di kolom Lama Menganggur (bulan) (X3) > 3 yang akan dipilih.
- Kriteria bisa ditambahkan dengan button Add criterion
- Opsi lainnya silakan dicoba sendiri oleh Pembelajar
- Node Statistics dalam workflow adalah opsi yang dapat dilakukan jika User juga ingin mengetahui Statistics dari Data yang sudah disaring menggunakan Node Statistics.

Sebagian hasil penyaringan dapat dilihat pada Gambar 1.45

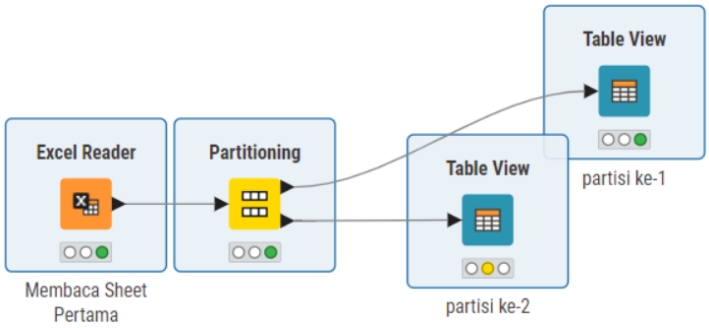
Rows: 12 | Columns: 5 Table Statistics

#	RowID	No *Number (integer)*	Mahasiswa *String*	IPK (X1) *Number (double)*	Lama Studi (tahun) (X2) *Number (double)*	Lama Menganggur (bulan) (X3) *Number (double)*
7	Row6	7	G	3.5	3.8	3.3
8	Row7	8	H	3.2	4.3	3.5
9	Row8	9	I	2.9	4.6	4
11	Row10	11	K	3.8	3.8	4
10	Row9	10	J	2.7	4.4	4.3

Gambar 1.45 Hasil Penyaringan node Row Filter pada Data

Workflow 4: Memartisi Baris Data

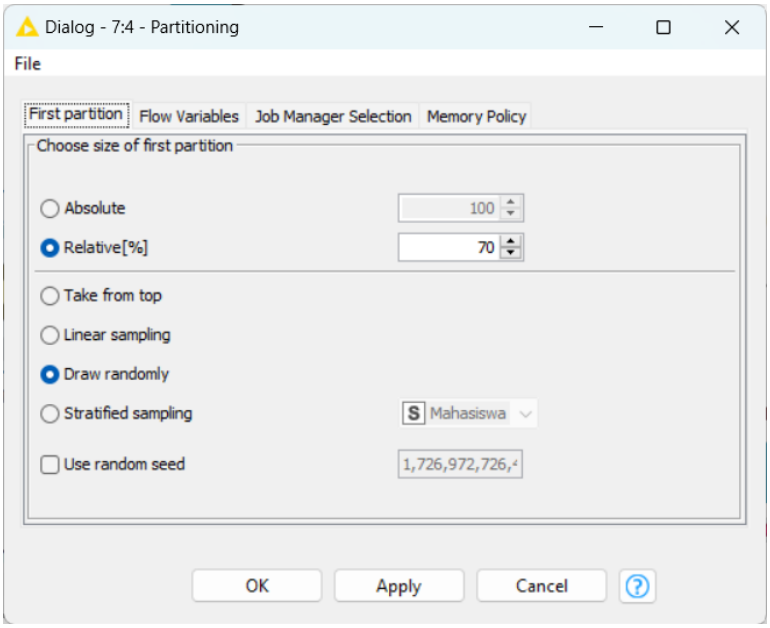
Node Partitioning berfungsi signifikan pada kegiatan Klasifikasi data, dimana baris-baris tabel data akan dibagi dua menjadi dataset latih (training dataset) dan uji (testing dataset). Workflow yang digunakan secara umum diberikan dalam Gambar 1.46



Gambar 1.46 Workflow untuk Memartisi Data

Form dialog Node Partitioning diberikan dalam Gambar 1.47, dengan penjelasaan sebagai berikut:

- Porsi pemartisian bisa ditentukan nilai nilai aboslut atau persentase. Di sini dipraktekan pemartisian sebanyak 70% untuk partisi pertama.

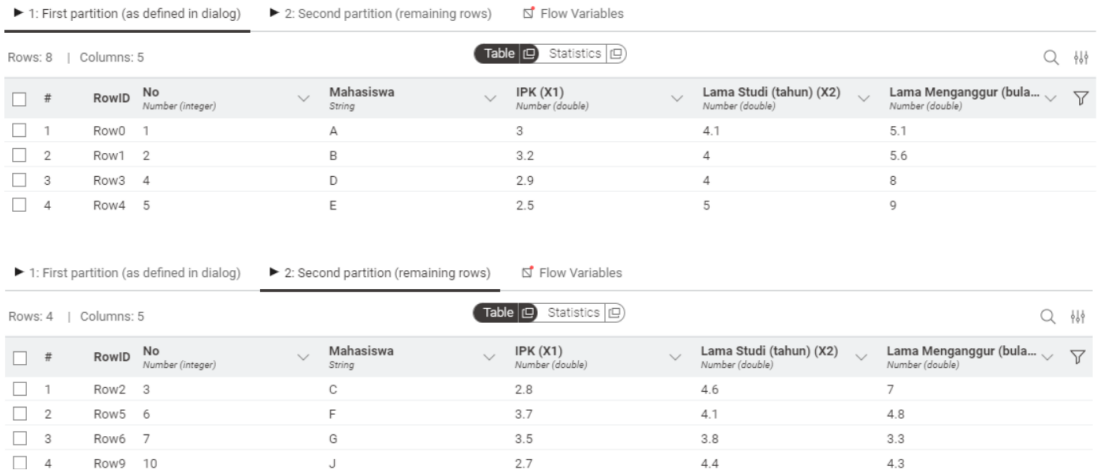


Gambar 1.47 Form Dialog Node Partitioning

- Pemilihan baris sesuai porsi dimaksud bisa dilakukan dengan beberapa opsi: mengambil dari baris paling atas (Take from top), pengambilan sampel liner (Linear sampling), pengambilan secara acak (Draw randomly), sampling berstrata (Stratified sampling)
- Opsi untuk menggunakan bibit random (random seed) bisa dipilih ketika pemilihan baris yang diinginkan adalah Stratified dan Random sampling.
  Penjelasan tiap jenis sampling dapat dipelajari sendiri oleh Pembelajar, misalnya melalui Panel kiri side-bar paling atas (klik dulu Node yang akan dilihat informasinya).

Hasil dari pemartisian dapat dilihat dengan cara:

– Table di output area, melalui dua tab: First partition dan Second partition yang masing-masing menampilkan isi partisi pertama dan kedua; lihat Gambar 1.48 atas dan bawah. Terlihat bawah di partisi pertama dan kedua terdapat 8 dan 4 baris data (70% dan 30% dari 12 baris data aslinya).

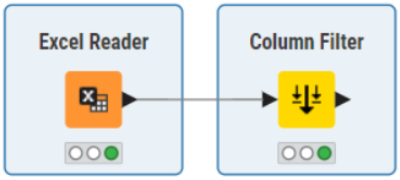


Gambar 1.48 (atas) Partisi pertama, (bawah) Partisi kedua

– Penggunaan Node Table View pada dua port output dari Node, dimana port atas mengalirkan data di First partition, dan port bawah untuk Secord partition. Hasil visualisasinya bisa dipelajari sendiri oleh Pembelajar.

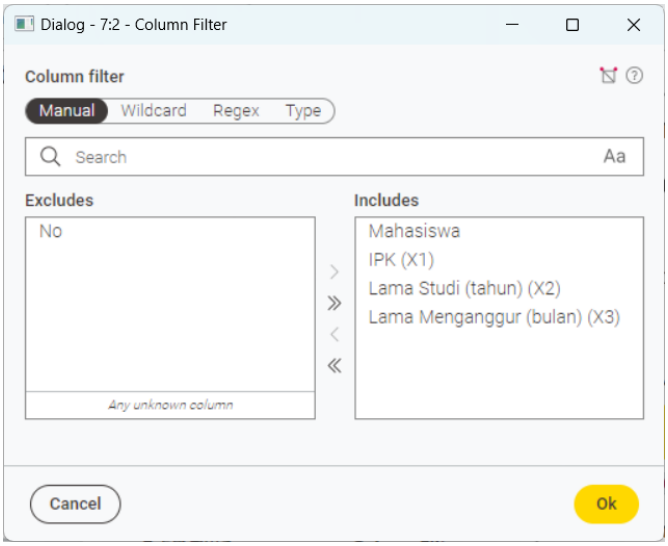
Workflow 5: Memilih Kolom Data

Node Column Filter mempraktekkan kegiatan menyeleksi fitur (Feature selection) pada DM, yaitu memilih kolom yang akan diproses di kegiatan berikutnya. Workflow pemilihan kolom diberikan di Gambar 1.49



Gambar 1.49 Workflow Pemilihan Kolom

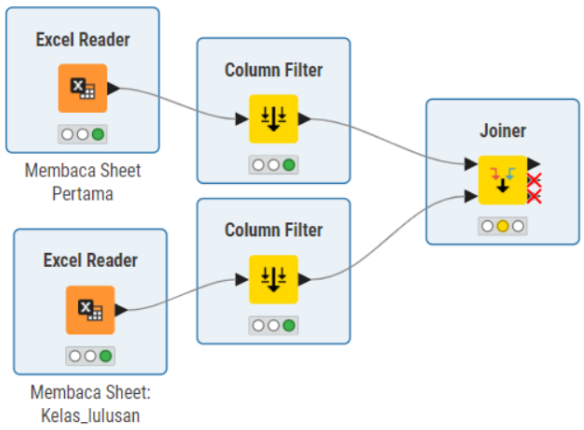
Form dialog Column Filter diberikan di Gambar 1.50, dimana terlihat fitur yang akan diikutsertakan (Includes) dan yang tidak (Excludes). Hasil dari penerapan Node ini sudah dapat dipahami oleh Pembelajar sehingga tidak dijelaskan lagi di sini.



Gambar 1.50 Form Dialog Column Filter

Workflow 6: Menggabung Tabel berdasarkan Baris

Node Joiner menggabungkan dua tabel serupa dengan *join* dalam basis data. Ini menggabungkan setiap baris dari port input atas dengan setiap baris dari port input bawah yang memiliki nilai identik di kolom yang dipilih. Baris yang tidak memiliki pasangan yang cocok juga dapat dihasilkan sebagai output. Workflow penggunaan Joiner diberikan dalam Gambar 1.51.



Gambar 1.51 Workflow penggunaan Joiner

Penjelasan workflow adalah sebagai berikut:

- Dua tabel excel akan digabungkan baris-barisnya, dimana Node Excel Reader atas memuat tabel data lulusan di Sheet pertama; Excel Reader di bawah memuat tabel data lulusan. Masing-masing tabel ini disebut Tabel atas (kiri) dan Tabel bawah (kanan)
- Jika tabel kiri terdiri atas 12 baris dengan nama mahasiswa A sampai L, maka tabel kanan terdiri atas 15 baris: yaitu sama dengan tabel kiri ditambah 3 data mahasiswa lain yaitu M, N dan O, seperti Gambar 1.52

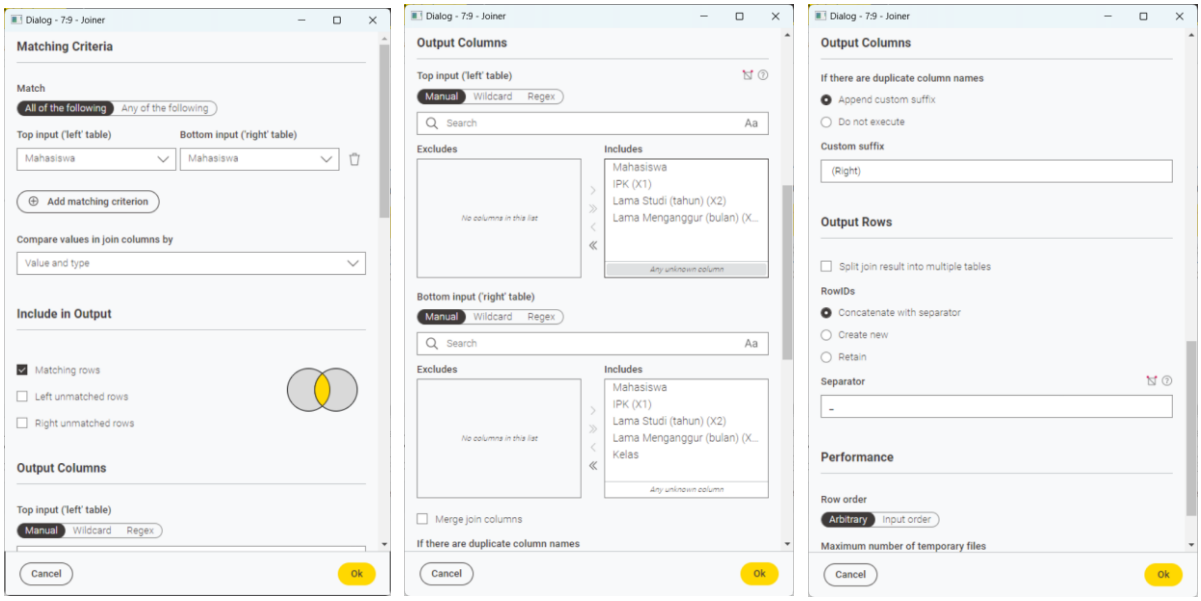
Rows: 15 | Columns: 6 Table Statistics

#	RowID	No (Number (integer))	Mahasiswa (String)	IPK (X1) (Number (double))	Lama Studi (tahun)... (Number (double))	Lama Menganggur ... (Number (double))	Kelas (String)
13	Row12	13	M	3.7	4.1	5	Cukup
14	Row13	14	N	3	5	7.4	Kurang
15	Row14	15	O	3.4	5	3.5	Baik

Gambar 1.52 Isi Tabel kedua yang berbeda dengan Tabel kesatu

- Tujuan Join di sini adalah menggabungkan dua tabel kiri dan kanan jika ada baris di kedua tabel yang sama Mahasiswa-nya.
- Penggunaan Column Filter sama dengan yang sudah dijelaskan sebelumnya.

Form Dialog Node Join ini cukup panjang dengan Opsi yang meliputi tabel kiri dan kanan, seperti dalam Gambar 1.53.



Gambar 1.53 Form Dialog Join

Penjelasan Form Dialog Join adalah sebagai berikut:

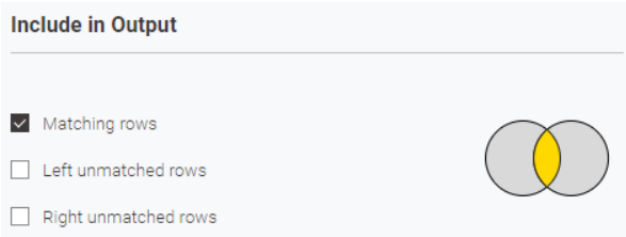
- **Kriteria Pencocokan (Matching Criteria):** Mendefinisikan logika untuk kriteria pencocokan:
  - **Semua dari berikut ini (All of the following)**: Jika dipilih, menggabungkan dua baris hanya ketika semua kriteria pencocokan terpenuhi.
  - **Salah satu dari berikut ini (Any of the following)**: Jika dipilih, menggabungkan dua baris ketika setidaknya satu dari kriteria pencocokan terpenuhi.
  - **Top Input** ('tabel kiri'): Pilih kolom dari tabel input atas yang akan digunakan untuk dibandingkan dengan kolom yang dipilih dari input bawah.
  - **Bottom Input** ('tabel kanan'): Pilih kolom dari tabel input bawah yang akan digunakan untuk dibandingkan dengan kolom yang dipilih dari input atas.
  - RowID dapat dibandingkan dengan RowID lain atau dengan kolom biasa, di mana RowID akan diinterpretasikan sebagai nilai string.

- **Compare values in join columns by:** Mendefinisikan cara membandingkan nilai di kolom join:

- **Value and Type**: Dua baris hanya akan cocok jika nilai mereka di kolom join yang dipilih memiliki nilai dan tipe yang sama. Misalnya, nilai Angka (integer) tidak akan pernah cocok dengan nilai Angka (long) karena mereka memiliki dua tipe yang berbeda.
- **String Representation**: Gunakan opsi ini jika Anda ingin nilai-nilai dikonversi menjadi string sebelum dibandingkan. Dengan cara ini, Anda hanya membandingkan nilai di kolom join yang dipilih.
- **Make integer types compatible**: Gunakan opsi ini untuk mengabaikan perbedaan tipe antara Angka (integer) dan Angka (long).

– **Include in Output:** mendefinisikan bagaimana output akan dihasilkan:
  - **Matching rows:** Sertakan hanya baris yang cocok pada pasangan kolom yang dipilih.
  - **Left unmatched rows:** Sertakan baris dari tabel input kiri yang tidak memiliki baris yang cocok di tabel input kanan.
  - **Right unmatched rows:** Sertakan baris dari tabel input kanan yang tidak memiliki baris yang cocok di tabel input kiri.

Dalam praktek ini, opsi pertama (**Matching rows**) yang dipilih. Diagram Venn dalam Gambar 1.54 membantu User memahami bahwa Baris akan digabung hanya jika ada fitur-fitur di kedua tabel yang memenuhi kriteria.



Gambar 1.54 Diagram Venn opsi *Matching rows*

– **Output Columns:** mendefinisikan kolom di kedua tabel yang akan diikutsertakan di output
– **If there are duplicate column names:** mendefinisikan jika ada nama kolom yang duplikasi dapat dilakukan dua opsi: menambahkah suffix (akhiran) kustom, atau tidak dieksekusi.
– **Output Rows:** mendefinisikan opsi terhadap baris yang dihasilkan yaitu: sambungkan dengan pemisah tertentu (concatenate with separator), buat baru (create new) atau biarkan apa adanya (retain). Separator untuk opsi pertama dapat didefinisikan di text-box di bawah opsi ini.
– **Split join result into multiple tables:** informasi dari opsi ini dapat dipelajari dan dipraktekkan sendiri oleh Pembelajar, misalnya melalui panel kiri side-bar.

Hasil dari menggunakan **matching rows**, adalah tabel baru Identik dengan Tabel kiri, karena hanya 12 mahasiswa A sampai L yang match/cocok dengan kriteria.

Untuk mendemonstrasikan hasil lainnya, maka kali ini akan digunakan juga kriteria **Right unmatched rows**, dimana baris di Tabel Kanan (bawah) yang tidak match juga akan diikutkan. Hasilnya diberikan dalam Gambar 1.55 yang memperlihatkan terdapat 15 baris

dengan tiga baris terakhir adalah baris di Tabel kanan yang tidak ada di Tabel kiri. Namun demikian, di sana tidak terlihat nama mahasiswa tambahannya. Untuk mengikutkan Nama Mahasiswa ini, diserahkan ke Pembelajar untuk bereksperimen dengan opsi yang ada.

1: Join result | 2: Left unmatched rows | 3: Right unmatched rows | Flow Variables

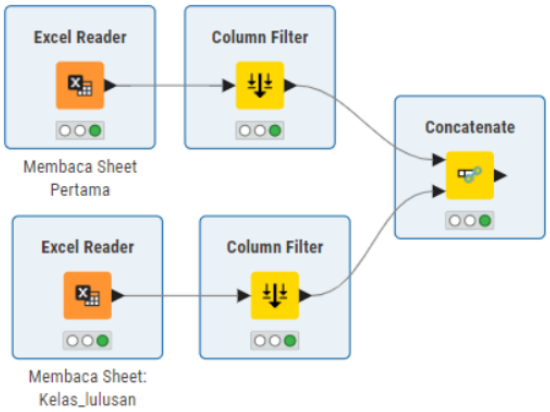
Rows: 15 | Columns: 9 — Table | Statistics

#	RowID	Mahasiswa (String)	IPK (X1) (Number)	Lama Stud... (Number)	Lama Men... (Number)	Mahasisw... (String)	IPK (X1) (... (Number)	Lama Stud... (Number)	Lama Men... (Number)	Kelas (String)
1	Row0_Row0	A	3	4.1	5.1	A	3	4.1	5.1	Cukup
2	Row1_Row1	B	3.2	4	5.6	B	3.2	4	5.6	Cukup
3	Row2_Row2	C	2.8	4.6	7	C	2.8	4.6	7	Kurang
4	Row3_Row3	D	2.9	4	8	D	2.9	4	8	Kurang
5	Row4_Row4	E	2.5	5	9	E	2.5	5	9	Kurang
6	Row5_Row5	F	3.7	4.1	4.8	F	3.7	4.1	4.8	Cukup
7	Row6_Row6	G	3.5	3.8	3.3	G	3.5	3.8	3.3	Baik
8	Row7_Row7	H	3.2	4.3	3.5	H	3.2	4.3	3.5	Baik
9	Row8_Row8	I	2.9	4.6	4	I	2.9	4.6	4	Cukup
10	Row9_Row9	J	2.7	4.4	4.3	J	2.7	4.4	4.3	Cukup
11	Row10_Row10	K	3.8	3.8	4	K	3.8	3.8	4	Baik
12	Row11_Row11	L	3.4	5	5.2	L	3.4	5	5.2	Cukup
13	?_Row12	?	?	?	?	M	3.7	4.1	5	Cukup
14	?_Row13	?	?	?	?	N	3	5	7.4	Kurang
15	?_Row14	?	?	?	?	O	3.4	5	3.5	Baik

Gambar 1.55 Tabel baru hasil Join “Right unmatched rows”

Workflow 7: Penyambungan Tabel berdasarkan Kolom

Node Concatenate berfungsi untuk menyambung dua tabel data berdasarkan kolom-kolom fitur, dengan workflow yang umum sebagaimana dalam Gambar 1.56. Tabel pada inport 0 atas diberikan sebagai tabel input pertama (port input atas), dan tabel pada inport 1 adalah tabel kedua. Kolom dengan nama yang sama akan digabungkan (jika tipe kolom berbeda, tipe kolom yang digunakan adalah tipe dasar umum dari kedua tipe kolom input). Jika salah satu tabel input memiliki nama kolom yang tidak dimiliki oleh tabel lain, kolom tersebut dapat diisi dengan nilai yang hilang atau dihapus, yaitu, mereka tidak akan ada di tabel output

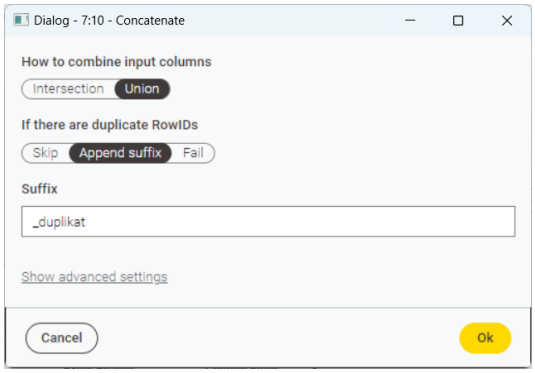


Gambar 1.56 Workflow Penyambungan Tabel dengan Concatenate

Dua tabel yaitu tabel pertama, memiliki kolom Mahasiswa berisi 12 Nama lulusan A sampai L tanpa fitur Kelas, dan tabel kedua, juga ada kolom Mahasiswa berisi 15 mahasiswa A

sampai O dengan fitur Kelas. Berbeda dengan Node Join, Concatenate memiliki opsi yang jauh lebih sederhana seperti dalam Gambar 1.57, yaitu:

- Cara mengombinasikan input: dapat dipilih intersection atau union. Intersection hanya menggabung kolom yang dimiliki kedua tabel saja; union menggabung kolom yang tidak dimiliki kedua tabel.
- Jika ada ID baris (RowID) yang sama, dapat diperlakukan: dilewati (skip), tambahkan akhiran (Append suffix), atau digagalkan (Fail). Nama akhirannya didefinisikan di bagian Suffix.



Gambar 1.57 Form Dialog Concatenate

Jika dipilih opsi Union dan Append suffix (tambah akhiran _duplikat), maka hasil penyambungan akan seperti Gambar 1.58. Dapatkah Pembelajar menjelaskan penyebabnya? Bagaimana jika dipilih Intersection, apa hasilnya?

▶ 1: Concatenated table | Flow Variables

Rows: 15 | Columns: 5 — Table | Statistics

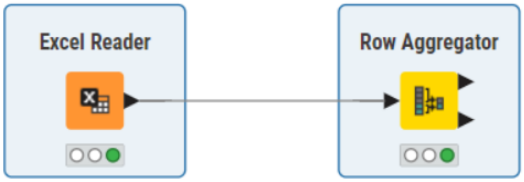
#	RowID	Mahasiswa String	IPK (X1) Number (double)	Lama Studi (tahun) (X2) Number (double)	Lama Menganggur (bula... Number (double)	Kelas String
1	Row0	A	3	4.1	5.1	?
2	Row1	B	3.2	4	5.6	?
3	Row2	C	2.8	4.6	7	?
4	Row3	D	2.9	4	8	?
5	Row4	E	2.5	5	9	?
6	Row5	F	3.7	4.1	4.8	?
7	Row6	G	3.5	3.8	3.3	?
8	Row7	H	3.2	4.3	3.5	?
9	Row8	I	2.9	4.6	4	?
10	Row9	J	2.7	4.4	4.3	?
11	Row10	K	3.8	3.8	4	?
12	Row11	L	3.4	5	5.2	?
13	Row12	M	3.7	4.1	5	Cukup
14	Row13	N	3	5	7.4	Kurang
15	Row14	O	3.4	5	3.5	Baik

Gambar 1.58 Hasil Concatenate dengan Opsi Union dan Append Suffix

Workflow 8: Agregasi Baris terhadap Kolom

Analis data suatu saat memerlukan 'kesimpulan' dari nilai dari beberapa baris atau kelompok / kelas baris yang ada berdasarkan kolom tertentu. Kesimpulan dimaksud

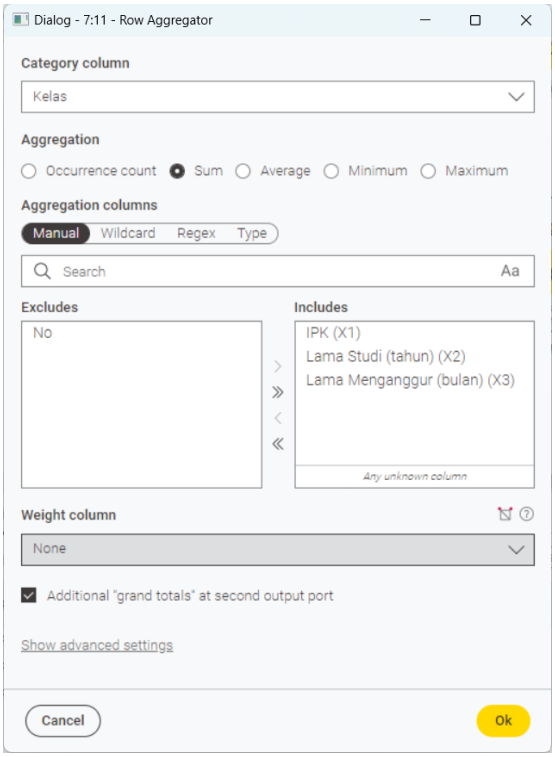
misalnya jumlah kemunculan baris (*occurence count*), atau nilai rerata (*average*) kelompok baris. Node Row Aggregator dapat mengakomodasi kebutuhan ini. Workflow yang digunakan umumnya seperti diberikan dalam Gambar 1.59.



Gambar 1.59 Workflow Agregasi Baris

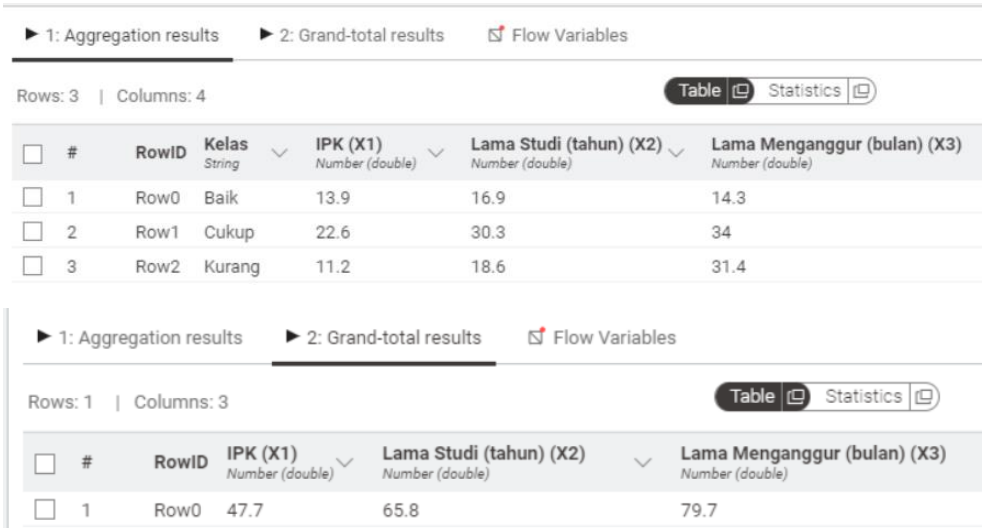
Form dialog Row Aggregator diberikan dalam Gambar 1.60 dengan opsi sebagai berikut:

- Category column, dapat digunakan untuk mengagregasi anggota tiap kelas atau kategori. Misalnya terhadap kolom “Kelas”
- Fungsi agregasi yang dapat dipilih untuk ‘menyimpulkan’ nilai yaitu: jumlah kemunculan (occurence count), jumlah (Sum), rerata (Average), minimum dan maksimum. Di praktek kali ini dipilih Sum
- Kolom yang diagregasi, dalam hal ini dipilih seluruh kolom numerik kecual No.
- Pembobotan kolom, yaitu dengan memilih kolom yang mendefinisikan bobot di mana suatu nilai dikalikan sebelum agregasi. Perhatikan bahwa hanya fungsi agregasi "Jumlah" dan "Rata-rata" yang mendukung kolom bobot.
- Opsi untuk menambah port output “Grand-total”
- Opsi lainnya silakan dicoba sendiri oleh Pembelajar dalam eksperimennya.



Gambar 1.60 Form Dialog Row Aggregator

Hasil dari opsi-opsi yang dipilih diberikan dalam Gambar 1.61, dimana di bagian atas adalah hasil agregasi Sum terhadap tiap anggota Kelas Baik, Cukup dan Kurang, sedangkan di bagian bawah adalah hasil Grand-total atau total keseluruhan Sum di semua kelas.



Gambar 1.61 Hasil Row Aggregator dengan fungsi Sum pada Kelas

Workflow 9: Value Lookup

Node Value Lookup digunakan untuk mencari nilai dari tabel data berdasarkan nilai kunci yang cocok di tabel kamus (*dictionary table*). Kegunaan dan cara kerjanya sebagai berikut:

1. **Input Ganda**: Node ini memiliki dua input, yaitu:
    - **Tabel data**: Tabel utama yang berisi data yang ingin ditambahkan informasinya.
    - **Tabel kamus (dictionary)**: Tabel referensi yang berisi pasangan kunci dan nilai yang akan digunakan untuk pencarian.
2. **Kolom Pencarian**: tentukan satu kolom dari tabel data yang akan digunakan sebagai kunci pencarian di tabel kamus. Kunci ini digunakan untuk mencocokkan nilai di kolom kunci dari tabel kamus.

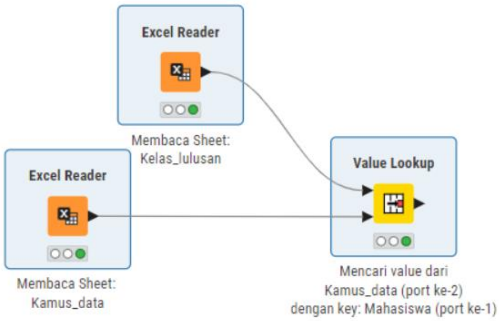
Workflow yang umumnya digunakan untuk menerapkan Value Lookup diberikan dalam Gambar 1.62, dengan skenario Lookup sebagai berikut:

- Tabel atas (kiri) berisi fitur nama Mahasiswa yang lulus dan fitur lainnya, namun tidak ada fitur kota asal Mahasiswa
- Tabel bawah (kanan) adalah Kamus data yang hanya berisi fitur nama Mahasiswa dan Kota asalnya.
- Node Value Lookup akan membuat tabel baru berisi nama Mahasiswa yang lulus seperti yang ada di tabel atas, dan jika nama Mahasiswa tersebut ada di Kamus data

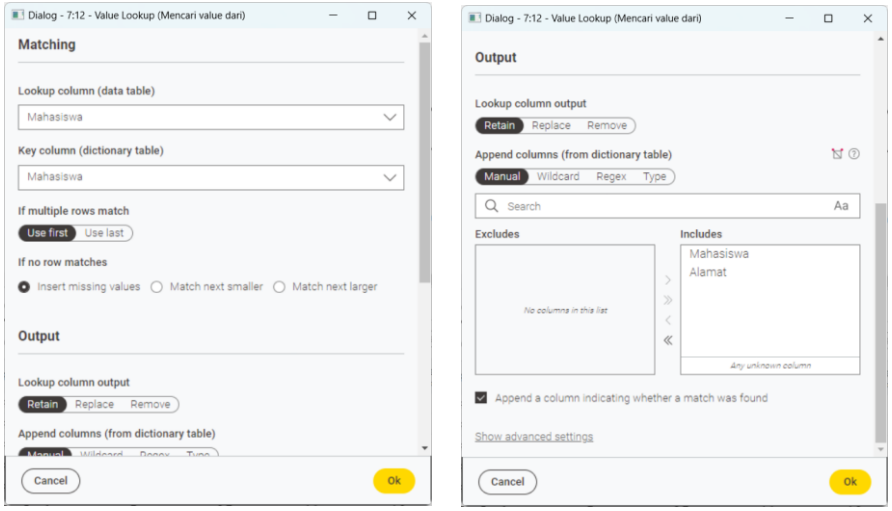
Form dialog Value Lookup diberikan dalam Gambar 1.63, dengan opsi:

- Kolom-kolom fitur kunci yang akan dipadankan datanya antara tabel data (port atas) dan kampus data (port bawah); dalam hal ini misalnya Mahasiswa
- If multiple rows match: perlakuan jika banyak baris yang cocok adalah gunakan yang pertama atau terakhir

– Jika tidak ada yang cocok, maka perlakukan: sisipkan nilai yang hilang (insert missing values), cocokkan dengan yang mirip berikutnya (match next similar) atau cocokkan yang lebih besar berikutnya (match next larger)
– Untuk output, digunakan untuk menentukan isi kolom yang dipilih sebagai kolom lookup (tabel data): *Retain*, isi kolom lookup tetap tidak berubah. *Replace*, isi sel digantikan dengan nilai dari tabel kamus. Jika ada kecocokan, nilai dari kolom yang dipilih dimasukkan, jika tidak, nilai asli dipertahankan atau diganti dengan nilai kosong. *Delete*, kolom lookup dihapus seluruhnya dari tabel output.
– Tentukan fitur mana yang akan diikutkan atau tidak di bagian Includes atau excludes



Gambar 1.62 Workflow penerapan Value Lookup



Gambar 1.63 Form Dialog Value Lookup

Hasil dari penerapan opsi-opsi ini diberikan dalam Gambar 1.64, dimana terbentuk 15 baris dan 9 kolom, termasuk dua kolom Alamat yang berhasil di-Lookup dari Kamus data, dan Match Found bertipe Boolean, yang terisi dengan nilai *True*.

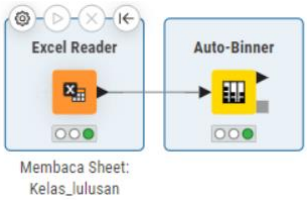
Rows: 15 | Columns: 9 — Table | Statistics

#	RowID	No	Mahasiswa	IPK (X1)	Lama Stud...	Lama Men...	Kelas	Mahasisw...	Alamat	Match Fou...
		Number (integ...	String	Number (doubl...	Number (doubl...	Number (doubl...	String	String	String	Boolean value
1	Row0	1	A	3	4.1	5.1	Cukup	A	Malang	true
2	Row1	2	B	3.2	4	5.6	Cukup	B	Pasuruan	true
3	Row2	3	C	2.8	4.6	7	Kurang	C	Malang	true

Gambar 1.64 Hasil Value Lookup

Workflow 10: Auto Binner

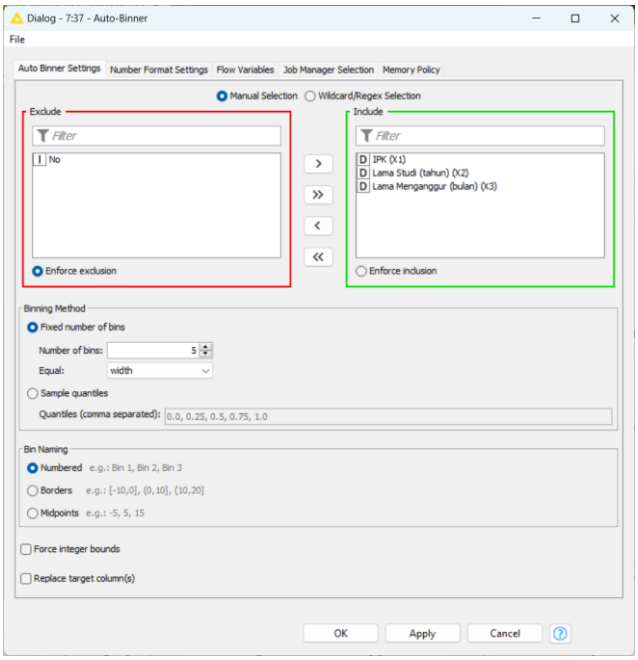
Node Auto Binner memungkinkan pengelompokan data *numerik* ke dalam interval, yang disebut **bins**. Workflow penggunaan Auto Binner diberikan dalam Gambar 1.65



Gambar 1.65 Workflow penggunaan Auto Binner

Form dialog Auto Binner diberikan dalam Gambar 1.66, dimana terdapat beberapa opsi:

- Memilih kolom fitur numerik yang akan dibuat intervalnya (bins)
- Metode binning:
  - Gunakan *Fixed number of bins* untuk interval dengan lebar yang sama atau frekuensi elemen yang sama.
  - *Sample quantiles* menghasilkan bin berdasarkan daftar probabilitas Kuartil. Nilai terkecil sesuai dengan probabilitas 0, dan yang terbesar dengan probabilitas 1. Metode estimasi yang digunakan adalah Tipe 7, seperti di R, S, dan Excel.
- Penamaan Bin: *Numbered* untuk bin yang diberi label angka dengan awalan "Bin"; Borders menggunakan notasi interval "(a,b]", atau; Midpoints menampilkan titik tengah interval.
- Paksa batas integer: Memaksa batas interval menjadi integer dengan mengonversi batas desimal, di mana batas bawah adalah nilai terendah yang dibulatkan ke bawah dan batas atas adalah nilai tertinggi yang dibulatkan ke atas.
  Contoh: [0.1,0.9], (0.9,1.8] menjadi [0,1], (1,2], dan [3.9,4.1], (4.1,4.9], (4.9,5.1] menjadi [3,5], (5,6]



Gambar 1.66 Form dialog Auto Binner

Hasil dari opsi sebagaimana dalam Gambar 1.66, diberikan dalam Gambar 1.67, dimana terlihat terdapat kolom-kolom IPK, Lama Studi dan Lama Menganggur yang sudah diberi nama intervalnya dengan awalan "Bin", dengan jumlah interval 5. Penerapan kombinasi opsi yang ada diserahkan ke Pembelajaran sendiri dalam eksperimennya.

Rows: 15 | Columns: 9 — Table | Statistics

#	RowID	No (Number (integ...)	Mahas... (String)	IPK (X1) (Number (doubl...)	Lama Stud... (Number (doubl...)	Lama Men... (Number (doubl...)	Kelas (String)	IPK (X1) [... (String)	Lama Stud... (String)	Lama Men... (String)
1	Row0	1	A	3	4.1	5.1	Cukup	Bin 2	Bin 2	Bin 2
2	Row1	2	B	3.2	4	5.6	Cukup	Bin 3	Bin 1	Bin 3
3	Row2	3	C	2.8	4.6	7	Kurang	Bin 2	Bin 4	Bin 4
4	Row3	4	D	2.9	4	8	Kurang	Bin 2	Bin 1	Bin 5
5	Row4	5	E	2.5	5	9	Kurang	Bin 1	Bin 5	Bin 5
6	Row5	6	F	3.7	4.1	4.8	Cukup	Bin 5	Bin 2	Bin 2
7	Row6	7	G	3.5	3.8	3.3	Baik	Bin 4	Bin 1	Bin 1

Gambar 1.67 Hasil Binning pada IPK, Lama Studi dan Lama Menganggur

Literasi Data dalam Data Tekstual

Literasi kedua adalah mengenai data yang berbentuk tekstual, yang biasanya diperoleh dari pengumpulan data hasil pencarian di mesin pencari (Google dll.), abstrak publikasi ilmiah, konten *blog*, *curriculum vitae*, berita online, komentar orang terhadap produk, status media sosial, iklan lowongan kerja, dan lain-lain. Data tekstual mungkin saja disimpan dalam sebuah file teks, dimana tiap baris mewakili sebuah rekaman data, seperti abstrak berita atau sumber lainnya; atau mungkin juga di dalam sebuah data tabular, yang mana terdapat kolom yang bertipe string atau dokumen yang berisi data sebagaimana dimaksud.

Data tekstual untuk praktek KNIME disimpan dalam file teks bernama "searching-en.txt", berisi 10 hasil pencarian di Google dengan kata kunci 'data mining and data science'. Lima baris pertama data tekstual ini ditayangkan dalam Gambar 1.66.

```
1  Data Science: Data Science is a field or domain which includes and involves
   working with a huge amount of data and uses it for building predictive,
   prescriptive and prescriptive analytical models. Data Mining: Data Mining is a
   technique to extract important and vital information and knowledge from a huge
   set/libraries of data.
2  Data Science: An interdisciplinary field, data science relies on scientific
   methods, processes, algorithms, and systems to extract or extrapolate knowledge
   and insights from structured and unstructured data. Knowledge from data is then
   applied across a wide range of domains. Data Mining: The process of discovering
   patterns in large data sets through the use of methods involving a combination of
   machine learning, statistics, and database systems.
3  Data science is an interdisciplinary field combining algorithms, scientific
   methods, and systems to extract insights and knowledge from unstructured and
   structured data. To uncover these insights, data mining employs various methods,
   such as statistical analysis, machine learning, and pattern recognition
   algorithms.
4  Data mining and data science have become ubiquitous terms used interchangeably in
   analytics and business contexts. However, they refer to related but distinct
   processes, mindsets, and capabilities for extracting value from data.
5  Data mining is the process of discovering patterns, trends, and insights from
   vast datasets. It involves applying various techniques such as clustering,
   classification, regression, and association rule mining to identify meaningful
   relationships within the data. The primary goal of data mining is to extract
   valuable knowledge from raw, unstructured datasets.
```

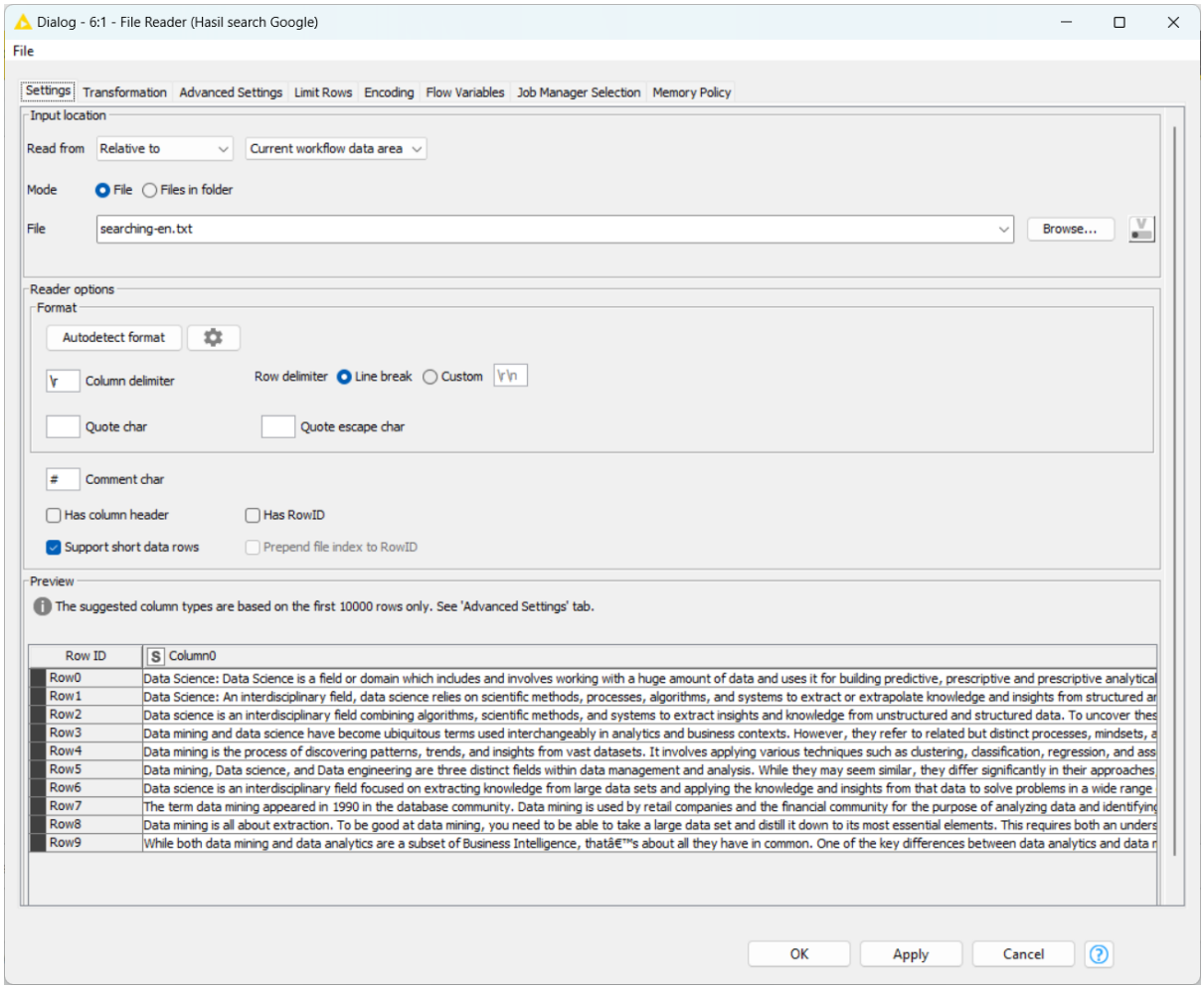
Gambar 1.66 Data tekstual hasil pencarian di Google

Workflow 1: Membaca data di file teks

Node yang digunakan untuk membaca sebuah file teks adalah File Reader seperti yang diberikan dalam Gambar 1.67. Form dialog Node File Reader dijelaskan dalam Gambar

1.68, yang jika diperhatikan, mirip dengan Form dialog Node CSV Reader. Pembelajar dipersilakan untuk memeriksanya lebih lanjut.

Gambar 1.67 Node File Reader



Gambar 1.68 Form dialog Node File Reader

Analis mungkin akan memperlakukan data tekstual ini dengan beragam pendekatan. Misalnya membiarkannya seperti data aslinya, yaitu semua data per baris disimpan di sebuah kolom saja; dalam contoh ini Column0. Agar dapat tampil seperti Gambar 1.68, pemisah kolom (*column delimiter*) diset ke '\r' atau *carriage return* (pindah baris seperti di'enter').

Jika Pembelajar memasukkan delimiter lain, misalnya sebuah spasi kosong, maka tiap baris data akan dipisahkan setiap spasi kosong; akibatnya akan terbentuk kolom-kolom yang berisi kata-kata seperti dalam Gambar 1.69. Namun untuk praktek selanjutnya, kita akan

membiarkan satu baris utuh tanpa pemisahan sebagai dataset. Opsi di form dialog File Reader pun diserahkan kepada Pembelajaran untuk mendalaminya.

Preview

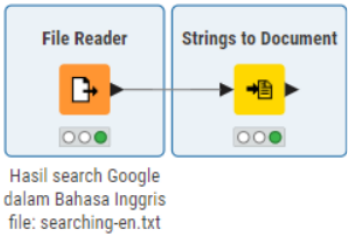
The suggested column types are based on the first 10000 rows only. See 'Advanced Settings' tab.

Row ID	Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10	Column
Row0	Data	Science:	Data	Science	is	a	field	or	domain	which	includes	and
Row1	Data	Science:	An	interdisciplin...	field,	data	science	relies	on	scientific	methods,	processes,
Row2	Data	science	is	an	interdisciplin...	field	combining	algorithms,	scientific	methods,	and	systems
Row3	Data	mining	and	data	science	have	become	ubiquitous	terms	used	interchange...	in
Row4	Data	mining	is	the	process	of	discovering	patterns,	trends,	and	insights	from
Row5	Data	mining,	Data	science,	and	Data	engineering	are	three	distinct	fields	within
Row6	Data	science	is	an	interdisciplin...	field	focused	on	extracting	knowledge	from	large
Row7	The	term	data	mining	appeared	in	1990	in	the	database	community.	Data
Row8	Data	mining	is	all	about	extraction.	To	be	good	at	data	mining,
Row9	While	both	data	mining	and	data	analytics	are	a	subset	of	Business

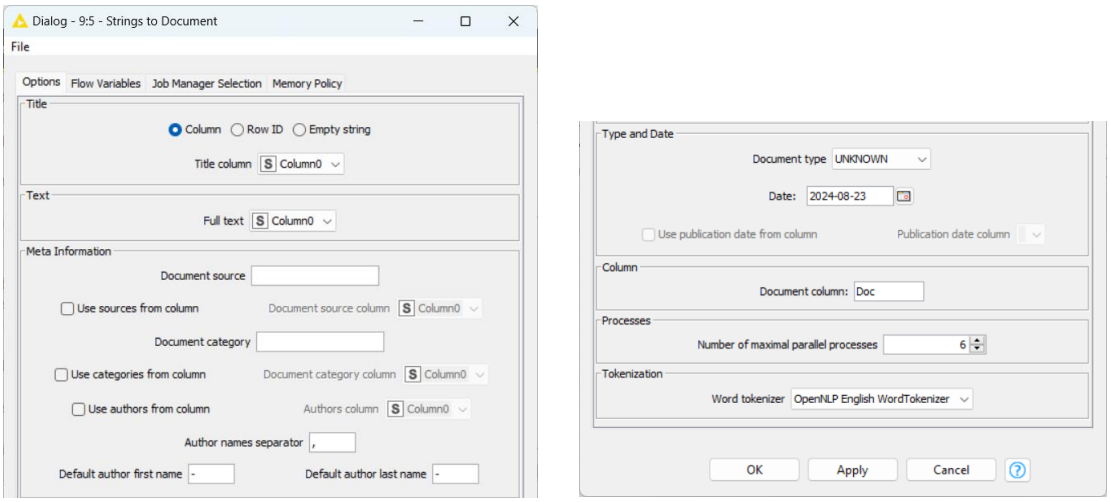
Gambar 1.69 Tiap baris dipisahkan per kata yang menjadi kolom

Workflow 2: Mengubah tipe String menjadi Text Document

KNIME memiliki beberapa tipe data untuk memperlakukan data tekstual, antara lain String dan Text Document. Text Document disediakan untuk mengakomodasi kegiatan ekstraksi informasi dari dokumen yang mungkin berbentuk artikel Transaction (jurnal seperti milik IEEE), proceeding (buku hasil seminar/konferensi), dan Buku. Dokumen artikel memiliki meta informasi seperti nama Author dan Kategori. Namun untuk dokumen pada umumnya, node ini juga efektif untuk digunakan. Gambar 1.70 adalah workflow untuk mengubah tipe String dari Column0 yang berisi data hasil pencarian Google, menjadi tipe Text document. Opsi di Forum dialog node ini digambarkan dalam Gambar 1.71



Gambar 1.70 Workflow Penggunaan node Strings to Document



Gambar 1.70 Form dialog Node Strings to Document

Opsi pengaturan yang tersedia dalam Form dialog adalah:

- **Title**: Pilih apakah akan menggunakan isi kolom, row ids, atau string kosong sebagai judul.
- **Title column**: Kolom yang berisi string untuk digunakan sebagai judul (jika "Use title from column" dipilih, jika tidak, judul default akan dibuat).
- **Full text**: Kolom yang berisi string untuk digunakan sebagai teks.
- **Document source**: Sumber yang ditetapkan untuk semua dokumen (jika "Use sources from column" tidak dipilih).
- **Use sources from column**: Jika dipilih, nilai string dari kolom tertentu akan digunakan sebagai sumber dokumen.
- **Document source column**: Kolom yang berisi string yang digunakan sebagai sumber dokumen. Tidak ada sumber yang ditetapkan untuk nilai yang kosong.
- **Document category**: Kategori yang ditetapkan untuk semua dokumen (jika "Use categories from column" tidak dipilih).
- **Use categories from column**: Jika dipilih, nilai string dari kolom tertentu akan digunakan sebagai kategori dokumen.
- **Document category column**: Kolom yang berisi string yang digunakan sebagai kategori. Tidak ada kategori yang ditetapkan untuk nilai yang kosong.
- **Use author(s) from column**: Jika dipilih, nilai string dari kolom tertentu akan digunakan sebagai penulis.
- **Authors column**: Kolom yang berisi nama penulis dalam bentuk string yang dipisahkan oleh tanda koma. Nama kedua akan ditambahkan ke nama depan.
- **Author name separator**: String yang memisahkan nama penulis dalam kolom penulis.
- **Default author first name**: Nama depan penulis default jika "use author(s) from column" tidak dipilih.
- **Default author last name**: Nama belakang penulis default jika "use author(s) from column" tidak dipilih.
- **Document type**: Tipe yang ditetapkan untuk semua dokumen.
- **Date**: Tanggal publikasi yang ditetapkan untuk semua dokumen (jika "Use publication date from column" tidak dipilih).
- **Use publication date from column**: Jika dipilih, nilai tanggal dari kolom tertentu akan digunakan sebagai tanggal publikasi dokumen.
- **Publication date column**: Kolom yang berisi tanggal yang digunakan sebagai tanggal publikasi. Jika "Use publication date from column" dipilih, tanggal diambil dari kolom, jika tidak, tanggal saat ini dari field "Publication date" digunakan.
- **Document column**: Tentukan nama kolom dokumen yang akan dibuat.
- **Word tokenizer**: Pilih tokenizer yang digunakan untuk tokenisasi kata. Untuk preferensi, silakan dicek Preferences -> KNIME -> Textprocessing untuk deskripsi masing-masing tokenizer.

Hasil dari opsi yang diset dalam Gambar 1.70, diberikan dalam Gambar 1.71, dimana sudah terlihat bahwa kolom Doc berisi hampir sama dengan Column0, namun dengan tipe Text Document yang ditandai dengan adanya tanda quotes di awal dan akhir tiap baris data.

Alasan transformasi tipe data ini adalah, KNIME menyediakan metode data mining yang hanya menerima data input dengan format tertentu, misalnya Text document, Bit vector, Set, List, dan lain-lain.

Rows: 10 | Columns: 2 Table Statistics

#	RowID	Column0 *String*	Doc *Text document*
1	Row0	Data Science: Data Science is a field or domain which includes and involves w...	"Data Science: Data Science is a field or domain which includes and involves w...
2	Row1	Data Science: An interdisciplinary field, data science relies on scientific metho...	"Data Science: An interdisciplinary field, data science relies on scientific metho...
3	Row2	Data science is an interdisciplinary field combining algorithms, scientific meth...	"Data science is an interdisciplinary field combining algorithms, scientific meth...
4	Row3	Data mining and data science have become ubiquitous terms used interchang...	"Data mining and data science have become ubiquitous terms used interchang...
5	Row4	Data mining is the process of discovering patterns, trends, and insights from v...	"Data mining is the process of discovering patterns, trends, and insights from v...

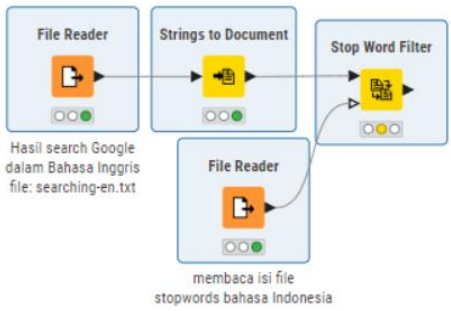
Gambar 1.71 Hasil transformasi String ke Document

Workflow 3: Menyaring Stop Word

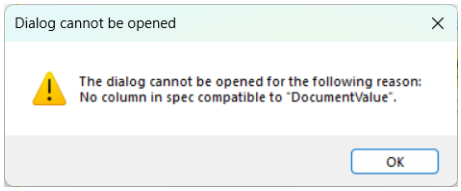
Stop words adalah kata-kata umum dalam bahasa yang sering muncul tetapi memiliki nilai informatif yang rendah, seperti "dan", "di", "adalah", "the", "is", dan "on". Kata-kata ini tidak membantu dalam memahami makna utama teks, terutama dalam proses Text Mining atau Natural Language Processing (NLP). Stopwords disarankan untuk dihapus karena:

- **Mengurangi kebisingan (noise)**: Stop words tidak memberikan informasi penting, sehingga menghapusnya membantu menyederhanakan analisis.
- **Meningkatkan efisiensi**: Mengurangi jumlah kata yang diproses mempercepat algoritma.
- **Fokus pada kata penting**: Dengan menghapus stop words, analisis dapat fokus pada kata-kata yang lebih relevan dan bermakna bagi konteks teks.
- **Mereduksi dimensi**: Menghapus stop words membantu **mereduksi dimensi** dalam model analisis teks. Dengan lebih sedikit kata unik (fitur), model menjadi lebih sederhana, mengurangi risiko **overfitting**, serta meningkatkan performa komputasi, sehingga analisis lebih efisien dan relevan.

Workflow untuk menyaring stop words secara umum diberikan dalam Gambar 1.72, dimana terlihat penggunaan Node Stop Word Filter dan node String to Document di tengah-tengah Node File Reader dan Stop Word Filter. Hal ini dilakukan karena Node Stop Word Filter hanya mengolah data bertipe *Text Document*. Jika ditarik benang proses langsung dari File Reader ke Stop Word Filter, maka akan memunculkan pesan peringatan yang menunjukkan ketiadaan kolom bertipe Text Document yang seperti dalam Gambar 1.73



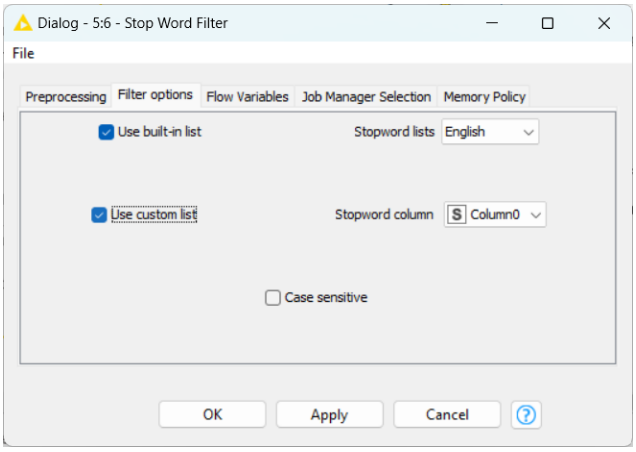
Gambar 1.72 Workflow penerapan Node Stop Word Filter



Gambar 1.73 Pesan Peringatan Ketiadaan Kolom bertipe Document

Pengaturan yang dapat dilakukan di Form Dialog Stop Words Filter:

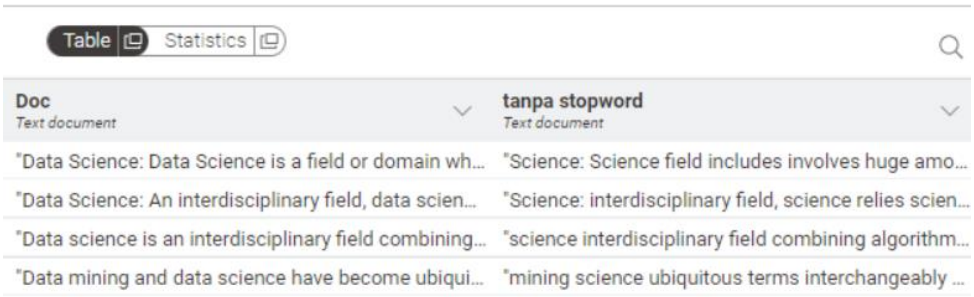
- Di Tab "Preprocessing": pilih kolom dokumen yang akan difilter, dan nama kolom hasil penyaringan, misalnya "tanpa stopword"
- Di Tab "Filter options": tentukan stop words built-in yang diterapkan, dalam hal ini dipilih "English"



Gambar 1.74 Form Dialog Stop Words Filter

- Penggunaan daftar stop words secara kustom, termasuk jika diinginkan stop words Bahasa Indonesia, dengan cara:
  - Gunakan Node file reader untuk membuka daftar kata stop word Bahasa Indonesia
  - Sambungkan output File reader dimaksud, ke port input kedua di Node Stop Words Filter (lihat Gambar 1.72), dan biasanya masuk ke Column0

Hasilnya diberikan dalam Gambar 1.75, dimana terlihat kolom tanpa stopword, beberapa kata sudah disaring seperti "is, a, an, and". Namun di sini kata "Data" pun tersaring. Mengapa demikian? Karena ternyata kata "Data" ada di dalam daftar stop words Bahasa Indonesia. Untuk menghindari hal ini, kita dapat membuang dulu kata-kata yang tidak ingin disaring dari daftar stop words sebelum menggunakannya di Node ini.



Gambar 1.75 Hasil penerapan Stop Words Filter

Workflow 4: Membuat Bag of Words

Selanjutnya adalah membuat Bag of Words dari sebuah dokumen. Workflow yang digunakan seperti dalam Gambar 1.76, dimana node **Bag of Words Creator** disambungkan setelah Node Stop Word Filter. Node ini diterapkan untuk menghasilkan

semua kata yang ada di semua baris dokumen, yang dalam praktek ini adalah yang ada di kolom tanpa stopword. Hasil dari node ini diberikan dalam Gambar 1.78, di kolom Term, yang memperlihatkan individual kata yang ada di seluruh dokumen.

Gambar 1.76 Workflow Bag of Words Creator

▶ 1: Documents output table | Flow Variables

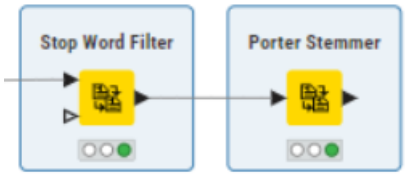
Rows: 251 | Columns: 2 — Table | Statistics

#	RowID	tanpa stopword (Text document)	Term (Term)
1	Row0	"Science: Science field includes involves huge amount building predictive, pre...	Science[]
2	Row1	"Science: Science field includes involves huge amount building predictive, pre...	:[]
3	Row2	"Science: Science field includes involves huge amount building predictive, pre...	field[]
4	Row3	"Science: Science field includes involves huge amount building predictive, pre...	includes[]
5	Row4	"Science: Science field includes involves huge amount building predictive, pre...	involves[]
6	Row5	"Science: Science field includes involves huge amount building predictive, pre...	huge[]

Gambar 1.78 Hasil Bag of Words

Workflow 5: Porter Stemmer

Stemmer adalah sebuah alat atau metode untuk mengambil bentuk dasar dari sebuah kata, yaitu dengan cara membersihkan imbuhannya. Dalam Bahasa Inggris, imbuhan termasuk ~ing, ~ed, ~ist, ~ation, dan lainnya, sedangkan daam Bahasa Indonesia termasuk awalan dan akhiran (me/mem~, di~, ber~, ~kan, dll.). Porter stemmer adalah salah satu alat stemming kata yang cukup populer dan dalam praktek ini, Porter Stemmer diterapkan pada hasil pembersihan Stowp Words Bahasa Inggris seperti dalam Gambar 1.79



Gambar 1.79 Penggunaan Porter Stemmer

Hasil dari Porter Stemmer diberikan dalam Gambar 1.80, dimana beberapa kata telah 'dipotong' imbuhannya, seperti science menjadi scienc, interdiciplinary menjadi interdiciplinari, dan lainnya. Pembelajar dapat mengamati sendiri perubahan yang dilakukan oleh Porter Stemmer ini. Selain Porter Stemmer juga tersedia stemming lainnya seperti Snowball dan Kuhlen yang dapat dipraktekan sendiri perbedaannya dengan Porter Stemmer.

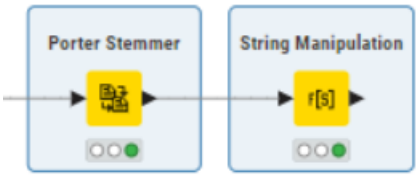
Table | Statistics

nostopword *Text document*	Stemmed *Text document*
"Data Science: Data Science field domain in...	"Data Scienc: Data Scienc field domain includ involv huge a...
"Data Science: interdisciplinary field, data s...	"Data Scienc: interdisciplinari field, data scienc reli scientif m...
"Data science interdisciplinary field combini...	"Data scienc interdisciplinari field combin algorithm, scientif ...

Gambar 1.80 Hasil Porter Stemmer

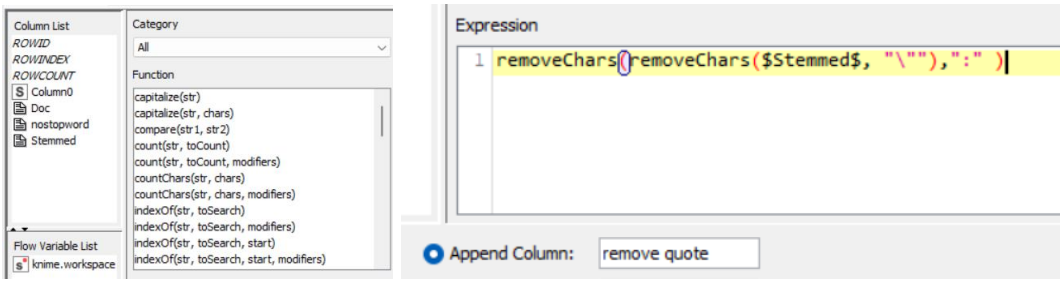
Workflow 6: String Manipulation

Setelah di-stemming, maka akan dipraktekan menggunakan node String Manipulation seperti nampak dalam Gambar 1.81.



Gambar 1.81 Workflow penggunaan String Manipulation

Di dalam form dialog node ini, manipulasi terhadap string dilakukan dengan mengunakan sejumlah fungsi string yang dikenal seperti capitalize, remove, removeChars, replace dan lainnya (Gambar 1.82). Namun untuk memanfaatkan fungsi ini, kita harus menuliskan ekspresinya yang dalam Buku ini tidak dijelaskan lebih lanjut. Ekspresi yang digunakan di sini adalah untuk membuang dua simbol atau karakter terhadap kolom Stemmed, secara rekursif menggunakan removeChars. Ekspresi removeChar di bagian dalam akan membuang tanda quote ("\"), dan setelah itu membuang tanda titik dua.



Gambar 1.82 Fungsi String dan Ekspresi penggunaannya

Hasilnya disimpan dalam kolom *remove quote*, diberikan dalam Gambar 1.83, dimana tanda quote dan titik dua sudah hilang, dan tipe data menjadi String kembali.

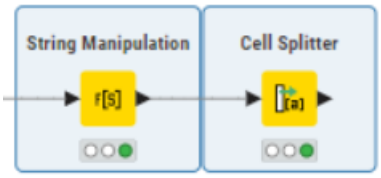
Table | Statistics

Stemmed *Text document*	remove quote *String*
"Data Scienc: Data Scienc field domain inclu...	Data Scienc Data Scienc field domain includ involv huge am...
"Data Scienc: interdisciplinari field, data scie...	Data Scienc interdisciplinari field, data scienc reli scientif m...
"Data scienc interdisciplinari field combin alg...	Data scienc interdisciplinari field combin algorithm, scientif ...

Gambar 1.83. Hasil String Manipulation

Workflow 7: Memisahkan Sel

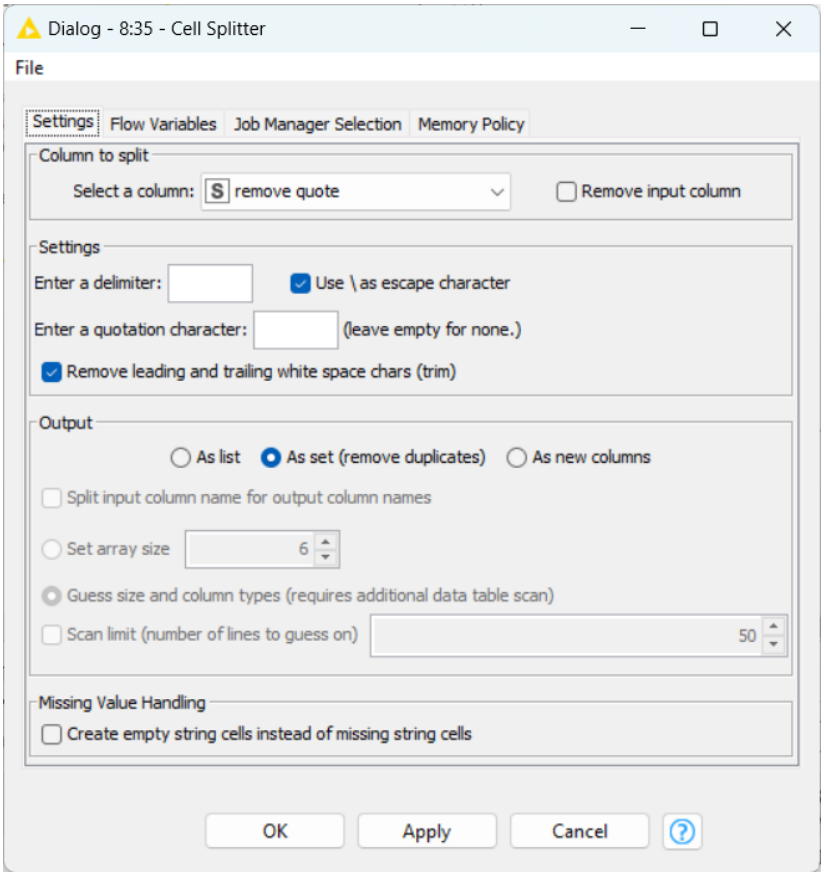
Setelah dimanipulasi, maka kolom remove quote akan dipisahkan tiap katanya menggunakan Cell Splitter menggunakan workflow seperti dalam Gambar 1.84.



Gambar 1.85 Workflow penggunaan Cell Splitter

Form dialog node ini diberikan dalam Gambar 1.85 dengan opsi yang akan dipraktekkan adalah:

- Enter delimiter: pemisah kata, dalam praktek ini digunakan sebuah spasi kosong
- Remove leading and trailing white space: membuang spasi sebelum dan setelah kata yang dipotong
- Output As Set (remove duplicate), menyimpan hasil pemisahan ke dalam Set, dimana hanya satu kata unique saja yang disimpan.



Gambar 1.86 Form Dialog Cell Splitter

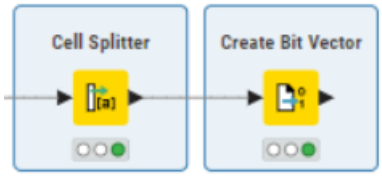
Hasilnya diberikan dalam Gambar 1.87, dimana terdapat Set yang menyimpan kata-kata yang ada di tiap dokumen atau string. Untuk praktek lanjutan, Pembelajar dapat mencoba tipe List atau As new columns sebagai Output.

remove quote String	remove quote_SplitResultSet Set
Data Scienc Data Scienc fiel...	[Data,Scienc,field,...]
Data Scienc interdisciplinari ...	[Data,Scienc,interdisciplinari,...]
Data scienc interdisciplinari ...	[Data,scienc,interdisciplinari,...]

Gambar 1.87 Hasil Cell Splitter

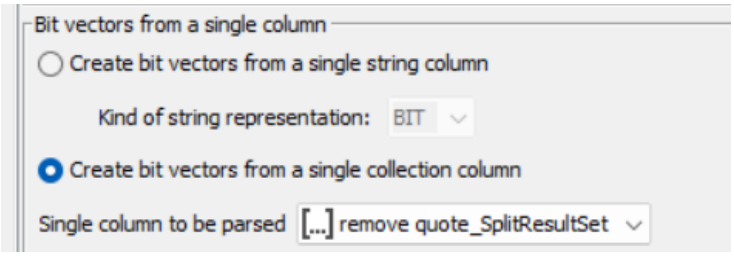
Workflow 8: Membuat Vektor Bit

Vektor bit adalah salah satu bentuk data yang sering diminta oleh algoritma data mining seperti Association rule mining. Workflow berikut mendemokan pembuatan Vektor bit menggunakan node Create Bit Vector, terhadap kolom yang sudah dipisahkan katanya dengan Cell Splitter (Gambar 1.88)



Gambar 1.88 Workflow pengunaan Create Bit Vector

Dalam form Dialog node ini, yang disetting adalah opsi Create bit vectors from a single collection column atau membuat vektor bit dari kolom koleksi (Set, List) tunggal (Gambar 1.89)



Gambar 1.89 Opsi yang disetting di Form Dialog Node Create Bit Vector

Hasilnya disimpan dalam kolom Bitvector, seperti dalam Gambar 1.90. Secara ringkas sebuah vektor bit memiliki panjang sama dengan jumlah kata yang ada di seluruh dokumen. Tiap kata yang dibaca sudah ditentukan posisinya di vektor bit, dan kata pertama yang dibaca menjadi bit ke-0 alias berada di paling kanan vektor bit. Jika di suatu baris, kata tersebut ada, maka diberi angka 1, jika sebaliknya maka diberi 0.

Misalnya, akan dibentuk bit vektor dari himpunan [a, b, c], [a, x, y], maka akan terbentuk bit vektor 000111 untuk [a, b, c], dimana bit ke-0, 1 dan 2 berturut-turut mewakili keberadaan [a, b, c]; untuk set [a, x, y] terbentuk vektor bit 11001, dimana angka 1 di sana mewakili keberadaan a di bit ke-0, x dan y di bit ke-4 dan 5.

remove quote... Set	BitVector Bit vector
[Data,mine,extract...	00000000000000000011111111100000000000000000000000000000000000000000000000000000...
[Data,mine,data,...]	00000000000000000000000000000000000000000000000000000000000000000000000000000000...
[Data,Scienc,field,...]	00000000000000000000000000000000000000000000000000000000000000000000000000000000...
[data,mine,analyt,...]	11111111111111111100000000000000000000000000000000000000000000000000000000000000...

Gambar 1.90 Bit Vector yang Dihasilkan

Workflow 9: Membuat Vektor Dokumen berbasis Term Frequency

Vektor dokumen juga dibutuhkan oleh beberapa program data mining, seperti classification atau clustering, dimana tiap vektor merupakan representasi keberadaan kata dalam dokumen tersebut. Prinsipnya mirip dengan Vektor Bit, namun bukan binary digit (Bit) yang dihasilkan tapi tabel seluruh kata yang ada di dokumen, dimana tiap sel berisi sebuah angka yang mewakili bobot kata tersebut.

Bobot yang akan digunakan adalah Term frequency (TF), yang mengukur seberapa sering sebuah kata (term) muncul dalam suatu dokumen relatif terhadap panjang dokumen tersebut. Secara umum, TF membantu memberikan gambaran tentang kata mana yang lebih menonjol dalam dokumen individual.

Gambar 1.91 Workflow Pembuatan Document Vector

Workflow yang digunakan diberikan dalam Gambar 1.91 yang merupakan kelanjutan dari Workflow 6; dengan node utama yang digunakan yaitu:

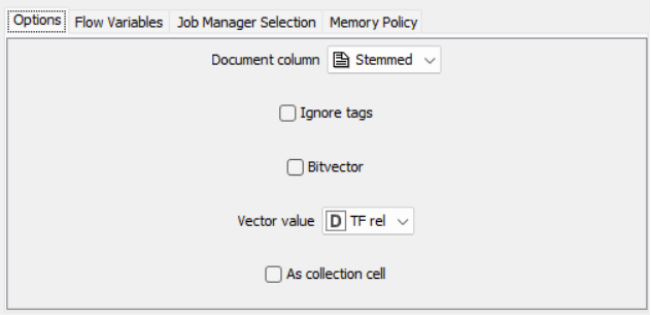
1. **TF**, Menghitung frekuensi relatif dari setiap istilah (term) berdasarkan dokumen, dan menambahkan kolom berisi nilai **tf**. Nilai **tf** dihitung dengan membagi frekuensi absolut istilah dalam dokumen dengan jumlah semua istilah dalam dokumen tersebut. TF ditarik dari node Bag of Words Creator. Dalam Form dialognya, pilih kolom *Stemmed* dan Relative Frequency (di tab TF options). Hasil dari TF diberikan dalam Gambar 1.92, pada kolom TF rel.

Table | Statistics

Term Term	TF rel Number (double)
Data[]	0.118
Science[]	0
:[]	0.059
is[]	0
a[]	0
field[]	0.029

Gambar 1.92 Hasil perhitungan TF

2. **Document Vector**, menghasilkan vektor dokumen dari kolom TF rel. Dalam form dialognya (Gambar 1.93), opsi yang diset antara lain nama kolom yang diproses, nilai vektor yang dirujuk yaitu TF rel. Opsi Bitvector tidak dicek.



Gambar 1.93. Opsi dalam Form dialog Document Vector

Hasilnya diberikan dalam Gambar 1.94, dimana setiap baris dokumen disajikan dalam bentuk vektor dokumen, yang tiap kolom kata-nya mewakili nilai TF dari kata tersebut.

Rows: 10 | Columns: 237

Table | Statistics

#	RowID	Document *Text document*	Data[] *Number (dou...*	Science[] *Number (dou...*	:[] *Number (dou...*	is[] *Number (dou...*	a[] *Number (dou...*	field[] *Number (dou...*	or[] *Number (dou...*	domain[] *Number (dou...*
1	Row0	"Data Scienc: ...	0.118	0	0.059	0	0	0.029	0	0.029
2	Row1	"Data Scienc: ...	0.039	0	0.039	0	0	0.02	0	0
3	Row2	"Data mine da...	0.042	0	0	0	0	0	0	0
4	Row3	"Data mine ex...	0.05	0	0	0	0	0	0	0
5	Row4	"Data mine pr...	0.024	0	0	0	0	0	0	0

Gambar 1.94 Hasil Document Vector