

DATA WAREHOUSING & DATA MINING

Edisi Revisi



Dr. Eng. Yusuf Sulistyo Nugroho, S.T., M.Eng.
Endang Wahyu Pamungkas, S.Kom., M.Kom., PhD.
Dimas Aryo Anggoro, S.Kom., M.Sc.
Helmi Imaduddin, S.Kom., M.Eng.
Wiwit Supriyanti S.Kom., M.Kom.



Data Warehousing

dan Data Mining

— Edisi Revisi —

Dr. Eng. Yusuf Sulistyo Nugroho, S.T., M.Eng.
Endang Wahyu Pamungkas, S.Kom., M.Kom., PhD.
Dimas Aryo Anggoro, S.Kom., M.Sc.
Helmi Imaduddin, S.Kom., M.Eng.
Wiwit Supriyanti, S.Kom., M.Kom.



Data Warehousing dan Data Mining

— Edisi Revisi —

Penulis:

Dr. Eng. Yusuf Sulistyo Nugroho, S.T., M.Eng.
Endang Wahyu Pamungkas, S.Kom., M.Kom., PhD.
Dimas Aryo Anggoro, S.Kom., M.Sc.
Helmi Imaduddin, S.Kom., M.Eng.
Wiwit Supriyanti, S.Kom., M.Kom.

Layouter dan Desain Cover

Amirul Ihsan

ISBN:

Cetakan 1, September 2022
©2022 Hak cipta pada penulis dilindungi undang-undang

Penerbit **Muhammadiyah University Press**
Universitas Muhammadiyah Surakarta
Gedung i Lantai 1
Jl. A Yani Pabelan Tromol Pos 1 Kartasura Surakarta 57102
Jawa Tengah - Indonesia
Telp : (0271) 717417 Eks. 2172
Web : mup.ums.ac.id
Email : muppress@ums.ac.id

Kata Pengantar

Alhamdulillah, puji dan syukur disampaikan kehadirat Allah SWT yang telah memberi kesempatan kepada penulis untuk menyusun buku “Praktikum Data Warehousing dan Data Mining” ini.

Buku praktikum ini mengalami beberapa pengembangan sejak versi pertama yang digunakan dalam mata kuliah “Data Warehousing dan Data Mining”. Buku ini mengalami perbaikan di sana-sini terkait kesalahan ketik dan update pengetahuan. Pada tahun 2016, ditambahkan dua bab baru yang bersifat opsional untuk praktikum terakhir. Pada tahun 2018 dilakukan penambahan materi tentang algoritma estimasi menggunakan regresi linier sederhana. Pada tahun 2022 dilakukan penambahan materi tentang principal component analysis.

Modul dalam buku ini dibagi menjadi dua bagian. Bagian pertama tentang pengenalan dan proses pengolahan data warehousing sedangkan pada bagian kedua tentang implementasi teknik data mining menggunakan bahasa pemrograman python.

Penulis berharap buku ini bermanfaat dan dapat digunakan secara maksimal dalam memahamkan mahasiswa tentang implementasi data warehousing dan data mining. Tak lupa pula penulis mengucapkan terima kasih kepada berbagai pihak yang membantu dalam penyelesaian buku ini, mulai dari pimpinan Program Studi Informatika, sejawat dosen terutama yang pengajar paralel matakuliah “Data Warehousing dan Data Mining”, asisten dan staf. Kritik dan saran sangat diharapkan demi penyempurnaan buku petunjuk praktikum ini.

Surakarta, Agustus 2022
Penulis

Daftar Isi

KATA PENGANTAR	iii
DAFTAR ISI.....	iv
MODUL 1 PERANCANGAN STAR SCHEMA DAN SNOWFLAKE.....	1
1.1. Tujuan.....	1
1.2. Landasan Teori.....	1
1.2.1. <i>Star Schema</i>	2
1.2.2. <i>Snowflake Schema</i>	4
1.3. Alat dan Bahan.....	9
1.4. Langkah-langkah Praktikum	9
1.5. Tugas.....	11
MODUL 2 DASAR ETL (EXTRACT TRANSFORM LOAD)	13
2.1. Tujuan.....	13
2.2. Landasan Teori.....	13
2.3. Alat dan Bahan.....	16
2.4. Langkah-langkah Praktikum	17
2.4.1. Ekstraksi Data Excel ke Pentaho.....	17
2.4.2. Transformasi Data	22
2.5. Tugas.....	32
MODUL 3 TABEL DIMENSI.....	33
3.1. Tujuan.....	33
3.2. Landasan Teori.....	33
3.3. Alat dan Bahan.....	34
3.4. Langkah-langkah Praktikum	34
3.4.1. Menghubungkan MySQL dengan Pentaho	35
3.4.2. Tabel Dimensi Karyawan	38
3.4.3. Tabel Dimensi Waktu.....	52
3.5. Tugas.....	64

MODUL 4 TABEL FAKTA.....	67
4.1. Tujuan.....	67
4.2. Landasan Teori.....	67
4.3. Alat dan Bahan.....	67
4.4. Langkah-langkah Praktikum	68
4.5. Tugas.....	83
 MODUL 5 PIVOT TABLE DAN CHART	 85
5.1. Tujuan.....	85
5.2. Landasan Teori.....	85
5.3. Alat dan Bahan.....	86
5.4. Langkah-langkah Praktikum	86
5.4.1. Membuat Pivot Table	86
5.4.2. Menambahkan Tipe Summary Baru	90
5.4.3. Calculated Field dan Calculated Item di Pivot Table.....	93
5.4.4. Operasi Roll Up dan Drill Down.....	98
5.4.5. Menggunakan Pivot Chart	101
5.5. Tugas.....	104
 MODUL 6 PENGENALAN APLIKASI DATA MINING	 107
6.1. Tujuan.....	107
6.2. Landasan Teori.....	107
6.2.1. Pengertian Data Mining.....	107
6.2.2. Manfaat Penggunaan Data Mining.....	108
6.2.3. Proses Data Mining	109
6.3. Alat dan Bahan.....	110
6.4. Pengenalan Perangkat Lunak Data Mining.....	111
6.4.1. Perangkat Lunak 1: Microsoft Excel.....	111
6.4.2. Perangkat Lunak 2: Weka.....	111
6.4.3. Perangkat Lunak 3: RapidMiner	112
6.5. Penggunaan Python untuk Data Mining	114
6.6. Langkah-langkah Praktikum	116
6.7. Tugas.....	123

MODUL 7 DATA PREPROCESSING	125
7.1. Tujuan.....	125
7.2. Landasan Teori.....	125
7.3. Alat dan Bahan.....	129
7.4. Langkah-langkah Praktikum	129
7.5. Tugas.....	145
 MODUL 8 ALGORITMA NAÏVE BAYES	 147
8.1. Tujuan.....	147
8.2. Landasan Teori.....	147
8.3. Alat dan Bahan.....	149
8.4. Langkah-langkah Praktikum	149
8.5. Tugas.....	158
 MODUL 9 ALGORITMA DECISION TREE (POHON KEPUTUSAN).....	 159
9.1. Tujuan.....	159
9.2. Landasan Teori.....	159
9.3. Alat dan Bahan.....	161
9.4. Langkah-langkah Praktikum	161
9.5. Tugas.....	172
 MODUL 10 REGRESI LINIER SEDERHANA	 173
10.1. Tujuan.....	173
10.2. Landasan Teori.....	173
10.3. Alat dan Bahan.....	175
10.4. Langkah-langkah Praktikum	175
10.5. Tugas.....	188
 MODUL 11 CLUSTERING: ALGORITMA K-MEANS	 191
11.1. Tujuan.....	191
11.2. Landasan Teori.....	191
11.3. Alat dan Bahan.....	193
11.4. Langkah-langkah Praktikum	193

11.5. Tugas.....	200
MODUL 12 INDUKSI DAN ATURAN ASOSIASI	203
12.1. Tujuan.....	203
12.2. Landasan Teori.....	203
12.3. Alat dan Bahan.....	205
12.4. Langkah-langkah Praktikum	205
12.4.1. Mengimport Library	205
12.4.2. Membaca Dataset.....	205
12.4.3. Mengkonversikan Dataframe ke dalam Array.....	206
12.4.4. Membuat Model Aturan Asosiasi	207
12.4.5. Mencetak Rules, Support, Confidence, dan Lift Ratio	208
12.5. Tugas.....	210
MODUL 13 PRINCIPAL COMPONENT ANALYSIS.....	211
13.1. Tujuan.....	211
13.2. Landasan Teori.....	211
13.3. Alat dan Bahan.....	213
13.4. Langkah-langkah Praktikum	213
13.4.1. Mengimport Library	215
13.4.2. Membaca Dataset.....	215
13.4.3. Menghapus Kolom yang Tidak Diperlukan.....	216
13.4.4. Menampilkan Jumlah Fitur	217
13.4.5. Implementasi PCA dengan Jumlah Fitur Awal	218
13.4.6. Menampilkan Hasil Variance pada Tiap <i>Principal Components</i>	218
13.4.7. Menampilkan Beberapa <i>Principal Component</i> Pertama Dengan Cumulative Explained Ratio Minimal 90%	219
13.4.8. Implementasi PCA dengan Jumlah Fitur yang Dikurangi....	220
13.5. Tugas.....	221

Modul 1

Perancangan Star Schema dan Snowflake

1.1 Tujuan

1. Mahasiswa mampu menjelaskan prosedur perancangan *Star Schema* atau *Snowflake*.
2. Mahasiswa mampu merancang *Star Schema* atau *Snowflake* menggunakan program aplikasi tertentu.

1.2 Landasan Teori

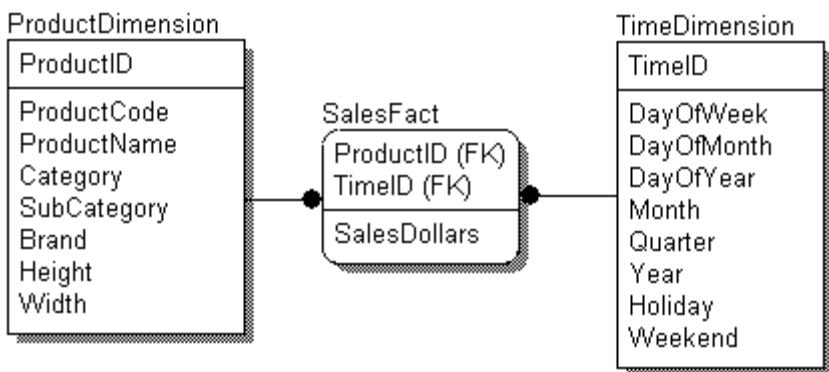
Apa hubungan antara *star schema* dan *snowflake* dengan basis data? Kedua istilah skema ini mewakili struktur basis data yang umum digunakan pada basis data OLAP (*On Line Analytical Processing*) untuk kebutuhan *data warehouse*. Dalam banyak pelajaran tentang basis data, kedua skema ini jarang disampaikan akibat penerapannya yang tidak sesuai untuk model basis data OLTP (*On Line Transactional Processing*). Mekanisme normalisasi juga tidak banyak berlaku untuk kedua jenis skema basis data ini. Fokus utama materi dasar-dasar basis data adalah proses manipulasi data, dalam hal ini bagaimana merancang sistem basis data yang dapat melayani sekian transaksi DML mulai *Insert*, *Update*, dan *Delete*? Bagaimana melakukan normalisasi pada struktur basis data untuk mendapatkan struktur yang ideal? Bagaimana mengatur transaksi antar klien agar tidak muncul *deadlock*? Dan banyak pertanyaan yang muncul terkait dengan sistem basis data.

Struktur data pada OLAP jauh lebih sederhana, mengingat data-data yang akan tersimpan di dalamnya tidak banyak mengalami perubahan dimana lebih banyak transaksi *selection* (*read only* – hanya baca) daripada DML. Jika pada OLTP, konsep ACID (*atomicity, consistency, isolation, durability*) menjadi properti utama yang harus melekat pada setiap transaksi data dari dan ke aplikasi klien maka dalam OLAP yang lebih diutamakan adalah kecepatan perolehan datanya (*data retrieval*). Tidak hanya struktur basis datanya yang berbeda, namun konfigurasi server basis datanya pun akan berbeda antara OLAP dan OLTP.

1.2.1. *Star Schema*

Dalam *data warehouse*, data-datanya akan disimpan dalam tabel fakta dan tabel dimensi. Tabel fakta akan menyimpan data-data utama sementara tabel dimensi mendeskripsikan setiap nilai dari suatu dimensi dan dapat direlasikan ke tabel fakta jika diperlukan. Data fakta merupakan data yang terukur besarnya, sebagai contoh adalah jumlah siswa, banyaknya rupiah yang diperoleh, rata-rata IPK, dan sejenisnya. Untuk lebih menjelaskan data fakta, maka kondisi saat data tersebut diukur turut disampaikan. Data kondisi inilah yang dipetakan dalam bentuk data dimensi. Kondisi yang dipetakan dalam dimensi umumnya berupa kondisi waktu, kondisi produk atau item, dan kondisi geografisnya. Mendesain struktur *star schema*, dimulai dengan menentukan data apa yang ingin dilihat oleh pengguna (besarannya) dan bagaimana pengguna melihat data tersebut (kondisi atau dimensinya).

Tabel dimensi memiliki *primary key* sederhana yang mengandung hanya satu atau dua kolom saja. Namun, tabel fakta akan memiliki sekumpulan *foreign key* yang disusun dari *primary key* komposit dan merupakan gabungan kolom-kolom tabel dimensi yang berelasi. Untuk lebih jelasnya, berikut contoh struktur *star schema*.



Gambar 1.1 Contoh Star Schema

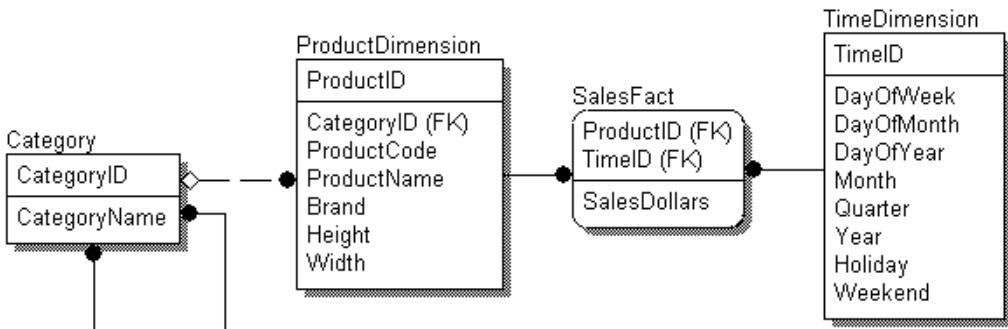
Untuk struktur *star schema* seperti gambar 1.1, data dalam tabel fakta yang diukur adalah hasil penjualan (dalam mata uang dollar) berdasarkan dimensi atau kondisi produk yang dijual (*product*) serta waktu penjualan (*time*). Misalkan dimensi produk, yang menyimpan informasi-informasi seputar produk. Produk ini dapat dikelompokkan ke dalam kategori, dan di dalam kategori inipun bisa ditemukan sub-kategori. Misalkan dalam sebuah basis data terdapat kode produk X1001 yang merujuk pada kripik tempe, maka akan masuk ke dalam kategori Nabati, dan sub-kategori Tempe. Untuk lebih mengelompokkan produk tersebut, dapat pula dibuatkan sub-kategori berikutnya. Namun kunci dari informasi produk tersebut tersimpan dalam kolom di tabel dimensi, dan tidak dibutuhkan tabel lain untuk menjelaskan detil produk. Semakin beragam jenis kondisi data yang ingin diamati, maka akan semakin besar ukuran tabel fakta yang dimuat.

Dalam *star schema*, query yang terbentuk antara tabel fakta dan sejumlah tabel dimensi dinamakan *star query*. Setiap tabel dimensi direlasikan dengan tabel fakta berdasarkan kolom *primary key* dan *foreign key*, namun diantara masing-masing tabel dimensi tidak ada yang saling berelasi (tidak ada hubungan data). Query yang terbentuk menyebabkan proses eksekusi yang lebih optimal, karena rencana eksekusi query dalam DBMS akan lebih cepat dengan setiap tabel hanya berelasi dengan satu tabel yang lain.

Ada kalanya tabel dimensi mengandung data yang duplikat pada satu atau lebih kolom. Jika mengikuti azas normalisasi, maka struktur basis data yang terbentuk bukan lagi *star schema* namun akan menjadi *snowflake schema*.

1.2.2. Snowflake Schema

Struktur basis data ini lebih kompleks dari pada star schema, dengan menormalisasi tabel-tabel dimensi yang berukuran besar dengan satu atau lebih kolom yang memiliki duplikasi data. Misalkan jika tabel dimensi Product dinormalisasi maka akan menghasilkan struktur seperti berikut:



Gambar 1.2 Contoh Bentuk Snowflake

Tabel dimensi dinormalisasi untuk mengurangi redundansi data (duplikasi), sehingga strukturnya akan lebih ramping. Dengan pengelompokan ini, data akan lebih mudah dibaca dan membantu pengembang aplikasi untuk menata desain antarmuka sistem dan filtering data. Struktur ini akan menghemat kapasitas storage, namun waktu eksekusi data akan lebih lama mengingat jumlah tabel dimensi yang direlasikan lebih banyak dan membutuhkan tambahan relasi foreign key. Query yang terbentuk lebih kompleks, yang mengakibatkan kinerja query menurun. Pada penerapan yang lebih umum, tabel dimensi tidak diturunkan dengan lebih banyak tabel dimensi lain dan pengelompokan data diatur secara hard-coded di kode program aplikasinya.

Fokus penggunaan datawarehouse adalah kecepatan akses dan eksekusi data, bukanlah ukuran data yang lebih kecil atau struktur basis data yang lebih ramping. Sehingga bijaksana dalam menetapkan struktur data star maupun snowflake schema akan menentukan kinerja layanan datawarehouse yang dimiliki.

Tahap pertama dari perancangan data warehouse adalah mendefinisikan informasi-informasi apa saja yang dibutuhkan oleh manajemen. Agar kebutuhan ini dapat didefinisikan dengan tepat, maka pemahaman akan peran dan tugas manajemen yang membutuhkan informasi tersebut mutlak harus dilakukan lebih dulu. Jika sudah dipahami, selanjutnya kita hanya tinggal “menjawab” pertanyaan-pertanyaan berikut:

1. Siapa yang membutuhkan informasi dari data warehouse?
2. Informasi apa saja yang dibutuhkan tersebut?
3. Seperti apa layout dan isi informasi-informasi itu?
4. Kapan informasi tersebut digunakan?
5. Untuk keperluan apa?
6. Basis data apa yang menjadi sumber untuk informasi tersebut?

Sebagai contoh, misalkan akan dibuat sebuah data warehouse penjualan (atau data mart penjualan tepatnya) untuk sebuah perusahaan dagang.

1. Siapa yang membutuhkan informasi dari data warehouse?
Manager Pemasaran
2. Informasi apa saja yang dibutuhkan Manager Pemasaran?
Barang apa yang paling banyak terjual di lokasi tertentu sepanjang tahun?
Barang apa yang paling banyak memberikan pendapatan sepanjang tahun?
3. Seperti apa layout dan isi informasi-informasi itu?
Barang yang paling banyak terjual di lokasi tertentu sepanjang tahun:

tahun	kecamatan	kategori	sum(total_penjualan)
2012	BANJARSARI	KONSUMSI	209
2012	JEBRES	ATK	95
2012	LAWEYAN	ATK	109
2012	SERENGAN	ATK	89
2012	JEBRES	KONSUMSI	106
2012	PASAR KLIWON	KONSUMSI	96
2012	BANJARSARI	ATK	200
2012	LAWEYAN	KONSUMSI	193
2012	PASAR KLIWON	ATK	91
2012	SERENGAN	KONSUMSI	139

Barang yang paling banyak memberikan pendapatan sepanjang tahun:

tahun	kategori	sub_kategori	sum(total_penerimaan)
2012	ATK	KERTAS	3560000
2012	ATK	PULPEN	472000
2012	ATK	SPIDOL	1269000
2012	KONSUMSI	SEMBAKO	524000
2012	KONSUMSI	SNACK	1669500

4. Untuk keperluan apa informasi tersebut?
Dasar untuk menentukan strategi penjualan barang
5. Kapan informasi tersebut digunakan?
Awal periode penjualan
6. Basis data apa yang menjadi sumber untuk informasi tersebut?
Basis data penjualan dengan skema sebagai berikut:
 - a) Kategori (#kelompok, sub_kategori, kategori)
 - b) Barang (#kode_barang, nama_barang, #kelompok, satuan, harga)
 - c) Lokasi (#kode_pos, kelurahan, kecamatan)

- d) Pelanggan (#kode_pelanggan, nama_pelanggan, alamat, kota, #kode_pos, telepon)
- e) Penjualan (#no_faktur, #kode_barang, jumlah)
- f) Pembayaran (#no_faktur, tanggal, total, diskon, #kode_pelanggan)

Tahap berikutnya yang harus dilakukan adalah menentukan *measure* dan *dimension* untuk semua informasi yang dibutuhkan manajemen. *Measure* adalah data numerik yang akan dicari jejak nilainya, sedangkan *dimension* adalah parameter atau sudut pandang terhadap *measure* sehingga dapat mendefinisikan suatu transaksi.

Sebagai contoh, untuk informasi “barang yang paling banyak terjual di lokasi tertentu sepanjang tahun”,

- 1. *Measure*: total penjualan
- 2. *Dimension*: barang, tahun (waktu/periode), lokasi

Sedangkan untuk informasi “barang yang paling banyak memberikan pendapatan sepanjang tahun”,

- 1. *Measure*: total pendapatan
- 2. *Dimension*: barang, tahun (waktu/periode)

Dimension mempunyai hirarki. Penentuan hirarki untuk *dimension* ini sepenuhnya tergantung kepada proses *drill down* dan *roll up* yang ingin dilakukan saat melakukan OLAP (*On Line Analytical Processing*) nanti.

Untuk contoh di atas, hirarki masing-masing *dimension* adalah:

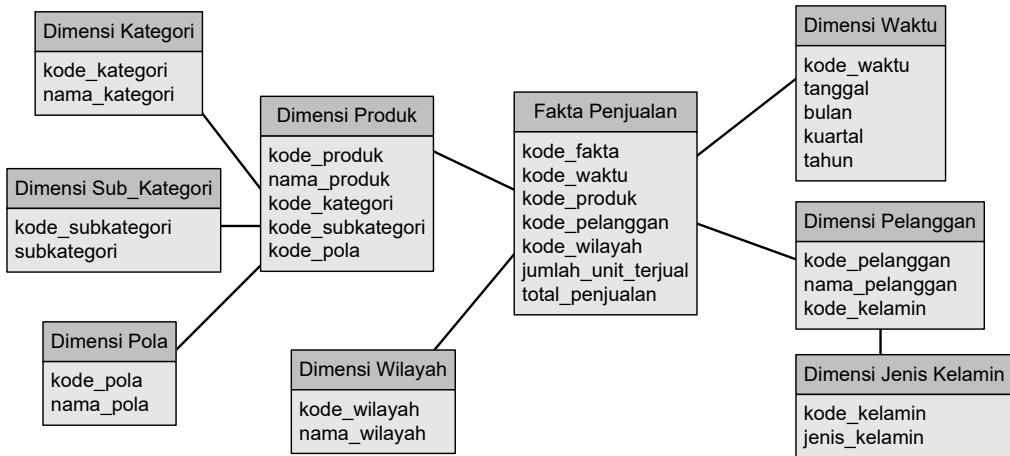
- 1. Barang: nama barang, sub-kategori, kategori
- 2. Periode: minggu, bulan, tahun
- 3. Lokasi: kelurahan, kecamatan, kota

Sedangkan *layout* dan isi informasinya dapat ditunjukkan oleh tabel berikut ini:

BARANG				PERIODE			LOKASI			TOTAL PENJUALAN	TOTAL PENERIMAAN			
KODE	NAMA	SUB KATEGORI	KATEGORI	MINGGU	BULAN	TAHUN	KELURAHAN	KECAMATAN	KOTA					
SP-001	BOARDMARKER SNOWMAN	SPIDOL	ATK	33	8	2012	JEBRES	JEBRES	SURAKARTA	12	Rp 78,000			
				38	9	2012	GAJAHAN	PASAR KLIWON	SURAKARTA	7	Rp 45,500			
				43	10	2012	KAUMAN	PASAR KLIWON	SURAKARTA	7	Rp 45,500			
				7	2	2012	GILINGAN	BANJARSARI	SURAKARTA	12	Rp 78,000			
										38	Rp 247,000			
PL-002	PULPEN PILOT	PULPEN	ATK	22	6	2012	DANUKUSUMAN	SERENGAN	SURAKARTA	3	Rp 9,000			
				26	6	2012	KARANGASEM	LAWEYAN	SURAKARTA	6	Rp 18,000			
				41	10	2012	KERTEN	LAWEYAN	SURAKARTA	8	Rp 24,000			
										17	Rp 51,000			
KH-005	A4 SINAR DUNIA	KERTAS	ATK	10	3	2012	MOJOSONGO	JEBRES	SURAKARTA	10	Rp 350,000			
				27	7	2012	TIPE	SERENGAN	SURAKARTA	4	Rp 140,000			
				46	11	2012	MANAHAN	BANJARSARI	SURAKARTA	9	Rp 315,000			
										23	Rp 805,000			
JUMLAH											78	Rp1,103,000		

Perancangan model konseptual data warehouse adalah tahap berikutnya yang harus dilaksanakan setelah tahap penentuan *measure* dan *dimension*. Pada tahap ini dibuat suatu model yang dapat menggambarkan data atau tabel apa saja yang akan disimpan dalam data warehouse, berikut keterhubungan diantaranya.

Data atau tabel dalam data warehouse tersebut dapat dimodelkan dengan menggunakan alat bantu pemodelan seperti E-R diagram, *star schema*, *snowflake schema*, atau FCO-IM (*Fully Communication Oriented Information Modelling*). Tetapi pada umumnya alat bantu yang digunakan adalah *star schema* atau *snowflake schema*. *Star schema* digunakan untuk menggambarkan *fact table*, yaitu tabel yang merepresentasikan *measure*, sebagai “pusat data”. Tabel ini nantinya akan terkoneksi dengan tabel-tabel yang mendeskripsikan dimensi untuk *measure* tersebut (*dimension table*). Sebagai contoh, *snowflake schema* untuk data warehouse penjualan di sebuah perusahaan batik adalah:



Gambar 1.3 Snowflake Schema Penjualan

1.3) Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi DBDesigner.
3. Modul Praktikum Data Warehousing dan Data Mining.

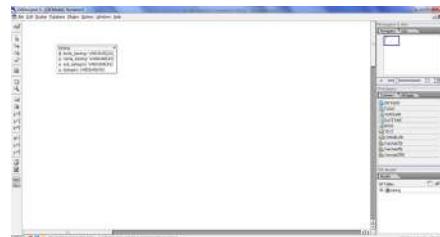
1.4) Langkah-langkah Praktikum

Menggambar *Star Schema* dengan menggunakan DB Designer :

1. Jalankan program aplikasi DB Designer untuk membuat desain *star schema*.
2. Klik button *new table* kemudian klik pada area kerja sehingga akan menghasilkan tabel baru.
3. Double klik pada tabel baru untuk membuka tabel editor, ganti nama pada *table name* dengan nama **barang**, kemudian isikan atribut tabel dengan data sebagai berikut :

Column Name	Data Type
kode_barang	Varchar(20)
nama_barang	Varchar(45)
sub_kategori	Varchar(45)
kategori	Varchar(45)

4. Klik pada *column name* kode_barang untuk mengatur kode_barang sebagai *primary key* sehingga berubah menjadi .
5. Klik untuk menutup *table editor* sehingga tabel barang menjadi:

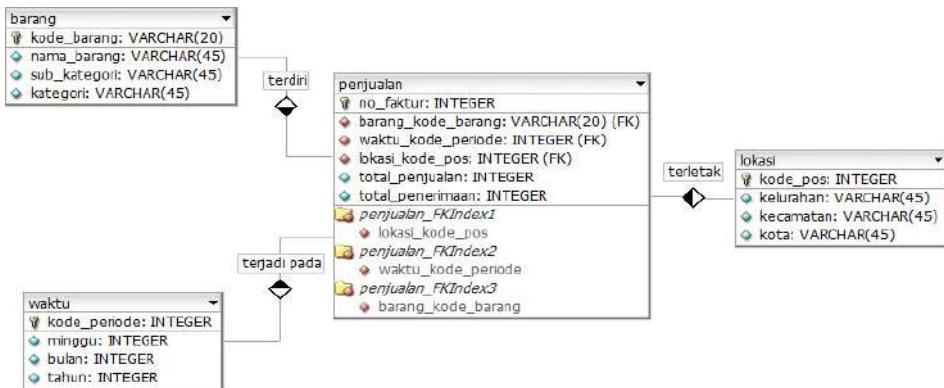


6. Ulangi kembali langkah 2 sampai 5 untuk membuat tabel **waktu**, **lokasi** dan **penjualan**.
7. Setelah semua tabel dibuat, hubungkan setiap tabel dengan tabel lain dengan button sebagai berikut :

Button	Fungsi Relationship
	1:n (<i>one to many</i>)
	1:1 (<i>one to one</i>)
	n:m (<i>many to many</i>)

Keterangan : klik salah satu button yang sesuai dengan kebutuhan kemudian klik tabel yang akan dihubungkan.

8. Ubah nama *relationship* dengan membuka *relationship editor*, sehingga setelah selesai hasil akhir manjadi seperti berikut :

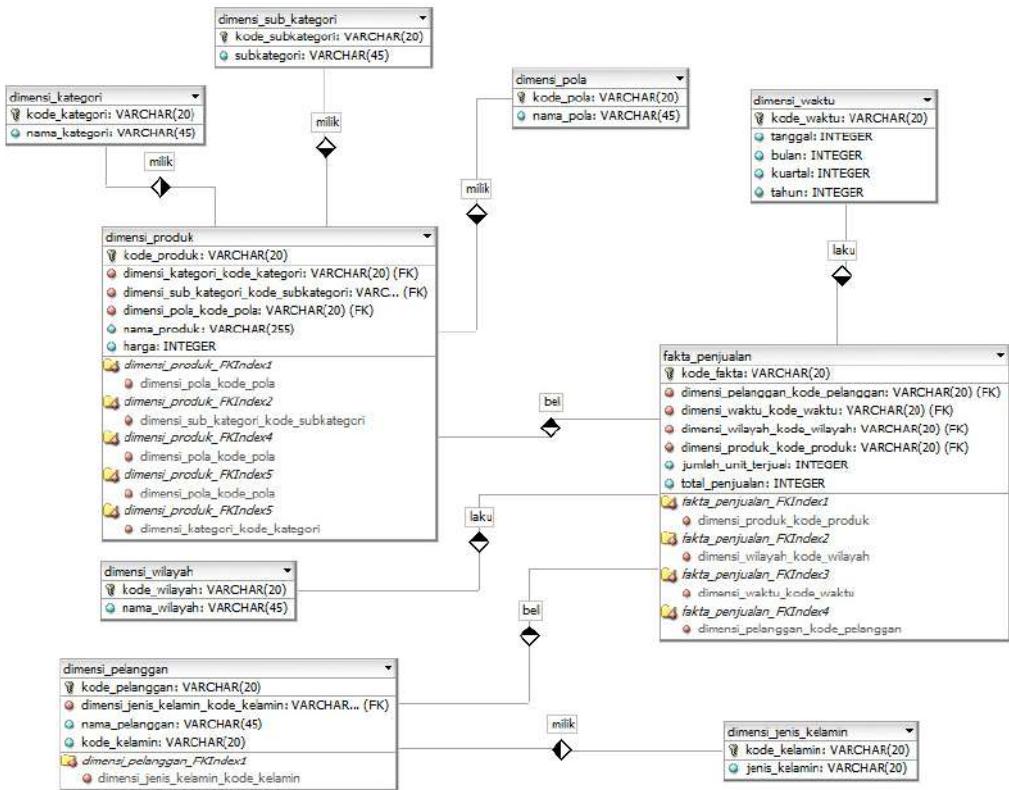


9. Simpan dengan nama file “**star schema penjualan.xml**” dalam folder “Praktikum Data Warehousing dan Data Mining”

Snowflake schema merupakan perbaikan dari *star schema*, sehingga cara penggambarannya pun mirip. Bedanya, satu atau beberapa hierarki yang ada pada *dimension table* dinormalisasi (dekomposisi) menjadi beberapa tabel yang lebih kecil.

1.5 Tugas

Rancanglah diagram *Snowflake schema* berdasarkan gambar di bawah dengan menggunakan DBDesigner seperti gambar berikut! Simpan dengan nama file “**snowflake penjualan.xml**” ke dalam folder “Praktikum Data Warehouse”



Modul 2

Dasar ETL (Extract Transform Load)

2.1 Tujuan

Mahasiswa mampu melakukan proses ekspor dan impor data yang merupakan bagian dalam proses ETL sebuah pengembangan *data warehouse*.

2.2 Landasan Teori

Ekstraksi (*extraction*) adalah proses pengambilan data dari sumber data dimana proses pengambilan data ini tidak mengambil keseluruhan data yang ada di *database* operasional, melainkan hanya mengambil data-data matang saja. Tahapan ini adalah yang paling pertama dalam proses ETL. Setelah ekstraksi, data ini akan ditransformasikan dan *di-load* ke dalam *data warehouse*.

Pendesainan dan pembuatan proses ekstraksi adalah satu kegiatan yang paling sering menyita waktu di dalam proses ETL dan dalam keseluruhan proses *data warehouse*. Data di-ekstrak tidak hanya sekali namun beberapa kali dalam suatu periode untuk mensuplai data ke dalam *data warehouse* dan menjaga agar *up-to-date*. Lebih jauh lagi, sistem sumber tidak dapat dimodifikasi atau bahkan kinerja dan ketersediaannya tidak dapat diatur untuk mengakomodasi kebutuhan proses ekstraksi *data warehouse*.

Ada dua bentuk metode ekstraksi *logical*:

1. Ekstraksi Statis (*Static Extraction*)

Data di-ekstrak secara lengkap dari sistem sumber. Ekstraksi ini melibatkan seluruh data yang sedang tersedia dalam sistem sumber. Data sumber disediakan dan tidak dibutuhkan logika informasi tambahan (seperti *timestamp*) yang dibutuhkan pada situs sumber. Sebuah contoh ekstraksi penuh adalah ekspor file dari sebuah tabel yang berbeda atau kueri remote SQL yang membaca sumber data lengkap. Proses ekstrak ini biasanya hanya dilakukan sekali di awal proses.

2. Ekstraksi Inkremental (*Incremental Extraction*)

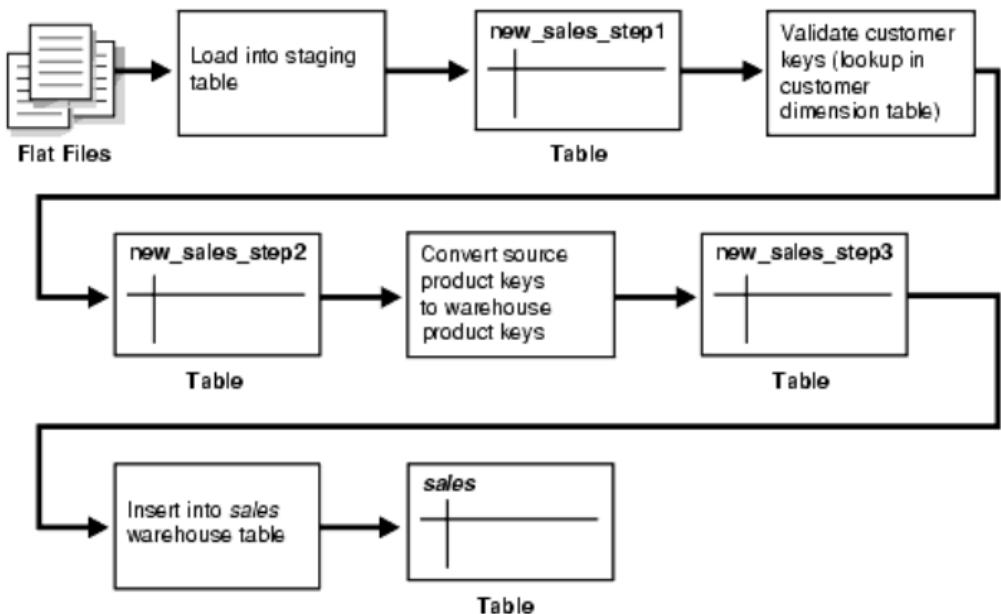
Pada poin waktu tertentu, hanya data yang memiliki histori data atau mengalami perubahan yang akan di-ekstrak. *Event* ini adalah proses ekstraksi yang dilakukan paling akhir atau sebagai contoh sebuah *event* bisnis yang kompleks seperti hari *booking* terakhir dari suatu periode fiskal. Informasi ini juga dapat disediakan oleh data sumber itu sendiri seperti sebuah kolom aplikasi, merefleksikan *timestamp* yang paling akhir berubah atau sebuah tabel yang berubah dimana sebuah mekanisme tambahan yang sesuai menjaga *track* perubahan selain transaksi yang permulaan.

Transformasi data sering kali sangat kompleks, dalam hal waktu proses, bagian proses ekstraksi, transformasi dan loading yang paling membutuhkan banyak biaya. Proses ini boleh jadi merentang dari konversi data sederhana hingga teknik pengumpulan data kompleks yang ekstrim.

Dari perspektif arsitektural, data dapat ditransformasikan dengan 2 cara:

1. *Multistage Data Transformation*

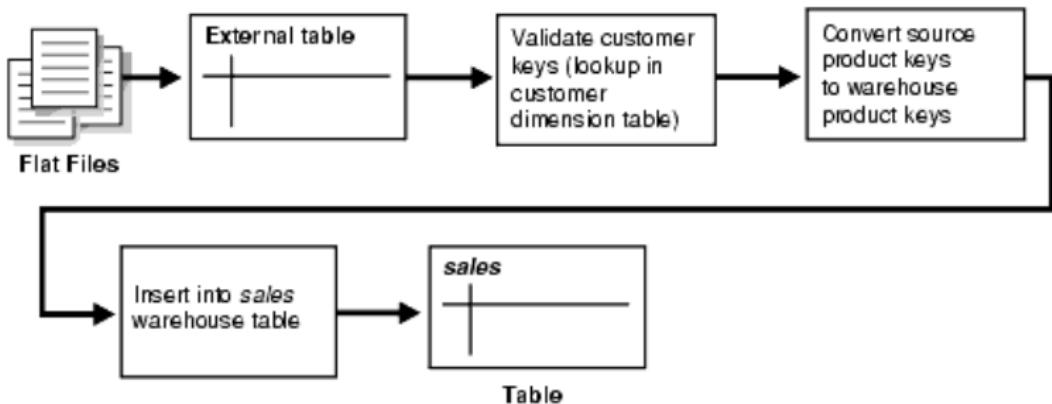
Logika transformasi data bagi kebanyakan data warehouse terdiri dari beberapa tahapan. Sebagai contoh, dalam transformasi *record* baru yang dimasukkan ke dalam sebuah tabel penjualan (*sales*), boleh jadi terdapat tahapan transformasi *logic* yang terpisah untuk memvalidasi masing-masing *key* dimensi. Gambaran secara grafis dari proses *transformation logic* adalah sebagai berikut :



Gambar 2.1 Transformasi Data Multistage

2. Pipelined Data Transformation

Arus proses ETL dapat diubah secara dramatis dan database menjadi sebuah bagian integral solusi ETL. Fungsionalitas barunya melukiskan beberapa pembentukan tahapan proses penting yang kuno ketika beberapa yang lainnya dapat dimodelkan kembali untuk menambah arus data dan transformasi data menjadi lebih dapat diukur. Kegiatannya bergeser dari transformasi serial hingga proses *load* (dengan kebanyakan kegiatan dilakukan di luar *database*) atau *load* kemudian proses transformasi untuk meningkatkan transformasi ketika proses *loading*.



Gambar 2.2 Gambar Transformasi Pipelined Data

Pentaho Data Integration (PDI) atau Kettle adalah software dari Pentaho yang dapat digunakan untuk proses ETL (*Extraction, Transformation* dan *Loading*). PDI dapat digunakan untuk migrasi data, membersihkan data, loading dari file ke *database* atau sebaliknya dalam volume besar. PDI menyediakan *graphical user interface* dan *drag-drop* komponen yang memudahkan *user*.

Elemen utama dari PDI adalah Transformation dan Job. Transformation adalah sekumpulan instruksi untuk merubah input menjadi output yang diinginkan (*input – proses – output*). Sedangkan Job adalah kumpulan instruksi untuk menjalankan transformasi.

Ada tiga komponen dalam PDI: Spoon, Pan dan Kitchen. Spoon adalah *user interface* untuk membuat Job dan Transformation. Pan adalah tools yang berfungsi membaca, merubah dan menulis data. Sedangkan Kitchen adalah program yang mengeksekusi Job.

2.3 Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi Ms. Office.
3. Program aplikasi Pentaho Data Integration.

4. Modul Praktikum Data Warehousing dan Data Mining.
5. Dataset yang bisa diunduh pada https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM

2.4 Langkah-langkah Praktikum

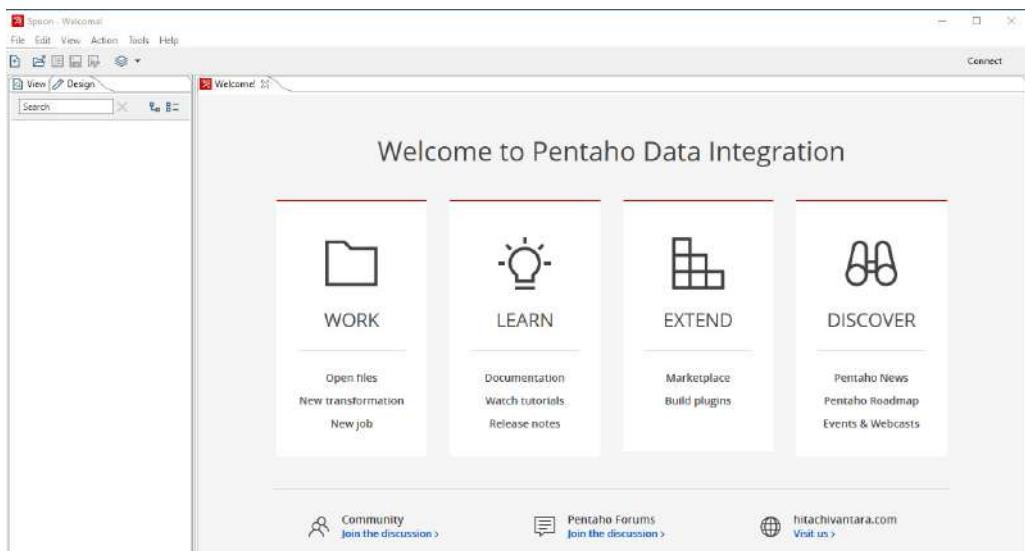
2.4.1 Ekstraksi Data Excel ke Pentaho

1. Buka program aplikasi **Ms. Excel** dan siapkan data seperti pada **Tabel Target Penjualan** di bawah ini. Anda juga bisa unduh pada URL berikut: https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM/src/branch/master/Data/ETL/Target%20Penjualan%20New.xlsx.

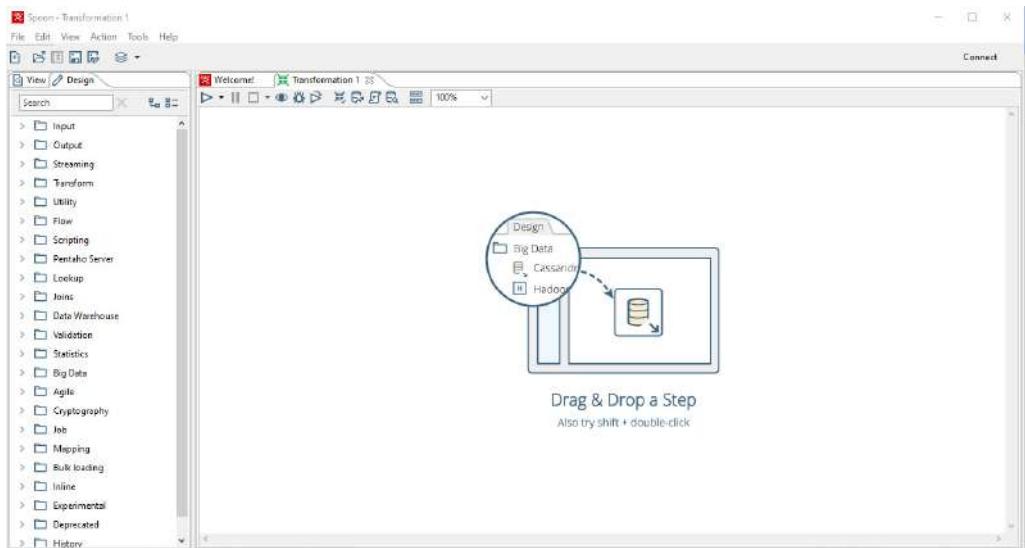
Tabel Target Penjualan

Kode Cabang	Kode Kategori	Kode Produk	Tahun	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
CABANG-555	PROD-0000001	PROD-0000001	2008	3264	2965	5538	4248	3050	4823	3375	3922	3783	4154	3173	4058
CABANG-039	PROD-0000003	PROD-0000003	2009	1906	1580	2033	1789	2405	2435	2084	2767	2828	2469	2528	2235
CABANG-039	PROD-0000004	PROD-0000004	2009	3554	2800	2506	2864	3340	3302	3172	2120	2054	2321	3200	3449
CABANG-039	PROD-0000005	PROD-0000005	2008	2667	2002	2021	3439	2169	3051	3159	2040	2238	2249	1932	1822
CABANG-039	PROD-0000006	PROD-0000006	2008	3940	3600	3282	2407	3773	2341	3742	2636	2859	3091	2317	4374
CABANG-039	PROD-0000007	PROD-0000007	2008	1670	1124	1763	1567	1609	1605	2026	1277	1173	1514	1725	1867
CABANG-039	PROD-0000008	PROD-0000008	2008	452	375	380	282	287	369	490	343	293	348	345	458
CABANG-039	PROD-0000009	PROD-0000009	2008	1353	1791	1996	2008	1466	1881	1939	2293	1220	1377	1262	1546
CABANG-039	PROD-0000010	PROD-0000010	2008	308	455	487	521	506	441	520	359	431	536	321	389
CABANG-039	PROD-0000011	PROD-0000011	2008	521	380	570	377	292	516	347	429	516	392	427	491
CABANG-039	PROD-0000012	PROD-0000012	2008	1005	1047	1628	1071	1467	1183	903	1351	1199	1305	1319	1363
CABANG-039	PROD-0000013	PROD-0000013	2008	3777	5230	3465	5721	5330	5617	4402	4350	5907	4372	5754	5280
CABANG-039	PROD-0000014	PROD-0000014	2008	2976	4216	4554	3105	4049	4243	3622	4030	5725	5603	3527	4656
CABANG-039	PROD-0000015	PROD-0000015	2008	9223	8017	5016	7549	5629	8474	6421	7097	7395	7864	7092	4807
CABANG-039	PROD-0000016	PROD-0000016	2008	42392	40194	46822	27986	24826	36440	34105	43519	36122	41572	31706	24928
CABANG-039	PROD-0000017	PROD-0000017	2008	533	343	465	528	379	524	411	445	382	294	594	629
CABANG-039	PROD-0000018	PROD-0000018	2008	4045	2525	3722	3236	2930	2784	4338	3413	4343	3666	4440	4364
CABANG-039	PROD-0000019	PROD-0000019	2008	1674	2384	2176	1653	2734	1863	1714	1753	1912	2324	3077	1936
CABANG-039	PROD-0000020	PROD-0000020	2008	1881	1698	1722	1330	2007	2147	2042	1480	1545	2332	2684	2329
CABANG-039	PROD-0000021	PROD-0000021	2008	4430	4889	3743	3536	4923	3172	5059	4787	3855	3888	3235	3597
CABANG-039	PROD-0000022	PROD-0000022	2008	31227	19207	26032	27068	33273	27687	31234	37479	29440	36294	31996	35864
CABANG-039	PROD-0000023	PROD-0000023	2008	5151	5229	4500	3906	4795	4036	6414	5240	4303	6353	5265	4573

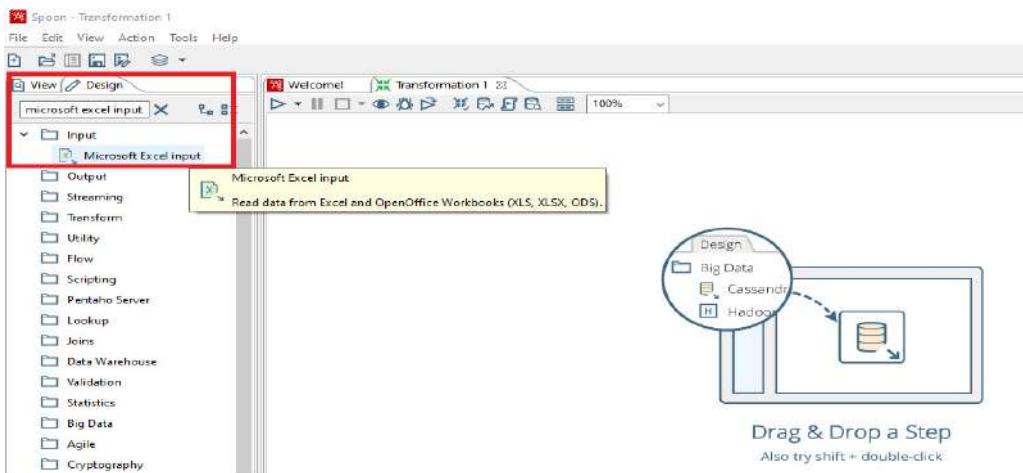
2. Jalankan aplikasi Pentaho Data Integration dengan mengaktifkan file **Spoon.bat** (klik kanan >> Run as Administrator), maka akan muncul tampilan awal seperti gambar di bawah.



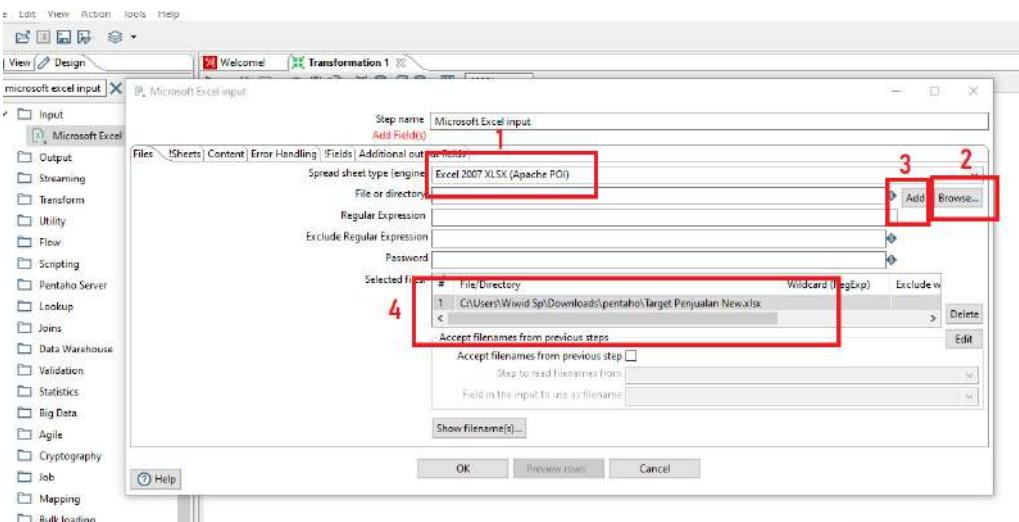
3. Buat transformasi baru dengan **File >> New >> Transformation (CTRL+N)**.



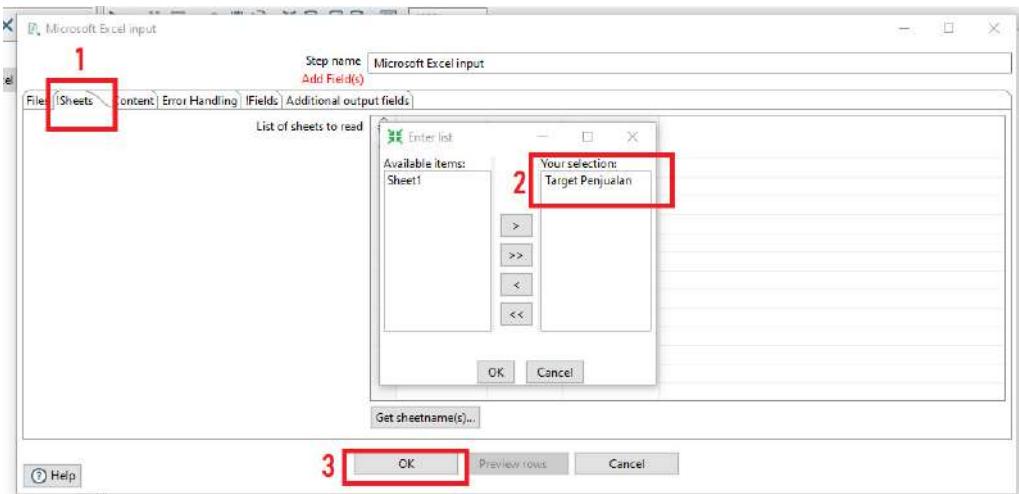
- Pada sidebar sebelah kiri pilih tab **Design**, ketikkan **Microsoft Excel input** pada kolom **Search** dan tarik ke canvas.



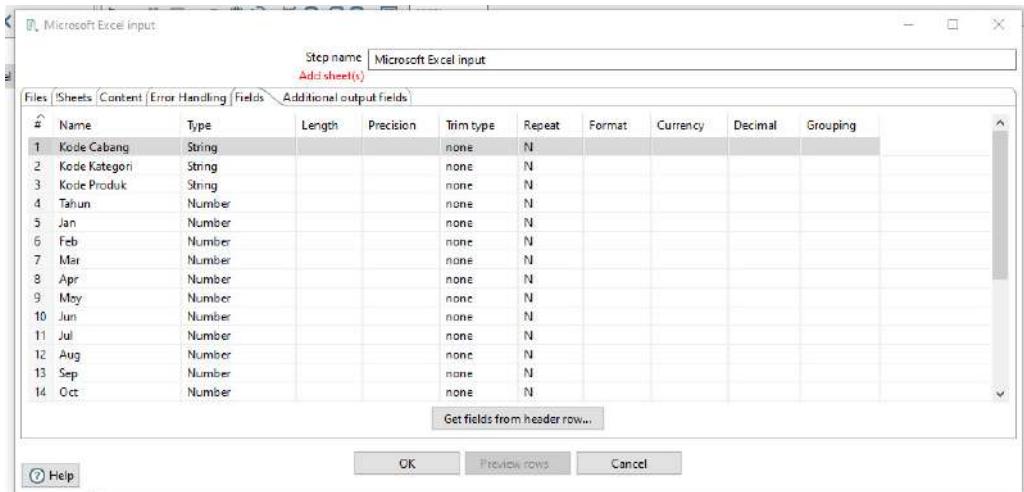
- Lakukan konfigurasi pada step **Microsoft Excel input** dengan cara klik kanan >> **Edit**. Pada tab **Files** ganti **Spread sheet type** menjadi Excel 2007, lalu pilih file excel yang ingin di-ekstrak pada **File or directory** dan tekan tombol **Add**. Pastikan file excel tersebut ada di tabel **Selected Files**.



6. Pada tab **!Sheets**, tekan tombol **Get sheetname(s)...** untuk memilih sheet pada file excel yang ingin di-ekstrak datanya. Pilih sheet pada **Available items** dan pindahkan ke kotak **Your selection**. Tujuannya agar sheet yang kita pilih sesuai, umumnya hal ini akan berpengaruh apabila pada file excel tersebut terdapat lebih dari satu sheet.



7. Pada tab **!Fields**, tekan tombol **Get fields from header row...** untuk mendapatkan semua kolom yang ada pada sheet excel yang kita pilih.



8. Tekan tombol **Preview rows** untuk memastikan apakah data yang akan di-ekstrak sudah benar. Jika sudah benar, maka tekan tombol **OK**.

Excel preview data

Rows of step Microsoft Excel input (122 rows)

#	Kode Cabang	Kode Kategori	Kode Produk	Tahun	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1.	CABANG-555	PKD0-0000001	PKD0-00000001	2000.0	3264.0	2655.0	5538.0	4248.0	3050.0	4822.0	3175.0	3622.0	3793.0	4154.0	3172.0	4050.0
2.	CABANG-099	PKD0-0000002	PKD0-00000002	2000.0	1806.0	1580.0	1030.0	1789.0	2405.0	3425.0	2084.0	2767.0	2828.0	3468.0	2528.0	2235.0
3.	CABANG-099	PKD0-0000004	PKD0-00000004	2000.0	3154.0	2806.0	1506.0	2864.0	3340.0	3020.0	3172.0	2120.0	2054.0	2321.0	3206.0	3446.0
4.	CABANG-099	PKD0-0000005	PKD0-00000005	2000.0	2671.0	2002.0	2021.0	3419.0	2169.0	3091.0	3159.0	2040.0	2238.0	2249.0	1932.0	1822.0
5.	CABANG-099	PKD0-0000006	PKD0-00000006	2000.0	3940.0	3600.0	3252.0	3407.0	3772.0	2241.0	3742.0	2636.0	2859.0	3091.0	2517.0	4174.0
6.	CABANG-099	PKD0-0000007	PKD0-00000007	2000.0	1870.0	1124.0	1781.0	1587.0	1639.0	1405.0	2024.0	1277.0	1173.0	1514.0	1725.0	1807.0
7.	CABANG-099	PKD0-0000008	PKD0-00000008	2000.0	452.0	375.0	380.0	282.0	267.0	380.0	345.0	293.0	346.0	345.0	476.0	
8.	CABANG-099	PKD0-0000009	PKD0-00000009	2000.0	1353.0	179.0	1966.0	2000.0	1566.0	1884.0	1599.0	2200.0	1200.0	1262.0	1546.0	
9.	CABANG-099	PKD0-0000010	PKD0-0000010	2000.0	204.0	451.0	467.0	521.0	506.0	441.0	520.0	259.0	421.0	516.0	241.0	480.0
10.	CABANG-099	PKD0-0000011	PKD0-0000011	2000.0	511.0	880.0	570.0	377.0	262.0	516.0	347.0	426.0	316.0	892.0	421.0	491.0
11.	CABANG-099	PKD0-0000012	PKD0-0000012	2000.0	1005.0	1047.0	1626.0	1071.0	1467.0	1183.0	903.0	1351.0	1199.0	1395.0	1315.0	1365.0
12.	CABANG-099	PKD0-0000013	PKD0-0000013	2000.0	3777.0	5236.0	3465.0	5721.0	5330.0	5617.0	4461.0	4950.0	5907.0	4372.0	5754.0	5280.0
13.	CABANG-099	PKD0-0000014	PKD0-0000014	2000.0	2976.0	4216.0	4554.0	3165.0	4049.0	4243.0	3622.0	4050.0	5725.0	5660.0	3527.0	4655.0
14.	CABANG-099	PKD0-0000015	PKD0-0000015	2000.0	923.0	801.0	3016.0	7549.0	5629.0	8474.0	6421.0	7007.0	7393.0	7884.0	7052.0	4807.0
15.	CABANG-099	PKD0-0000016	PKD0-0000016	2000.0	42392.0	40794.0	48822.0	27896.0	24826.0	36440.0	34165.0	43515.0	36122.0	41572.0	31706.0	24926.0
16.	CABANG-099	PKD0-0000017	PKD0-0000017	2000.0	533.0	343.0	485.0	538.0	379.0	524.0	411.0	445.0	382.0	294.0	594.0	629.0
17.	CABANG-099	PKD0-0000018	PKD0-0000018	2000.0	4045.0	2521.0	3722.0	3216.0	2930.0	2794.0	4328.0	3413.0	4343.0	3666.0	4440.0	4364.0
18.	CABANG-099	PKD0-0000019	PKD0-0000019	2000.0	1674.0	2384.0	2176.0	1638.0	2734.0	1868.0	1114.0	1758.0	1912.0	2334.0	3077.0	1988.0
19.	CABANG-099	PKD0-0000020	PKD0-0000020	2000.0	1818.0	1698.0	1722.0	1310.0	2007.0	2147.0	2042.0	1403.0	1545.0	2332.0	2584.0	2329.0
20.	CABANG-099	PKD0-0000021	PKD0-0000021	2000.0	4430.0	4893.0	3743.0	3516.0	4823.0	3172.0	5058.0	4787.0	3855.0	3868.0	3235.0	3597.0
21.	CABANG-099	PKD0-0000022	PKD0-0000022	2000.0	31227.0	19207.0	26032.0	27648.0	32272.0	27678.0	31234.0	37479.0	29440.0	36294.0	21996.0	35646.0
22.	CABANG-099	PKD0-0000023	PKD0-0000023	2000.0	5113.0	3226.0	4300.0	3908.0	4795.0	4080.0	6414.0	5240.0	4383.0	6533.0	5255.0	4573.0
23.	CABANG-099	PKD0-0000024	PKD0-0000024	2000.0	5457.0	4965.0	4276.0	3950.0	4902.0	4125.0	6493.0	5523.0	5258.0	5781.0	6775.0	4971.0
24.	CABANG-099	PKD0-0000025	PKD0-0000025	2000.0	283.0	2865.0	2924.0	3613.0	3481.0	3931.0	3150.0	3541.0	2751.0	2775.0	2826.0	3637.0
25.	CABANG-099	PKD0-0000026	PKD0-0000026	2000.0	266.0	251.0	511.0	276.0	248.0	228.0	222.0	371.0	284.0	233.0	216.0	381.0
26.	CABANG-099	PKD0-0000027	PKD0-0000027	2000.0	4493.0	3828.0	4394.0	3587.0	3443.0	4697.0	4157.0	4834.0	4438.0	4040.0	4178.0	5182.0
27.	CABANG-099	PKD0-0000028	PKD0-0000028	2000.0	16732.0	22785.0	16965.0	17572.0	22063.0	25995.0	19101.0	25680.0	21570.0	16528.0	20259.0	18665.0
28.	CABANG-099	PKD0-0000029	PKD0-0000029	2000.0	1891.0	2235.0	2876.0	3217.0	2676.0	1995.0	2192.0	3132.0	3120.0	2477.0	2435.0	2231.0
29.	CABANG-099	PKD0-0000030	PKD0-0000030	2000.0	1262.0	1162.0	1510.0	1869.0	1618.0	1520.0	1788.0	2006.0	1888.0	1765.0	1957.0	1132.0
30.	CABANG-099	PKD0-0000031	PKD0-0000031	2000.0	7179.0	7513.0	8207.0	8265.0	9068.0	9574.0	8196.0	4497.0	6575.0	7783.0	7577.0	6025.0
31.	CABANG-099	PKD0-0000032	PKD0-0000032	2000.0	533.0	533.0	533.0	533.0	533.0	533.0	533.0	533.0	533.0	533.0	533.0	533.0

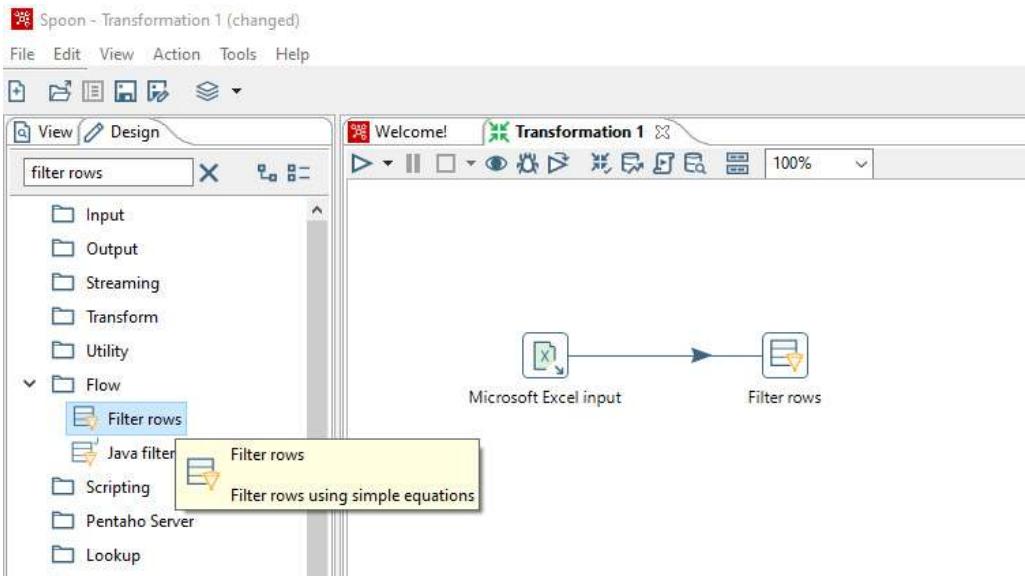
Close Show Log

9. Proses ekstraksi selesai.

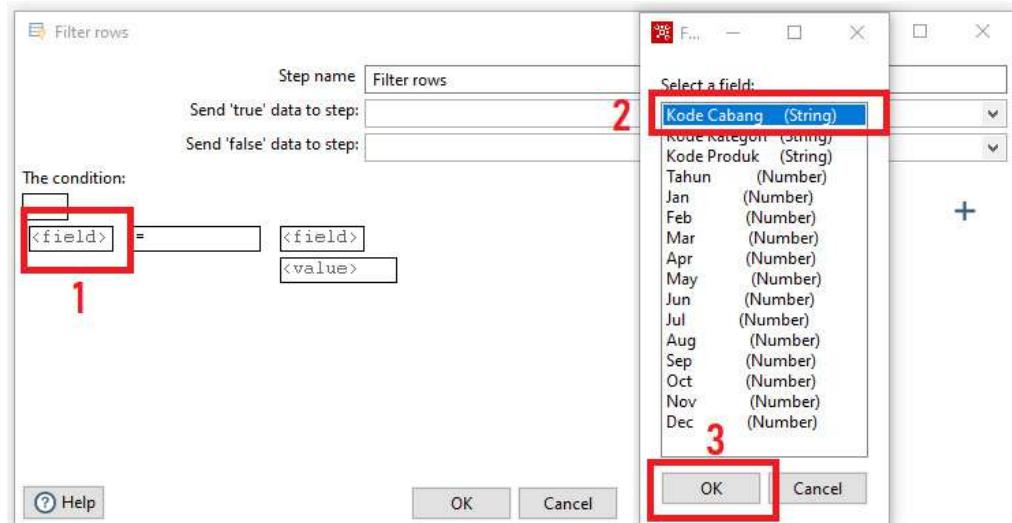
2.4.2 Transformasi Data

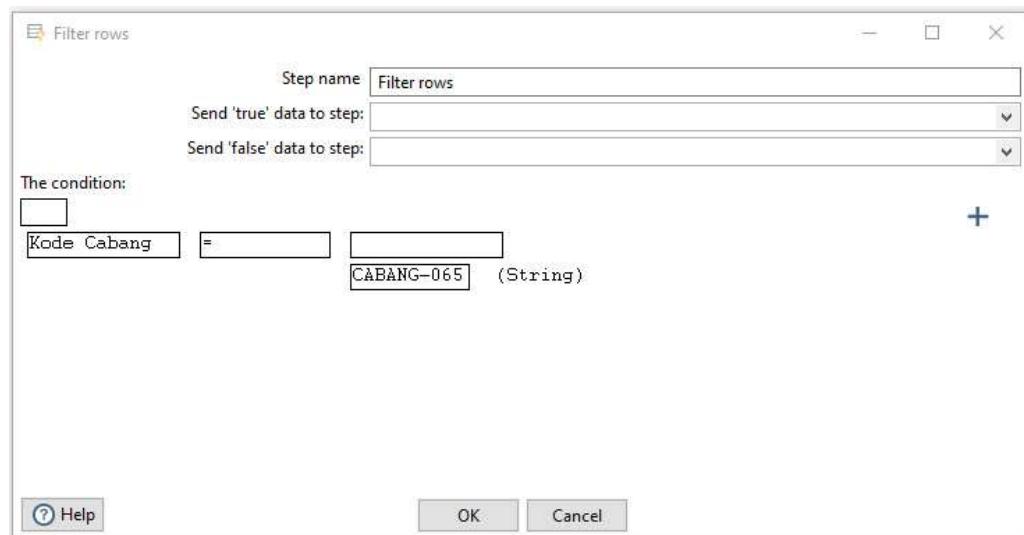
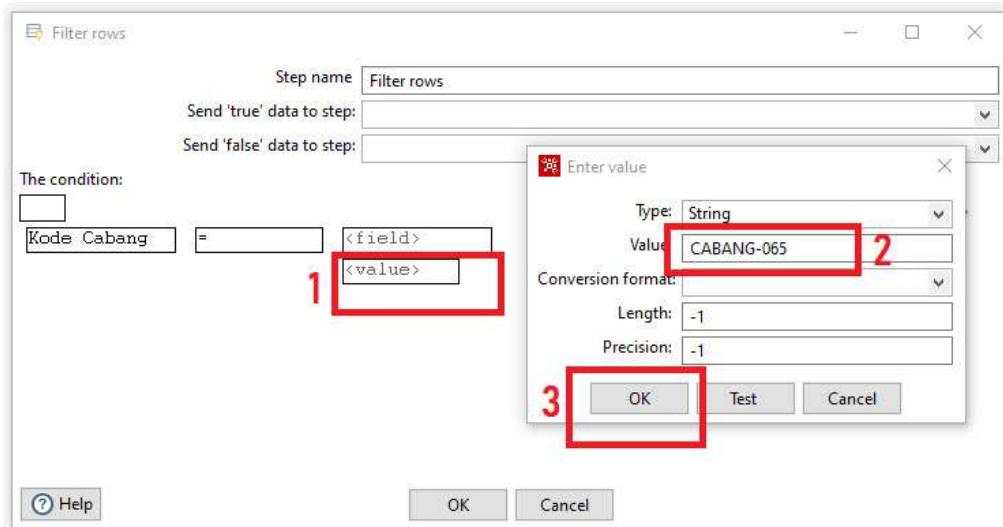
Pada kegiatan ini, akan dilakukan proses transformasi data yaitu filtering (filter data sehingga hanya menyisakan **Kode Cabang = 'CABANG-065'**), dilanjutkan melakukan penghitungan **Total_Target_Q1** dengan menambahkan data target pada bulan di Kuartal pertama (Jan, Feb, Mar). Tahapan praktikum sebagai berikut :

- Pada sidebar **Design** ketik **Filter rows** dan lakukan *drag and drop* ke canvas. Hubungkan step 1 (**Microsoft Excel input**) dengan step 2 (**Filter rows**) dengan cara tekan tombol Shift lalu klik kiri dan tarik dari step 1 hingga terhubung ke step 2.

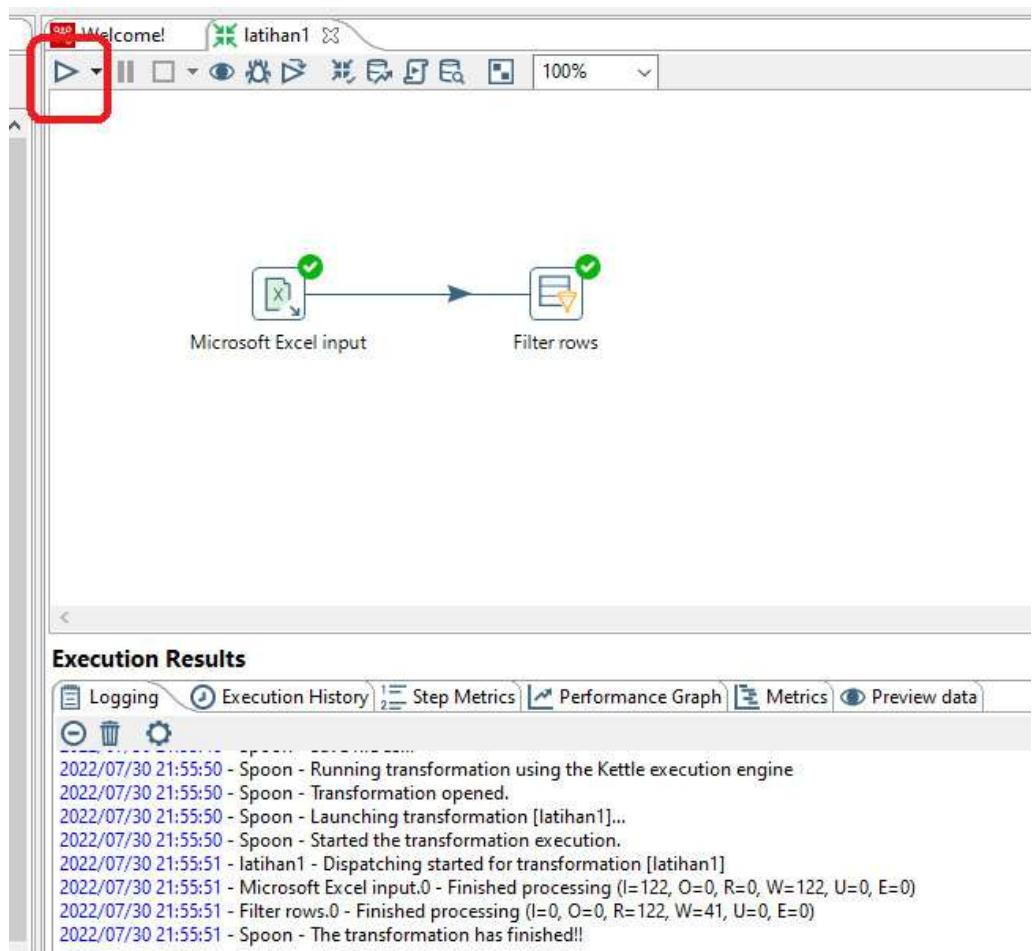


2. Konfigurasi filtering (filter data sehingga hanya menyisakan **Kode Cabang = 'CABANG-065'**). Klik kanan pada **Filter rows** pilih Edit, tambahkan kondisi seperti berikut lalu klik OK.

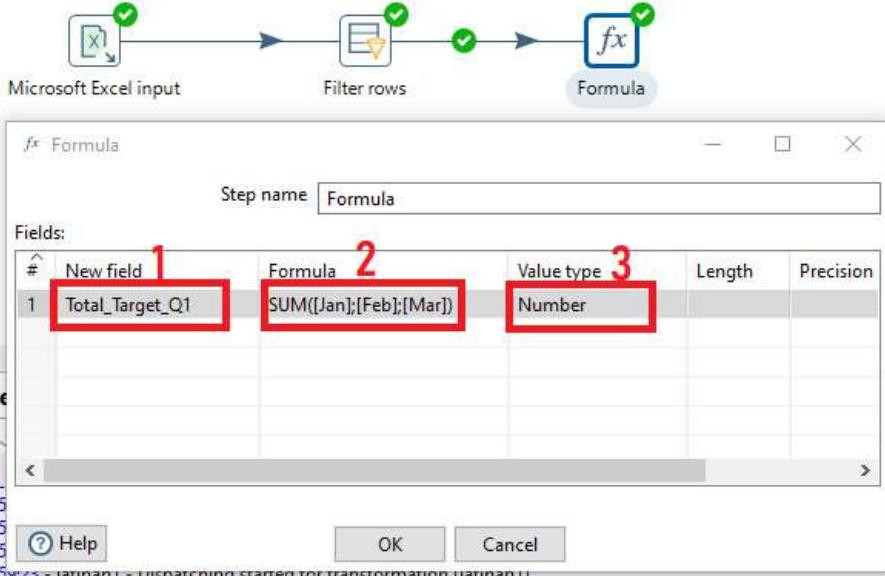




3. Jalankan dengan menekan tombol **Run** dan simpan.



4. Langkah selanjutnya yaitu melakukan penghitungan **Total_Target_Q1**. Pada sidebar **Design**, ketik **Formula** lalu *drag and drop* ke canvas, hubungkan dengan step sebelumnya (pilih **Result is TRUE**) lalu klik kanan dan Edit. Isikan seperti berikut.

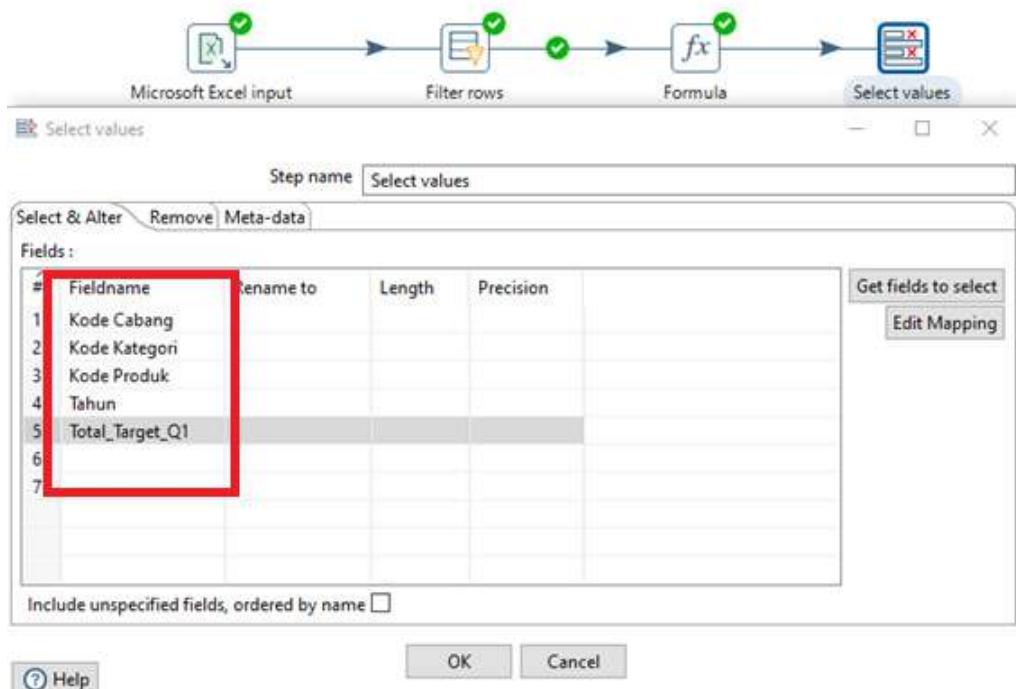


7/30 21:59:23 - latihan1 - Dispatching started for transformation [latihan1]
7/30 21:59:23 - Microsoft Excel input.0 - Finished processing (I=122, O=0, R=0, W=122, U=0, E=0)

5. Jalankan dengan menekan tombol **Run**. Untuk melihat apakah hasil yang diharapkan sudah sesuai atau belum, klik kanan pada step **Formula** dan pilih **Preview > Quick Launch**. Jika sudah sesuai maka proses transformasi data sudah selesai.

Rows of step: Formula (41 rows)																	
#	Kode Cabang	Kode Kategori	Kode Produk	Tahun	Jan	Feb	Mar	Apr	Mey	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total_Target_Q1
1	CABANG-065	PROD-0000001	PROD-0000001	2008.0	5228.0	3013.0	4492.0	2799.0	3050.0	4800.0	3758.0	5615.0	5294.0	5567.0	4043.0	4928.0	12733.0
2	CABANG-065	PROD-0000002	PROD-0000002	2008.0	1220.0	1833.0	1846.0	1707.0	1116.0	1488.0	1465.0	1675.0	1768.0	1315.0	1826.0	1168.0	4999.0
3	CABANG-065	PROD-0000003	PROD-0000003	2008.0	1922.0	2780.0	2533.0	1708.0	1737.0	2190.0	1843.0	2948.0	2872.0	2566.0	2404.0	1858.0	7255.0
4	CABANG-065	PROD-0000004	PROD-0000004	2008.0	3560.0	2729.0	2880.0	2413.0	3027.0	3700.0	3901.0	2864.0	2614.0	2797.0	2888.0	2645.0	8578.0
5	CABANG-065	PROD-0000005	PROD-0000005	2008.0	3324.0	3006.0	3181.0	3163.0	2675.0	1778.0	2168.0	2072.0	3106.0	2968.0	1833.0	2476.0	9711.0
6	CABANG-065	PROD-0000006	PROD-0000006	2008.0	3001.0	3208.0	3490.0	1960.0	3127.0	3211.0	3307.0	3253.0	3888.0	3057.0	2082.0	1428.0	1746.0
7	CABANG-065	PROD-0000007	PROD-0000007	2008.0	1402.0	1756.0	2215.0	1746.0	1594.0	1796.0	1184.0	2082.0	1428.0	1697.0	1391.0	5373.0	
8	CABANG-065	PROD-0000008	PROD-0000008	2008.0	458.0	435.0	303.0	485.0	489.0	353.0	497.0	401.0	379.0	319.0	324.0	1196.0	
9	CABANG-065	PROD-0000009	PROD-0000009	2008.0	1976.0	1660.0	2355.0	1727.0	2145.0	1668.0	1252.0	2537.0	1906.0	1407.0	1678.0	2192.0	
10	CABANG-065	PROD-0000010	PROD-0000010	2008.0	343.0	449.0	456.0	435.0	338.0	377.0	355.0	466.0	304.0	303.0	347.0	289.0	
11	CABANG-065	PROD-0000011	PROD-0000011	2008.0	480.0	413.0	551.0	425.0	521.0	394.0	365.0	510.0	355.0	440.0	272.0	390.0	
12	CABANG-065	PROD-0000012	PROD-0000012	2008.0	1165.0	1417.0	1336.0	1115.0	1266.0	1271.0	1084.0	1757.0	1089.0	1506.0	990.0	960.0	
13	CABANG-065	PROD-0000013	PROD-0000013	2008.0	4428.0	4588.0	4739.0	4949.0	4512.0	3652.0	4068.0	4035.0	4873.0	5133.0	4496.0	4605.0	
14	CABANG-065	PROD-0000014	PROD-0000014	2008.0	4504.0	2899.0	4851.0	4685.0	5434.0	4847.0	5258.0	5775.0	4889.0	2962.0	5194.0	4309.0	
15	CABANG-065	PROD-0000015	PROD-0000015	2008.0	9362.0	6855.0	6375.0	7346.0	6188.0	5472.0	8513.0	7066.0	7219.0	7144.0	6845.0	8377.0	
16	CABANG-065	PROD-0000016	PROD-0000016	2008.0	28936.0	24588.0	47387.0	3102.0	42083.0	43473.0	42398.0	32898.0	38483.0	46629.0	34963.0	36166.0	
17	CABANG-065	PROD-0000017	PROD-0000017	2008.0	353.0	417.0	414.0	323.0	388.0	443.0	482.0	567.0	367.0	432.0	538.0	418.0	
18	CABANG-065	PROD-0000018	PROD-0000018	2008.0	3417.0	3468.0	3576.0	3055.0	3066.0	4113.0	3507.0	2817.0	2262.0	3026.0	3505.0	4636.0	
19	CABANG-065	PROD-0000019	PROD-0000019	2008.0	2111.0	2501.0	2739.0	1896.0	2902.0	3069.0	2732.0	2355.0	1748.0	2015.0	1988.0	2759.0	
20	CABANG-065	PROD-0000020	PROD-0000020	2008.0	2016.0	1959.0	1536.0	2291.0	1332.0	1298.0	2094.0	2207.0	1424.0	1469.0	2086.0	2187.0	
21	CABANG-065	PROD-0000021	PROD-0000021	2008.0	3499.0	4836.0	5414.0	5902.0	5533.0	4176.0	6172.0	3394.0	3787.0	5340.0	4301.0	3383.0	
22	CABANG-065	PROD-0000022	PROD-0000022	2008.0	29006.0	17934.0	23424.0	23428.0	26193.0	27978.0	18227.0	32821.0	24588.0	29255.0	19935.0	28474.0	
23	CABANG-065	PROD-0000023	PROD-0000023	2008.0	3797.0	4531.0	4442.0	5544.0	6505.0	6801.0	7079.0	7885.0	6986.0	5337.0	4440.0	5761.0	
24	CABANG-065	PROD-0000024	PROD-0000024	2008.0	8067.0	5918.0	3732.0	5943.0	6149.0	4532.0	5107.0	4114.0	6623.0	6078.0	4326.0	4039.0	
25	CABANG-065	PROD-0000025	PROD-0000025	2008.0	4284.0	4173.0	3682.0	3521.0	2750.0	4435.0	3432.0	2783.0	3896.0	4618.0	3240.0	3082.0	
26	CABANG-065	PROD-0000026	PROD-0000026	2008.0	212.0	324.0	349.0	405.0	253.0	274.0	407.0	387.0	299.0	339.0	267.0	396.0	
27	CABANG-065	PROD-0000027	PROD-0000027	2008.0	5788.0	4281.0	5089.0	2821.0	3441.0	6078.0	6767.0	5644.0	4962.0	4109.0	4383.0	3014.0	
28	CABANG-065	PROD-0000028	PROD-0000028	2008.0	26632.0	14966.0	21073.0	14640.0	15920.0	25152.0	23668.0	22247.0	22400.0	16274.0	2112.0	27713.0	
29	CABANG-065	PROD-0000029	PROD-0000029	2008.0	2534.0	1381.0	2052.0	2444.0	2145.0	2896.0	2864.0	2389.0	1845.0	2652.0	2224.0	5967.0	

6. Selanjutnya kita dapat menyimpan data hasil transformasi data tersebut pada komputer dalam format tertentu, pada praktikum ini kita akan menyimpan dalam format excel. Pada sidebar **Design** pilih **Select values** lalu *drag and drop* ke canvas, hubungkan dengan step sebelumnya lalu klik kanan >> Edit seperti berikut.



Pada praktikum ini kolom yang akan ditampilkan hanya lima saja, yaitu : Kode Cabang, Kode Kategori, Kode Produk, Tahun dan Total_Target_Q1. Anda juga bisa menambahkan kolom bulan Jan-Des jika diperlukan. Periksa hasilnya dengan klik kanan **Preview >> Quick Launch**.

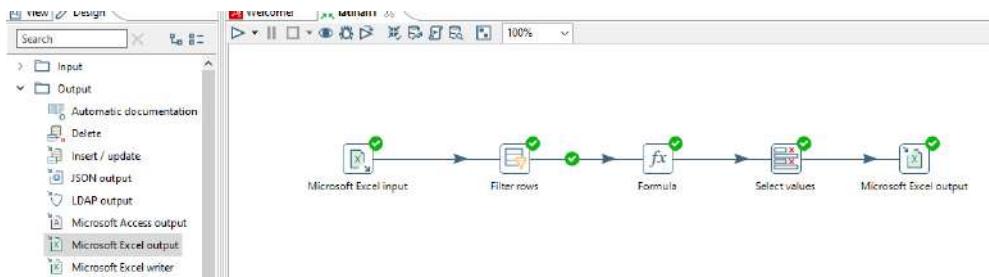
 Examine preview data

Rows of step: Select values (41 rows)

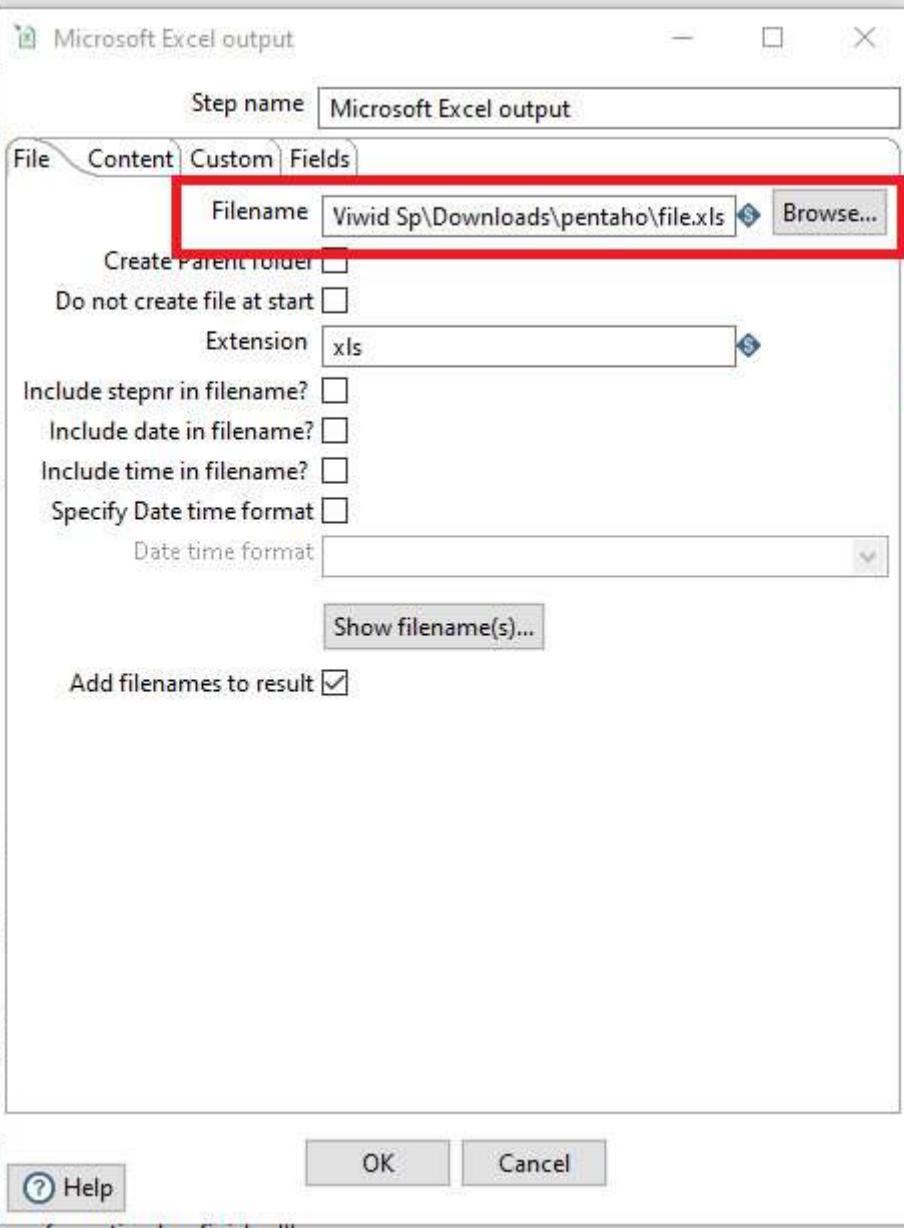
#	Kode Cabang	Kode Kategori	Kode Produk	Tahun	Total_Target_Q1
1	CABANG-065	PROD-0000001	PROD-0000001	2008.0	12733.0
2	CABANG-065	PROD-0000002	PROD-0000002	2008.0	4899.0
3	CABANG-065	PROD-0000003	PROD-0000003	2008.0	7255.0
4	CABANG-065	PROD-0000004	PROD-0000004	2008.0	8578.0
5	CABANG-065	PROD-0000005	PROD-0000005	2008.0	9711.0
6	CABANG-065	PROD-0000006	PROD-0000006	2008.0	9699.0
7	CABANG-065	PROD-0000007	PROD-0000007	2008.0	5373.0
8	CABANG-065	PROD-0000008	PROD-0000008	2008.0	1196.0
9	CABANG-065	PROD-0000009	PROD-0000009	2008.0	5934.0
10	CABANG-065	PROD-0000010	PROD-0000010	2008.0	1248.0
11	CABANG-065	PROD-0000011	PROD-0000011	2008.0	1444.0
12	CABANG-065	PROD-0000012	PROD-0000012	2008.0	3918.0
13	CABANG-065	PROD-0000013	PROD-0000013	2008.0	13755.0
14	CABANG-065	PROD-0000014	PROD-0000014	2008.0	12254.0
15	CABANG-065	PROD-0000015	PROD-0000015	2008.0	22592.0
16	CABANG-065	PROD-0000016	PROD-0000016	2008.0	100911.0
17	CABANG-065	PROD-0000017	PROD-0000017	2008.0	1184.0
18	CABANG-065	PROD-0000018	PROD-0000018	2008.0	10461.0
19	CABANG-065	PROD-0000019	PROD-0000019	2008.0	7351.0
20	CABANG-065	PROD-0000020	PROD-0000020	2008.0	5511.0
21	CABANG-065	PROD-0000021	PROD-0000021	2008.0	13749.0
22	CABANG-065	PROD-0000022	PROD-0000022	2008.0	70364.0
23	CABANG-065	PROD-0000023	PROD-0000023	2008.0	12590.0
24	CABANG-065	PROD-0000024	PROD-0000024	2008.0	17717.0
25	CABANG-065	PROD-0000025	PROD-0000025	2008.0	12139.0
26	CABANG-065	PROD-0000026	PROD-0000026	2008.0	885.0
27	CABANG-065	PROD-0000027	PROD-0000027	2008.0	15158.0
28	CABANG-065	PROD-0000028	PROD-0000028	2008.0	62671.0
29	CABANG-065	PROD-0000029	PROD-0000029	2008.0	5967.0
30	CABANG-065	PROD-0000030	PROD-0000030	2008.0	4198.0
31	CABANG-065	PROD-0000031	PROD-0000031	2008.0	15200.0

Close

7. Selanjutnya pada sidebar **Design**, ketik **Microsoft Excel output** lalu *drag and drop* ke canvas, hubungkan dengan step terakhir (**Select values**) seperti berikut ini.



8. Atur lokasi penyimpanan dan penamaan file dengan pilih klik kanan
-> Edit.



9. Jalankan dengan menekan tombol **Run**. Untuk melihat apakah isi file sudah sesuai atau belum dapat dilihat pada direktori lokasi file yang dipilih. Berikut adalah hasilnya.

	A	B	C	D	E
1	Kode Cabang	Kode Kategori	Kode Produk	Tahun	Total_Target_Q1
2	CABANG-065	PROD-0000001	PROD-0000001	2.008,00	12.733,00
3	CABANG-065	PROD-0000002	PROD-0000002	2.008,00	4.899,00
4	CABANG-065	PROD-0000003	PROD-0000003	2.008,00	7.255,00
5	CABANG-065	PROD-0000004	PROD-0000004	2.008,00	8.578,00
6	CABANG-065	PROD-0000005	PROD-0000005	2.008,00	9.711,00
7	CABANG-065	PROD-0000006	PROD-0000006	2.008,00	9.699,00
8	CABANG-065	PROD-0000007	PROD-0000007	2.008,00	5.373,00
9	CABANG-065	PROD-0000008	PROD-0000008	2.008,00	1.196,00
10	CABANG-065	PROD-0000009	PROD-0000009	2.008,00	5.934,00
11	CABANG-065	PROD-0000010	PROD-0000010	2.008,00	1.248,00
12	CABANG-065	PROD-0000011	PROD-0000011	2.008,00	1.444,00
13	CABANG-065	PROD-0000012	PROD-0000012	2.008,00	3.918,00
14	CABANG-065	PROD-0000013	PROD-0000013	2.008,00	13.755,00
15	CABANG-065	PROD-0000014	PROD-0000014	2.008,00	12.254,00
16	CABANG-065	PROD-0000015	PROD-0000015	2.008,00	22.592,00
17	CABANG-065	PROD-0000016	PROD-0000016	2.008,00	100.911,00
18	CABANG-065	PROD-0000017	PROD-0000017	2.008,00	1.184,00
19	CABANG-065	PROD-0000018	PROD-0000018	2.008,00	10.461,00
20	CABANG-065	PROD-0000019	PROD-0000019	2.008,00	7.351,00
21	CABANG-065	PROD-0000020	PROD-0000020	2.008,00	5.511,00
22	CABANG-065	PROD-0000021	PROD-0000021	2.008,00	13.749,00
23	CABANG-065	PROD-0000022	PROD-0000022	2.008,00	70.364,00
24	CABANG-065	PROD-0000023	PROD-0000023	2.008,00	12.590,00
25	CABANG-065	PROD-0000024	PROD-0000024	2.008,00	17.717,00
26	CABANG-065	PROD-0000025	PROD-0000025	2.008,00	12.139,00
27	CABANG-065	PROD-0000026	PROD-0000026	2.008,00	885,00

2.5 Tugas

Dengan menggunakan file excel “**Target Penjualan New.xlsx**”, tambahkan penghitungan Total Target pada Quartal 2 (Q2), Total Target pada Quartal 3 (Q3) serta Total Target pada Quartal 4 (Q4)!

Selesaikan tugas tersebut di kelas. Jika belum selesai, bisa dilanjutkan di rumah dan dikumpulkan sebelum pertemuan berikutnya sesuai arahan dosen pengampu.

Modul 3

Tabel Dimensi

3.1 Tujuan

1. Mahasiswa mampu melakukan proses ETL secara lebih lanjut pada pengembangan sebuah data warehouse.
2. Mahasiswa mampu melakukan proses perbaikan data sebelum data dimasukkan ke data warehouse.

3.2 Landasan Teori

Data extraction adalah proses pengambilan data yang diperlukan dari sumber data dan selanjutnya dimasukkan pada *staging area* untuk diproses pada tahap berikutnya. Terdapat berbagai tipe sumber data, berbagai format data, mesin yang berbeda, *software* dan arsitektur yang tidak sama. Sebelum proses ini dilakukan, sebaiknya perlu didefinisikan *requirement* terhadap sumber data yang akan digunakan untuk lebih memudahkan pada *extraction data*.

Data cleansing bertujuan untuk menghilangkan kesalahan-kesalahan pada data yang diakibatkan oleh proses transaksional. Latar belakang yang penting perlunya *data cleansing* adalah bahwa jika *data cleansing* ini salah maka hal terburuk yang terjadi adalah pemberian informasi yang salah kepada pengambil kebijakan. Jika informasi yang salah ini dipercaya maka keputusan yang diambil akan jatuh dan bisa mengakibatkan kerugian yang besar.

Di dalam model multidimensional, *database* terdiri dari beberapa tabel fakta (*fact tables*) dan tabel dimensi (*dimension tables*) yang saling

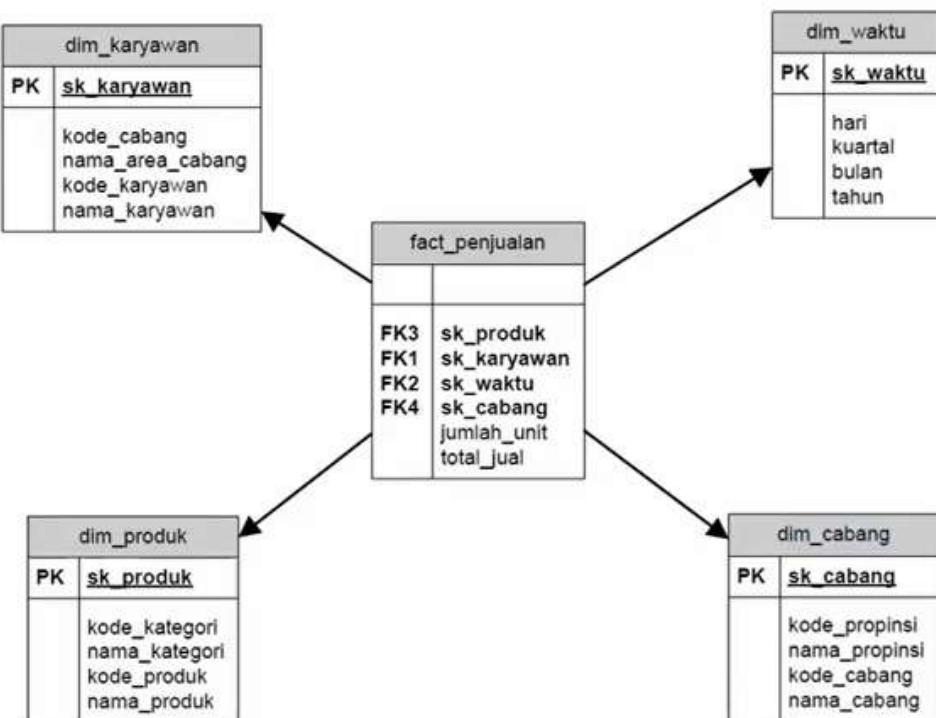
terkait. Suatu tabel fakta berisi berbagai nilai agregasi yang menjadi dasar pengukuran (*measure*) serta beberapa *key* yang terkait ke tabel dimensi yang akan menjadi sudut pandang dari *measure* tersebut.

3.3 Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi XAMPP.
3. Program aplikasi Pentaho Data Integration.
4. Modul Praktikum Data Warehousing dan Data Mining.

3.4 Langkah-langkah Praktikum

Pada praktikum ini, kita akan menggunakan database **phi_minimart** yang dapat diunduh melalui https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM/src/branch/master/Data/ETL/phi-minimart.docx. Langkah pertama jalankan aplikasi XAMPP, selanjutnya siapkan dua database: **phi_minimart** (sebagai data asal) dan **dw_phi** (digunakan untuk menyimpan tabel-tabel dimensi). Tabel dimensi yang akan kita buat ada empat : dimensi karyawan, dimensi waktu, dimensi produk, dimensi cabang. Star schema dari database PHI-minimart seperti gambar di bawah ini.



Gambar 3.1 Star Schema Database PHI-minimart

3.4.1 Menghubungkan MySQL dengan Pentaho

1. Unduh MySQL Connector di <https://downloads.mysql.com/archives/c-j/>.

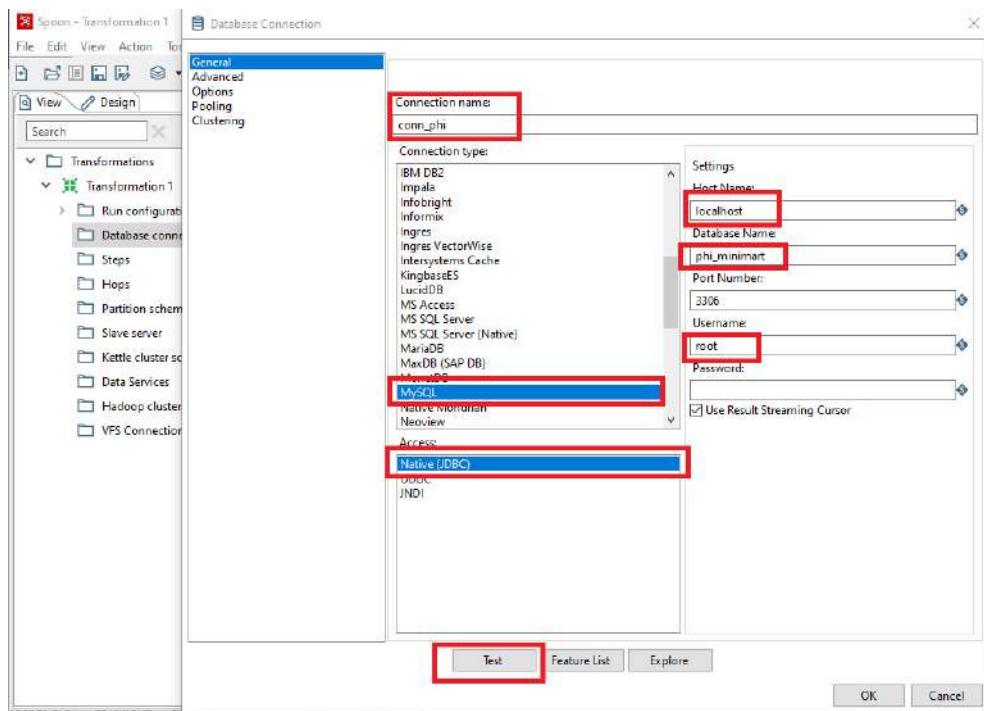
The screenshot shows the MySQL Product Archives page. At the top, there is a note: "Please note that these are old versions. New releases will have recent bug fixes and features! To download the latest release of MySQL Connector/J, please visit MySQL Downloads." Below this, there are dropdown menus for "Product Version" (set to 8.0.29) and "Operating System" (set to Platform Independent). Two download links are listed: one for a Compressed TAR Archive (mysql-connector-java-8.0.29.tar.gz) and one for a ZIP Archive (mysql-connector-java-8.0.29.zip). The ZIP archive link is highlighted with a red box. A note at the bottom suggests using MD5 checksums and GnuPG signatures for verification.

2. Ekstrak dan salin file *.jar pada direktori **lib** folder Pentaho / data-integration.

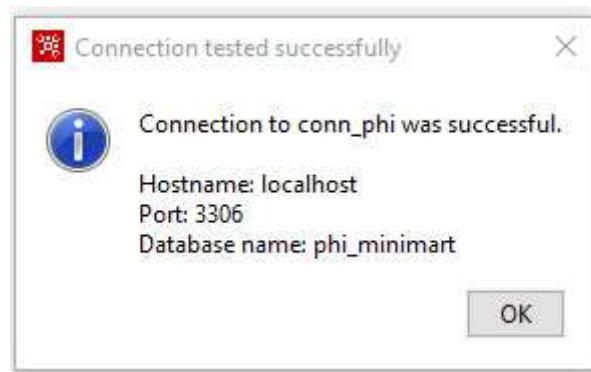
The screenshot shows a Windows File Explorer window with the path "This PC > Windows (C:) > data-integration > lib". The table lists several JAR files:

	Name	Date modified	Type	Size
20	mf-200507110943.jar	02/06/2021 18:39	Executable Jar File	171 KB
	mondrian-9.2.0.0-290.jar	02/06/2021 18:35	Executable Jar File	3,499 KB
	monetdb-jdbc-2.8.jar	02/06/2021 18:36	Executable Jar File	115 KB
	mstor-0.9.13.jar	02/06/2021 18:36	Executable Jar File	192 KB
	<input checked="" type="checkbox"/> mysql-connector-java-8.0.29.jar	08/03/2022 19:20	Executable Jar File	2,461 KB
	nbmdr-200507110943-custom.jar	02/06/2021 18:39	Executable Jar File	605 KB
	nekohtml-1.9.15.jar	02/06/2021 18:36	Executable Jar File	146 KB
	odfdom-java-0.8.6.jar	02/06/2021 18:36	Executable Jar File	3,938 KB
	ognl-2.6.9.jar	02/06/2021 18:36	Executable Jar File	165 KB

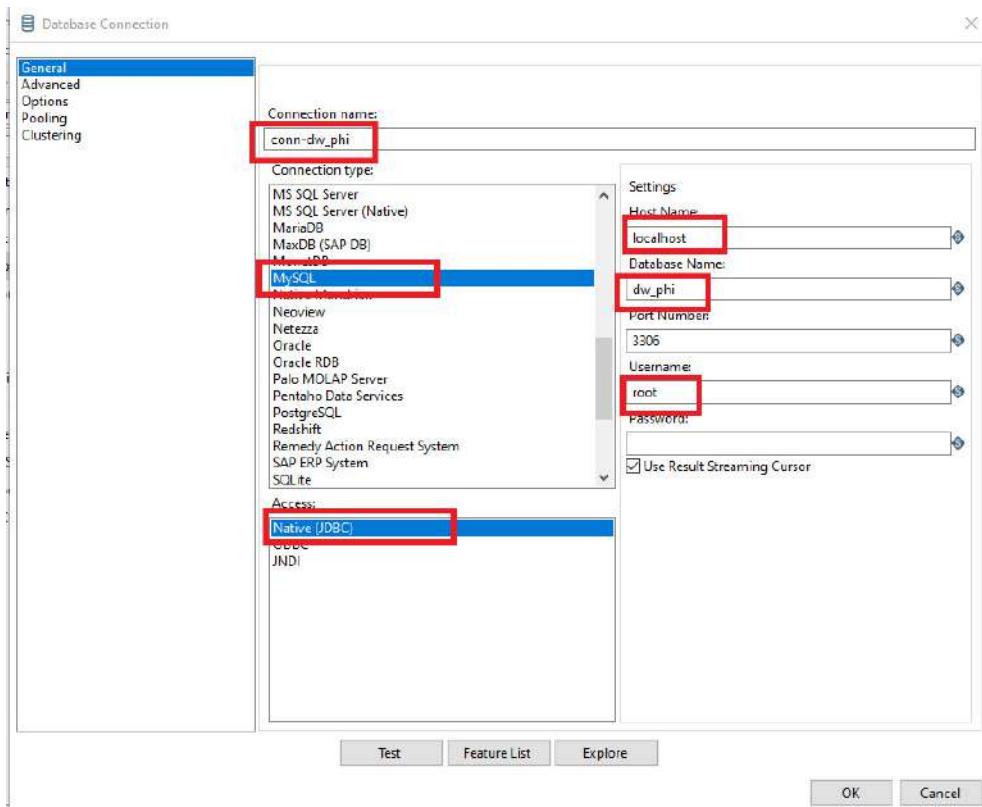
3. Jalankan aplikasi Pentaho, buat transformasi baru. Pada **View**, klik kanan **Database connections >> New**. Kita akan membuat dua connection, yang pertama untuk database **phi_minimart**. Isikan seperti gambar berikut.



4. Jika tidak ada kesalahan, akan tampil pesan seperti ini.

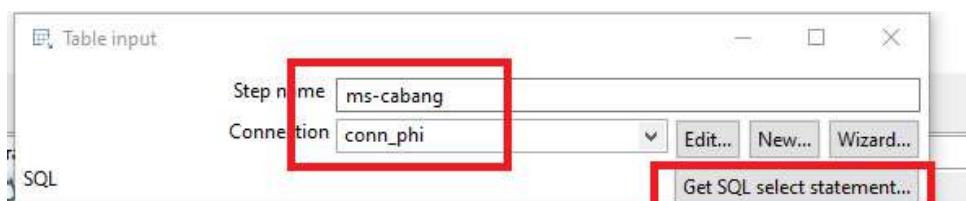
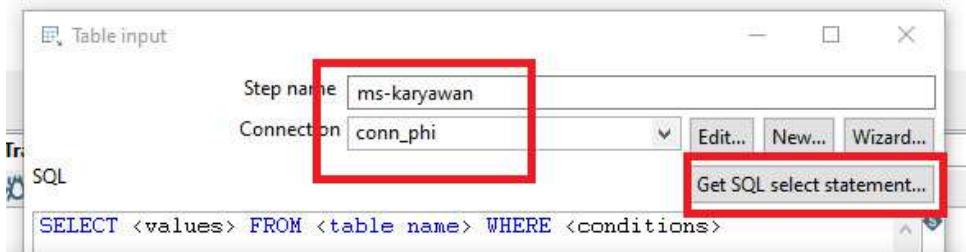


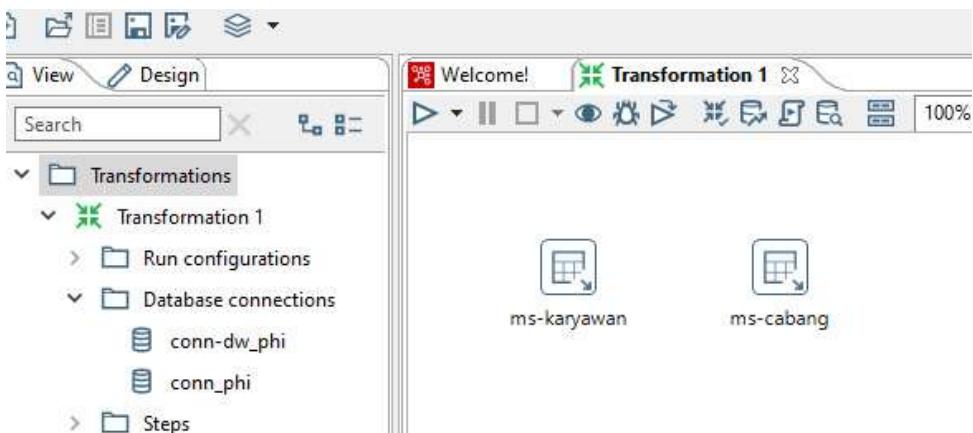
5. Connection yang kedua digunakan untuk database **dw_phi**.



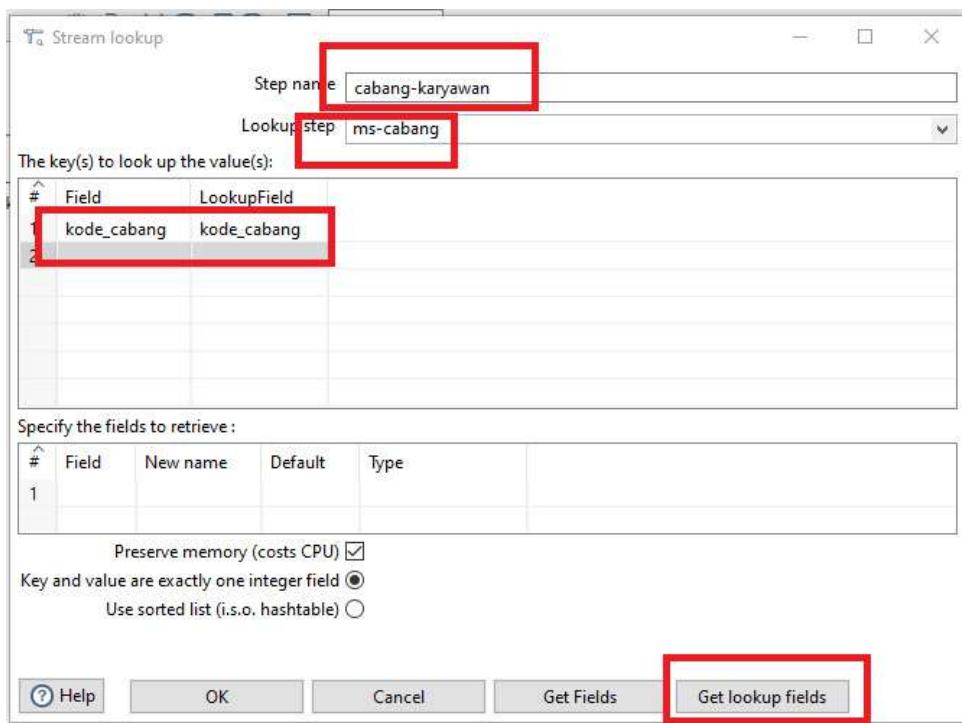
3.4.2 Tabel Dimensi Karyawan

1. Pada Design pilih step **Table input** sebanyak 2X, kemudian Edit menjadi **ms-karyawan** dan **ms-cabang**.





2. Gabungkan tabel karyawan dan tabel cabang menggunakan step **Stream lookup**. Edit seperti berikut.



3. Pada **Specify the fields to retrieve**, **nama_cabang** dapat diubah menjadi **nama_area_cabang**, sedangkan **kode_kota** dihapuskan saja.

Specify the fields to retrieve :				
#	Field	New name	Default	Type
1	kode_cabang			String
2	nama_cabang	nama_area_cabang		String
3				

4. Periksa hasilnya dengan klik kanan **Preview >> Quick Launch**.

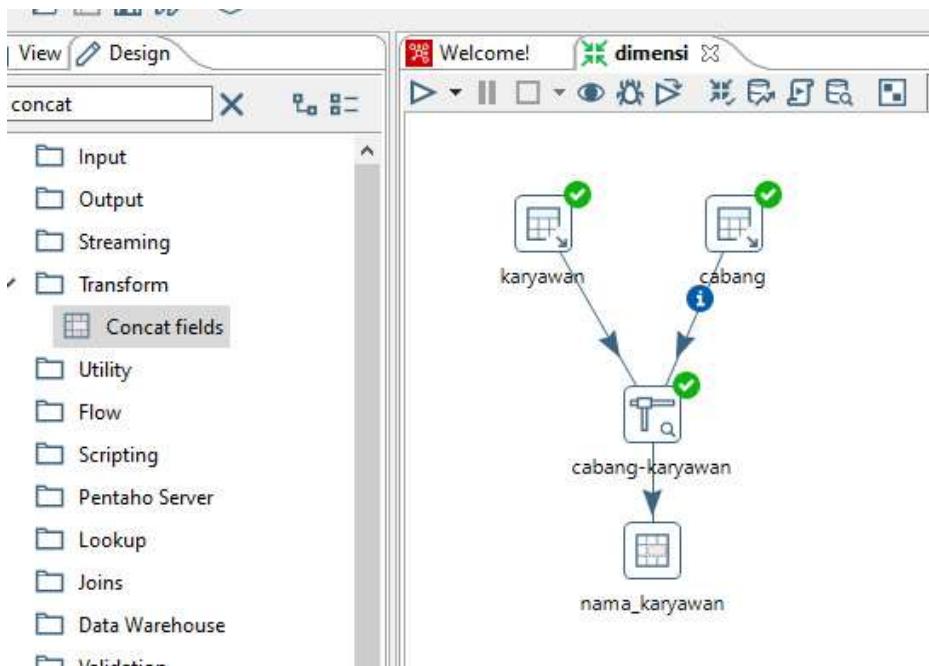
Examine preview data

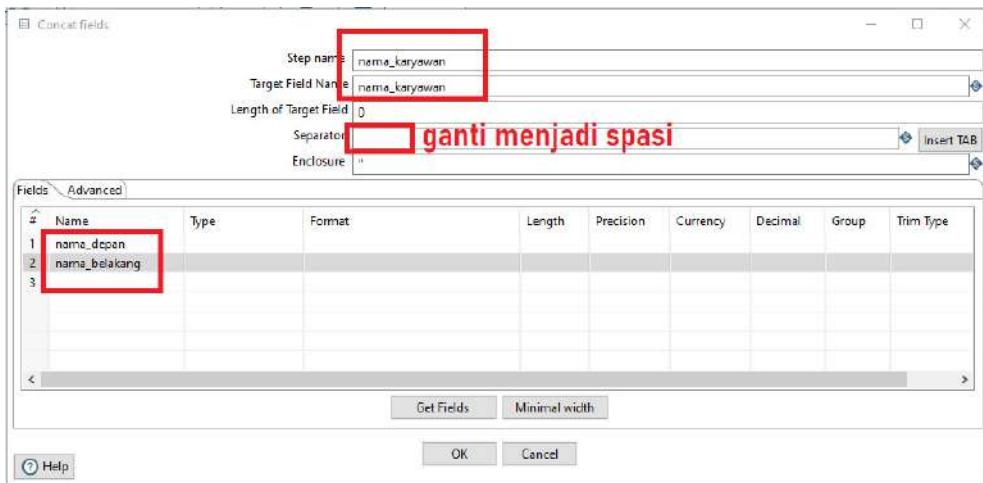
Rows of step: cabang-karyawan (30 rows)

#	kode_cabang	kode_karyawan	nama_depan	nama_belakang	jenis_kelamin	kode_cabang_1	nama_area_cabang
1	CABANG-039	039-147	Bintang	Maven	W	CABANG-039	PHI Mini Market - Makassar 01
2	CABANG-047	047-181	Eria	Setiawan	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01
3	CABANG-065	065-282	Galang	Setiawan	P	CABANG-065	PHI Mini Market - Surabaya 01
4	CABANG-039	039-031	Kristina	Damai	W	CABANG-039	PHI Mini Market - Makassar 01
5	CABANG-047	047-075	Eko	Rukun	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01
6	CABANG-065	065-076	Natali	Menawan	W	CABANG-065	PHI Mini Market - Surabaya 01
7	CABANG-039	039-214	Mawar	Mardi	W	CABANG-039	PHI Mini Market - Makassar 01
8	CABANG-047	047-055	Erman	Margo	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01
9	CABANG-065	065-061	Ayu	Pekerti	W	CABANG-065	PHI Mini Market - Surabaya 01
10	CABANG-039	039-044	Ferdy	Tenteram	P	CABANG-039	PHI Mini Market - Makassar 01
11	CABANG-047	047-133	Harum	Maven	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01
12	CABANG-065	065-023	Harum	Selangit	W	CABANG-065	PHI Mini Market - Surabaya 01
13	CABANG-039	039-212	Agus	Dewangga	P	CABANG-039	PHI Mini Market - Makassar 01
14	CABANG-047	047-031	Kristina	Damai	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01
15	CABANG-065	065-060	Mulia	Setiawan	P	CABANG-065	PHI Mini Market - Surabaya 01
16	CABANG-039	039-053	Galang	Terang	P	CABANG-039	PHI Mini Market - Makassar 01
17	CABANG-047	047-244	Budiwati	Ramah	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01
18	CABANG-065	065-007	Budi	Tenteram	P	CABANG-065	PHI Mini Market - Surabaya 01
19	CABANG-039	039-127	Lastri	Mardi	W	CABANG-039	PHI Mini Market - Makassar 01
20	CABANG-047	047-286	Kusuma	Dominik	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01
21	CABANG-065	065-258	Mulyo	Damai	P	CABANG-065	PHI Mini Market - Surabaya 01
22	CABANG-039	039-203	Eriq	Menawan	P	CABANG-039	PHI Mini Market - Makassar 01
23	CABANG-047	047-006	Agung	Alexander	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01
24	CABANG-065	065-094	Mariani	Damai	W	CABANG-065	PHI Mini Market - Surabaya 01
25	CABANG-039	039-156	Niken	Setiawan	W	CABANG-039	PHI Mini Market - Makassar 01
26	CABANG-047	047-105	Sentosa	Indrawan	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01
27	CABANG-065	065-206	Aris	Siberut	P	CABANG-065	PHI Mini Market - Surabaya 01
28	CABANG-039	039-084	Eriq	Jagat	P	CABANG-039	PHI Mini Market - Makassar 01
29	CABANG-047	047-128	Cahya	Terang	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01
30	CABANG-065	065-078	Aron	Zaminski	W	CABANG-065	PHI Mini Market - Surabaya 01

Close

5. Terlihat nama karyawan terbagi menjadi dua kolom yaitu **nama_depan** dan **nama_belakang**, kita akan jadikan satu kolom. Untuk menggabungkan dua buah field, kita gunakan step **Concat fields**.



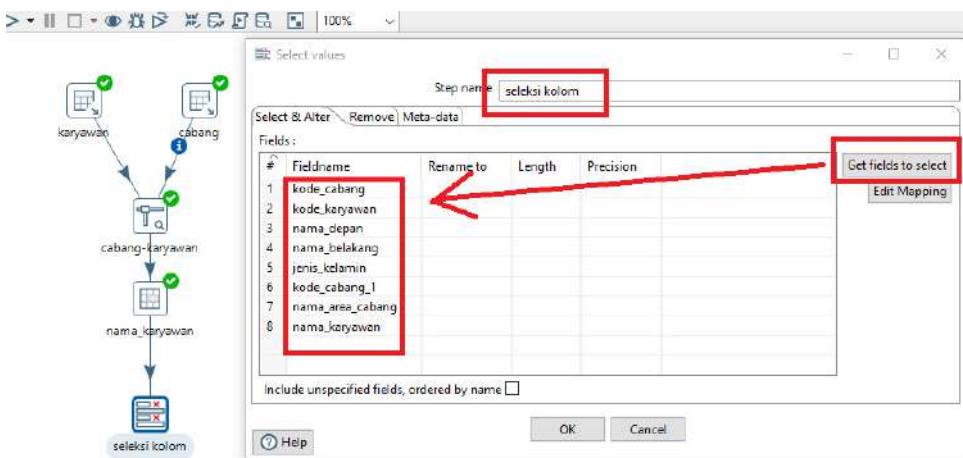


Examine preview data

Rows of step: nama_karyawan (30 rows)

#	kode_cabang	kode_karyawan	nama_depan	nama_belakang	jenis_kelamin	kode_cabang_1	nama_area_cabang	nama_karyawan
1	CABANG-039	039-147	Bintang	Maven	W	CABANG-039	PHI Mini Market - Makassar 01	Bintang Maven
2	CABANG-047	047-181	Eria	Setiawan	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Eria Setiawan
3	CABANG-065	065-282	Galang	Setiawan	P	CABANG-065	PHI Mini Market - Surabaya 01	Galang Setiawan
4	CABANG-039	039-031	Kristina	Damai	W	CABANG-039	PHI Mini Market - Makassar 01	Kristina Damai
5	CABANG-047	047-075	Eko	Rukun	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Eko Rukun
6	CABANG-065	065-076	Natali	Menawan	W	CABANG-065	PHI Mini Market - Surabaya 01	Natali Menawan
7	CABANG-039	039-214	Mawar	Mardi	W	CABANG-039	PHI Mini Market - Makassar 01	Mawar Mardi
8	CABANG-047	047-055	Erman	Margo	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Erman Margo
9	CABANG-065	065-061	Ayu	Pekerti	W	CABANG-065	PHI Mini Market - Surabaya 01	Ayu Pekerti
10	CABANG-039	039-044	Ferdy	Tenteram	P	CABANG-039	PHI Mini Market - Makassar 01	Ferdy Tenteram
11	CABANG-047	047-133	Harum	Maven	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Harum Maven
12	CABANG-065	065-023	Harum	Selangit	W	CABANG-065	PHI Mini Market - Surabaya 01	Harum Selangit
13	CABANG-039	039-212	Agus	Dewangga	P	CABANG-039	PHI Mini Market - Makassar 01	Agus Dewangga
14	CABANG-047	047-031	Kristina	Damai	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Kristina Damai
15	CABANG-065	065-060	Mulia	Setiawan	P	CABANG-065	PHI Mini Market - Surabaya 01	Mulia Setiawan
16	CABANG-039	039-053	Galang	Terang	P	CABANG-039	PHI Mini Market - Makassar 01	Galang Terang
17	CABANG-047	047-244	Budiwati	Ramah	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Budiwati Ramah
18	CABANG-065	065-007	Budi	Tenteram	P	CABANG-065	PHI Mini Market - Surabaya 01	Budi Tenteram
19	CABANG-039	039-127	Lastri	Mardi	W	CABANG-039	PHI Mini Market - Makassar 01	Lastri Mardi
20	CABANG-047	047-286	Kusuma	Dominik	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Kusuma Dominik
21	CABANG-065	065-238	Mulyo	Damai	P	CABANG-065	PHI Mini Market - Surabaya 01	Mulyo Damai
22	CABANG-039	039-203	Eriq	Menawan	P	CABANG-039	PHI Mini Market - Makassar 01	Eriq Menawan
23	CABANG-047	047-006	Agung	Alexander	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Agung Alexander
24	CABANG-065	065-094	Mariani	Damai	W	CABANG-065	PHI Mini Market - Surabaya 01	Mariani Damai
25	CABANG-039	039-156	Niken	Setiawan	W	CABANG-039	PHI Mini Market - Makassar 01	Niken Setiawan
26	CABANG-047	047-105	Sentosa	Indrawan	P	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Sentosa Indrawan
27	CABANG-065	065-206	Aris	Siberut	P	CABANG-065	PHI Mini Market - Surabaya 01	Aris Siberut
28	CABANG-039	039-084	Eriq	Jagat	P	CABANG-039	PHI Mini Market - Makassar 01	Eriq Jagat
29	CABANG-047	047-128	Cahya	Terang	W	CABANG-047	PHI Mini Market - Jakarta Pusat 01	Cahya Terang
30	CABANG-065	065-078	Aron	Zaminski	W	CABANG-065	PHI Mini Market - Surabaya 01	Aron Zaminski

6. Kita hanya membutuhkan **kode_cabang**, **nama_area_cabang**, **kode_karyawan**, **nama_karyawan**. Untuk menyortirnya kita gunakan step **Select values**.



- Kita atur sesuai urutan yang kita inginkan dan hapus yang tidak kita butuhkan.

The screenshot shows the same 'Select values' dialog box. The 'Fieldname' column now lists the following fields:

#	Fieldname
1	kode_cabang
2	nama_area_cabang
3	kode_karyawan
4	nama_karyawan

The 'Include unspecified fields, ordered by name' checkbox is unchecked. At the bottom, there are 'OK' and 'Cancel' buttons, and a 'Help' link.

8. Hasilnya dapat dilihat dengan klik kanan **Preview >> Quick Launch**.

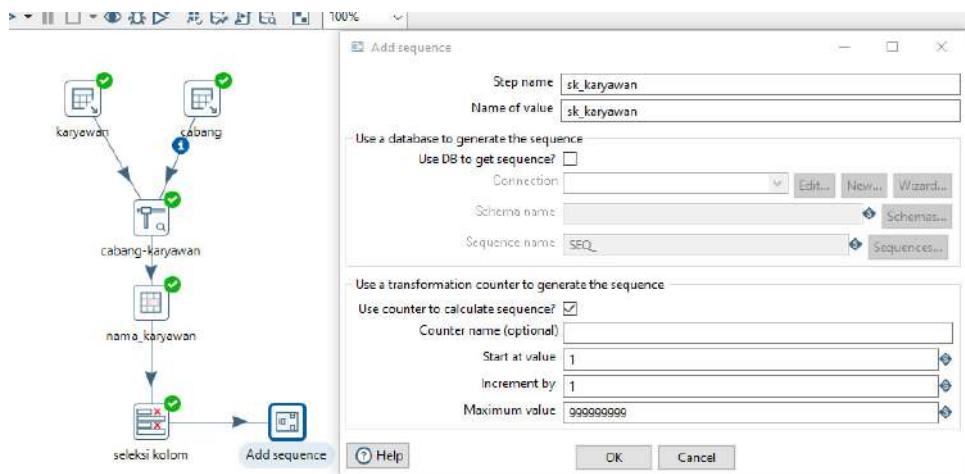
 Examine preview data

Rows of step: seleksi kolom (30 rows)

#	kode_cabang	nama_area_cabang	kode_karyawan	nama_karyawan
1	CABANG-039	PHI Mini Market - Makassar 01	039-147	Bintang Maven
2	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-181	Eria Setiawan
3	CABANG-065	PHI Mini Market - Surabaya 01	065-282	Galang Setiawan
4	CABANG-039	PHI Mini Market - Makassar 01	039-031	Kristina Damai
5	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-075	Eko Rukun
6	CABANG-065	PHI Mini Market - Surabaya 01	065-076	Natali Menawan
7	CABANG-039	PHI Mini Market - Makassar 01	039-214	Mawar Mardi
8	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-055	Erman Margo
9	CABANG-065	PHI Mini Market - Surabaya 01	065-061	Ayu Pekerti
10	CABANG-039	PHI Mini Market - Makassar 01	039-044	Ferdy Tenteram
11	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-133	Harum Maven
12	CABANG-065	PHI Mini Market - Surabaya 01	065-023	Harum Selangit
13	CABANG-039	PHI Mini Market - Makassar 01	039-212	Agus Dewangga
14	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-031	Kristina Damai
15	CABANG-065	PHI Mini Market - Surabaya 01	065-060	Mulia Setiawan
16	CABANG-039	PHI Mini Market - Makassar 01	039-053	Galang Terang
17	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-244	Budiwati Ramah
18	CABANG-065	PHI Mini Market - Surabaya 01	065-007	Budi Tenteram
19	CABANG-039	PHI Mini Market - Makassar 01	039-127	Lastri Mardi
20	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-286	Kusuma Dominik
21	CABANG-065	PHI Mini Market - Surabaya 01	065-258	Mulyo Damai
22	CABANG-039	PHI Mini Market - Makassar 01	039-203	Eriq Menawan
23	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-006	Agung Alexander
24	CABANG-065	PHI Mini Market - Surabaya 01	065-094	Mariani Damai
25	CABANG-039	PHI Mini Market - Makassar 01	039-156	Niken Setiawan
26	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-105	Sentosa Indrawan
27	CABANG-065	PHI Mini Market - Surabaya 01	065-206	Aris Siberut
28	CABANG-039	PHI Mini Market - Makassar 01	039-084	Eriq Jagat
29	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-128	Cahya Terang
30	CABANG-065	PHI Mini Market - Surabaya 01	065-078	Aron Zaminski

Close

9. Selanjutnya kita akan membuat kolom **sk_karyawan** (surrogate key) yang merupakan kunci untuk tabel dimensi karyawan. Tambahkan step **Add sequence**.



- Hasilnya dapat dilihat dengan klik kanan **Preview >> Quick Launch**.

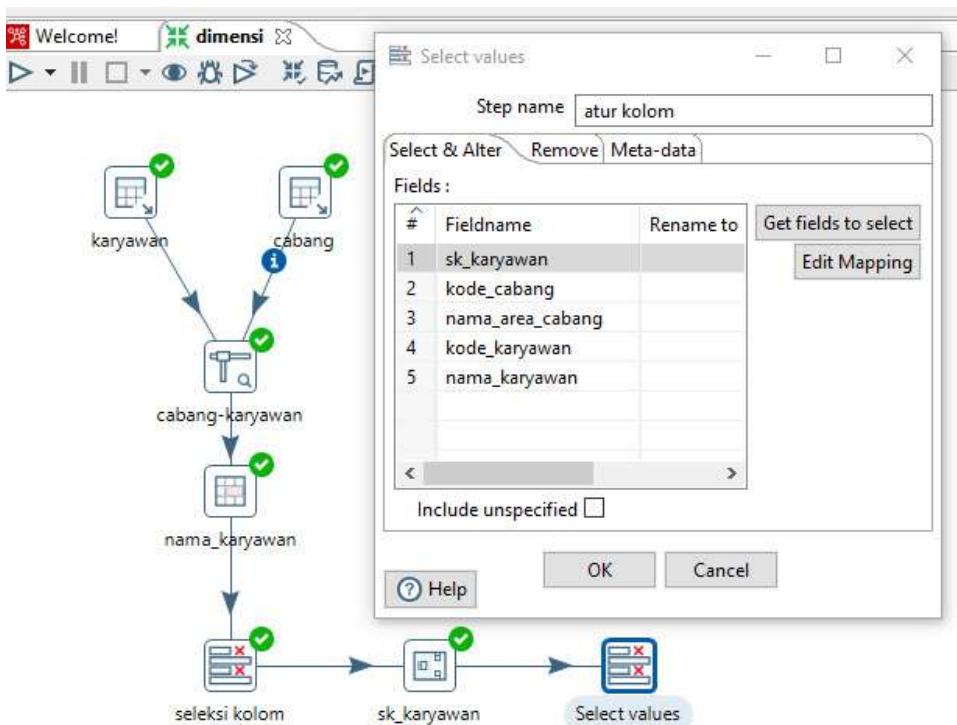
 Examine preview data

Rows of step: sk_karyawan (30 rows)

#	kode_cabang	nama_area_cabang	kode_karyawan	nama_karyawan	sk_karyawan
1	CABANG-039	PHI Mini Market - Makassar 01	039-147	Bintang Maven	1
2	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-181	Eria Setiawan	2
3	CABANG-065	PHI Mini Market - Surabaya 01	065-282	Galang Setiawan	3
4	CABANG-039	PHI Mini Market - Makassar 01	039-031	Kristina Damai	4
5	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-075	Eko Rukun	5
6	CABANG-065	PHI Mini Market - Surabaya 01	065-076	Natali Menawan	6
7	CABANG-039	PHI Mini Market - Makassar 01	039-214	Mawar Mardi	7
8	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-055	Erman Margo	8
9	CABANG-065	PHI Mini Market - Surabaya 01	065-061	Ayu Pekerti	9
10	CABANG-039	PHI Mini Market - Makassar 01	039-044	Ferdy Tenteram	10
11	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-133	Harum Maven	11
12	CABANG-065	PHI Mini Market - Surabaya 01	065-023	Harum Selangit	12
13	CABANG-039	PHI Mini Market - Makassar 01	039-212	Agus Dewangga	13
14	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-031	Kristina Damai	14
15	CABANG-065	PHI Mini Market - Surabaya 01	065-060	Mulia Setiawan	15
16	CABANG-039	PHI Mini Market - Makassar 01	039-053	Galang Terang	16
17	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-244	Budiwati Ramah	17
18	CABANG-065	PHI Mini Market - Surabaya 01	065-007	Budi Tenteram	18
19	CABANG-039	PHI Mini Market - Makassar 01	039-127	Lastri Mardi	19
20	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-286	Kusuma Dominik	20
21	CABANG-065	PHI Mini Market - Surabaya 01	065-258	Mulyo Damai	21
22	CABANG-039	PHI Mini Market - Makassar 01	039-203	Eriq Menawan	22
23	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-006	Agung Alexander	23
24	CABANG-065	PHI Mini Market - Surabaya 01	065-094	Mariani Damai	24
25	CABANG-039	PHI Mini Market - Makassar 01	039-156	Niken Setiawan	25
26	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-105	Sentosa Indrawan	26
27	CABANG-065	PHI Mini Market - Surabaya 01	065-206	Aris Siberut	27
28	CABANG-039	PHI Mini Market - Makassar 01	039-084	Eriq Jagat	28
29	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-128	Cahya Terang	29
30	CABANG-065	PHI Mini Market - Surabaya 01	065-078	Aron Zaminski	30

 Close

11. Agar kolom **sk_karyawan** bisa berada di urutan pertama (paling kiri), kita gunakan step **Select values** untuk mengurnyanya.



12. Periksa perubahannya dengan klik kanan **Preview >> Quick Launch**.

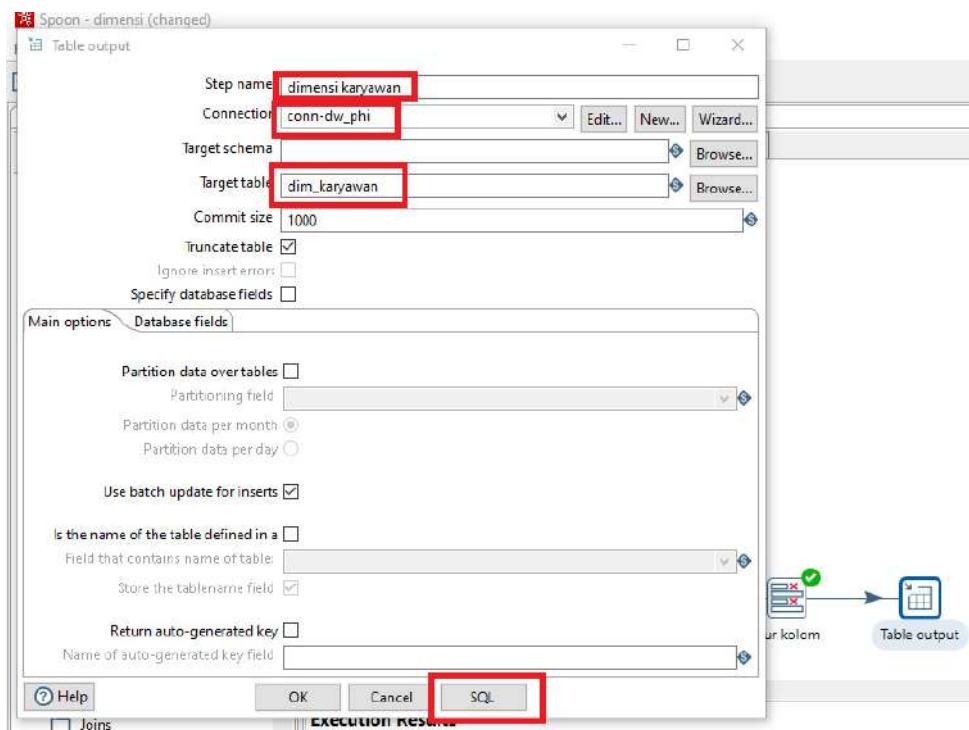
 Examine preview data

Rows of step: atur kolom (30 rows)

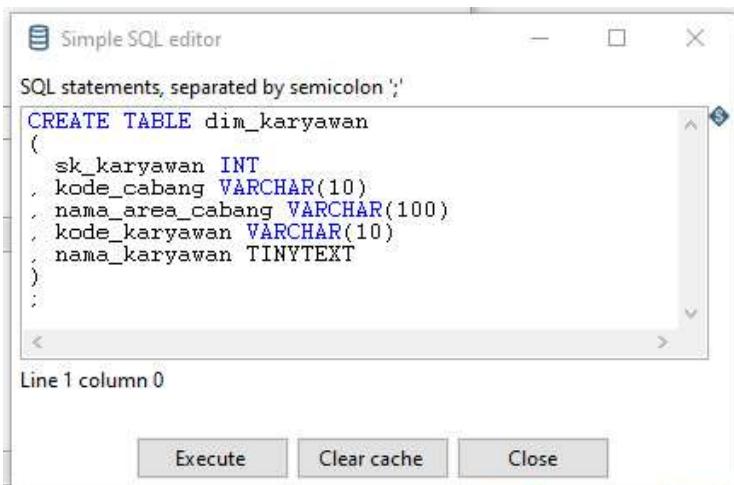
#	sk_karyawan	kode_cabang	nama_area_cabang	kode_karyawan	nama_karyawan
1	1	CABANG-039	PHI Mini Market - Makassar 01	039-147	Bintang Maven
2	2	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-181	Eria Setiawan
3	3	CABANG-065	PHI Mini Market - Surabaya 01	065-282	Galang Setiawan
4	4	CABANG-039	PHI Mini Market - Makassar 01	039-031	Kristina Damai
5	5	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-075	Eko Rukun
6	6	CABANG-065	PHI Mini Market - Surabaya 01	065-076	Natali Menawan
7	7	CABANG-039	PHI Mini Market - Makassar 01	039-214	Mawar Mardi
8	8	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-055	Erman Margo
9	9	CABANG-065	PHI Mini Market - Surabaya 01	065-061	Ayu Pekerti
10	10	CABANG-039	PHI Mini Market - Makassar 01	039-044	Ferdy Tenteram
11	11	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-133	Harum Maven
12	12	CABANG-065	PHI Mini Market - Surabaya 01	065-023	Harum Selangit
13	13	CABANG-039	PHI Mini Market - Makassar 01	039-212	Agus Dewangga
14	14	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-031	Kristina Damai
15	15	CABANG-065	PHI Mini Market - Surabaya 01	065-060	Mulia Setiawan
16	16	CABANG-039	PHI Mini Market - Makassar 01	039-053	Galang Terang
17	17	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-244	Budiwati Ramah
18	18	CABANG-065	PHI Mini Market - Surabaya 01	065-007	Budi Tenteram
19	19	CABANG-039	PHI Mini Market - Makassar 01	039-127	Lastri Mardi
20	20	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-286	Kusuma Dominik
21	21	CABANG-065	PHI Mini Market - Surabaya 01	065-258	Mulyo Damai
22	22	CABANG-039	PHI Mini Market - Makassar 01	039-203	Eriq Menawan
23	23	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-006	Agung Alexander
24	24	CABANG-065	PHI Mini Market - Surabaya 01	065-094	Mariani Damai
25	25	CABANG-039	PHI Mini Market - Makassar 01	039-156	Niken Setiawan
26	26	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-105	Sentosa Indrawan
27	27	CABANG-065	PHI Mini Market - Surabaya 01	065-206	Aris Siberut
28	28	CABANG-039	PHI Mini Market - Makassar 01	039-084	Eriq Jagat
29	29	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-128	Cahya Terang
30	30	CABANG-065	PHI Mini Market - Surabaya 01	065-078	Aron Zaminski

 Close

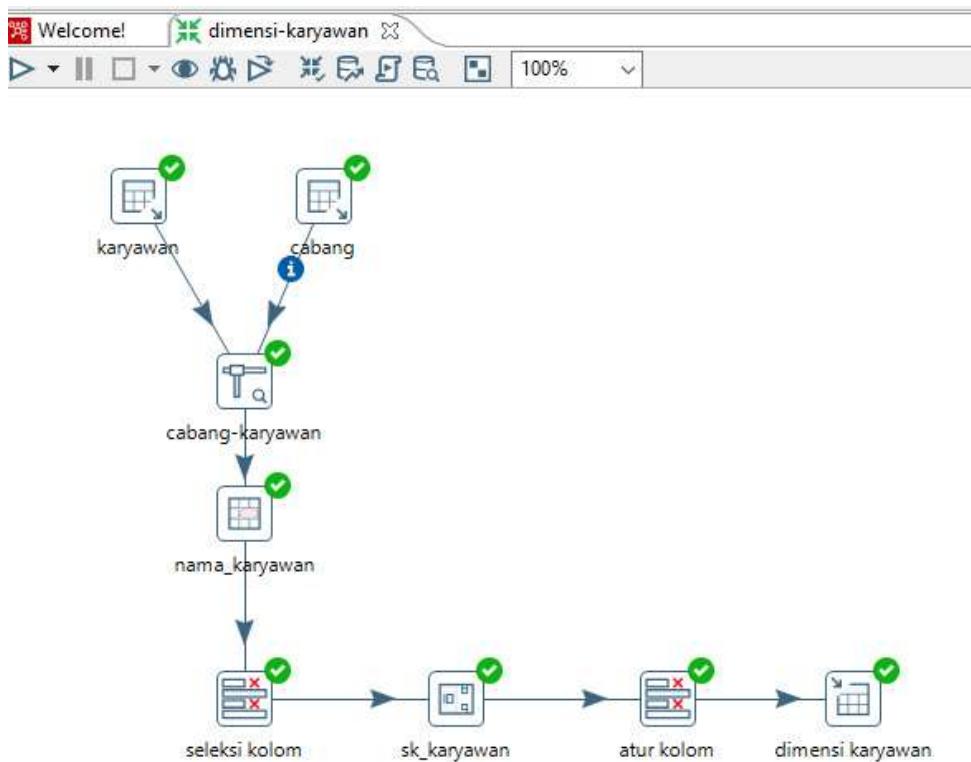
13. Langkah berikutnya kita simpan dalam database **dw_phi** dengan nama tabel **dim_karyawan**, menggunakan step **Table output**.



14. Tekan tombol **SQL** dan **Execute**.



15. Jangan lupa simpan file dan tekan tombol **Run** untuk mengeksekusi.



16. Tabel **dim_karyawan** sudah terbentuk pada database **dw_phi**.

← Server: 127.0.0.1 » Basis data: dw_phi » Tabel: dim_karyawan

Jelajahi Struktur SQL Cari Tambahan Eksport Impor

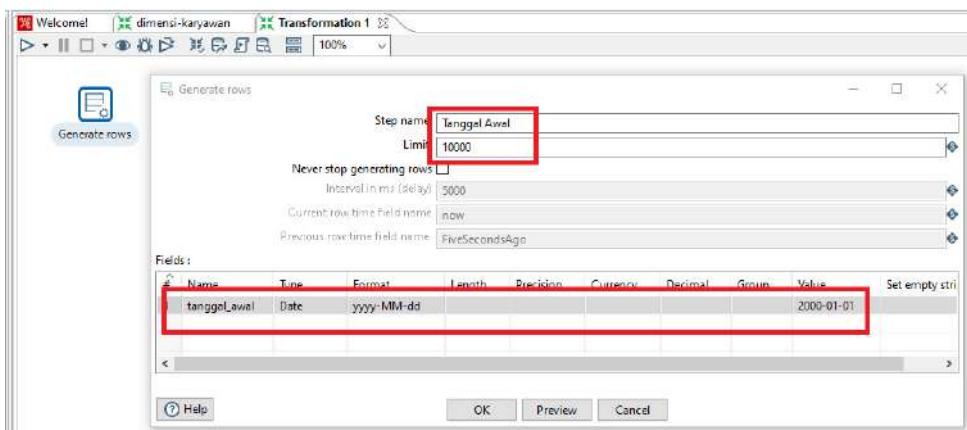
1 > >> Tampilkan semua Jumlah baris: 25 Saring baris: Cari di tabel ini

+ Opsi

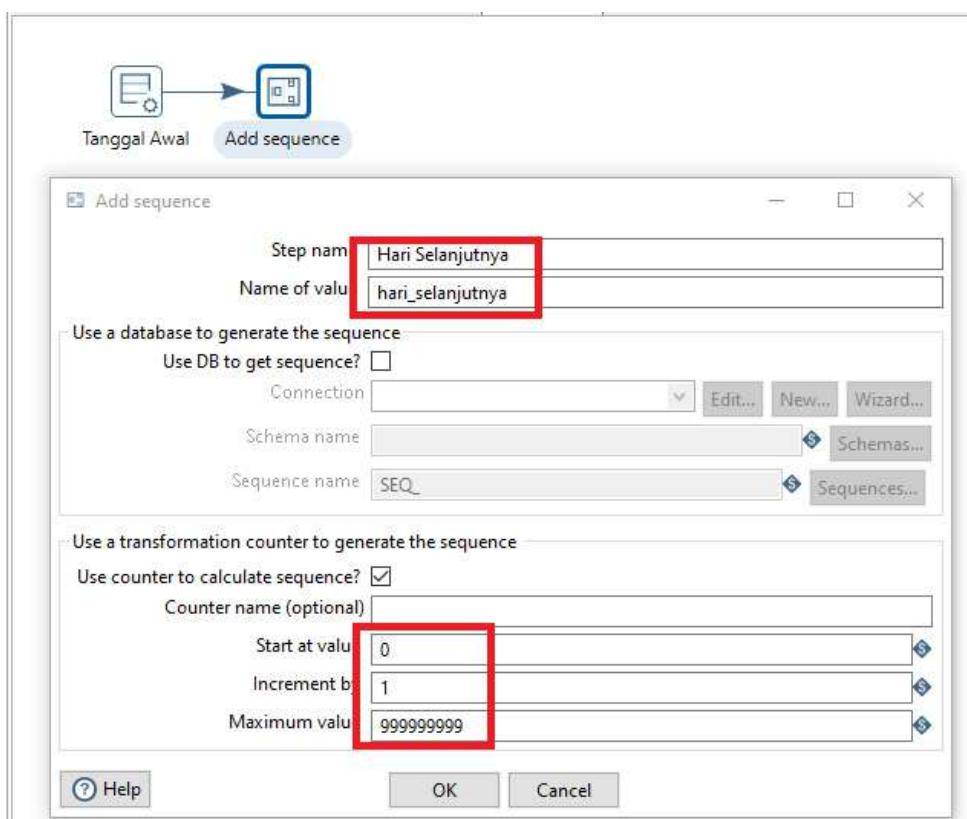
sk_karyawan	kode_cabang	nama_area_cabang	kode_karyawan	nama_karyawan
1	CABANG-039	PHI Mini Market - Makassar 01	039-147	Bintang Maven
2	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-181	Eria Setiawan
3	CABANG-065	PHI Mini Market - Surabaya 01	065-282	Galang Setiawan
4	CABANG-039	PHI Mini Market - Makassar 01	039-031	Kristina Damai
5	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-075	Eko Rukun
6	CABANG-065	PHI Mini Market - Surabaya 01	065-076	Natali Menawan
7	CABANG-039	PHI Mini Market - Makassar 01	039-214	Mawar Mardi
8	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-055	Erman Margo
9	CABANG-065	PHI Mini Market - Surabaya 01	065-061	Ayu Pekerti
10	CABANG-039	PHI Mini Market - Makassar 01	039-044	Ferdy Tenteram
11	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-133	Harum Maven
12	CABANG-065	PHI Mini Market - Surabaya 01	065-023	Harum Selangit
13	CABANG-039	PHI Mini Market - Makassar 01	039-212	Agus Dewangga
14	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-031	Kristina Damai
15	CABANG-065	PHI Mini Market - Surabaya 01	065-060	Mulia Setiawan
16	CABANG-039	PHI Mini Market - Makassar 01	039-053	Galang Terang
17	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-244	Budiwati Ramah
18	CABANG-065	PHI Mini Market - Surabaya 01	065-007	Budi Tenteram
19	CABANG-039	PHI Mini Market - Makassar 01	039-127	Lastri Mardi
20	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-286	Kusuma Dominik
21	CABANG-065	PHI Mini Market - Surabaya 01	065-258	Mulyo Damai
22	CABANG-039	PHI Mini Market - Makassar 01	039-203	Eriq Menawan
23	CABANG-047	PHI Mini Market - Jakarta Pusat 01	047-006	Agung Alexander
24	CABANG-065	PHI Mini Market - Surabaya 01	065-094	Mariani Damai
25	CABANG-039	PHI Mini Market - Makassar 01	039-156	Niken Setiawan

3.4.3 Tabel Dimensi Waktu

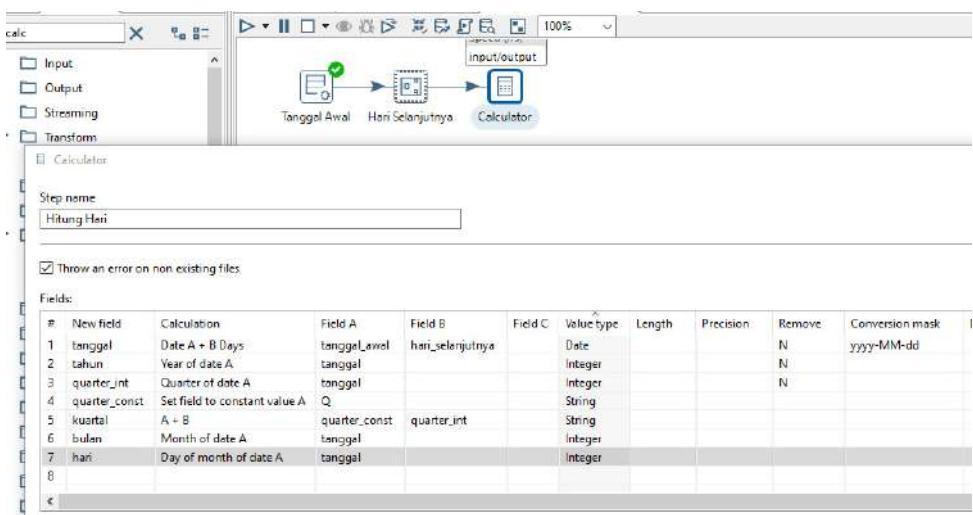
- Buat transformasi baru, kemudian cari step Generate rows.



2. Tambahkan step Add sequence.



3. Tambahkan step **Calculator**. Isi fields seperti di bawah ini. Periksa dengan **Preview >> Quick Launch**.



 Examine preview data

Rows of step: Hitung Hari (1000 rows)

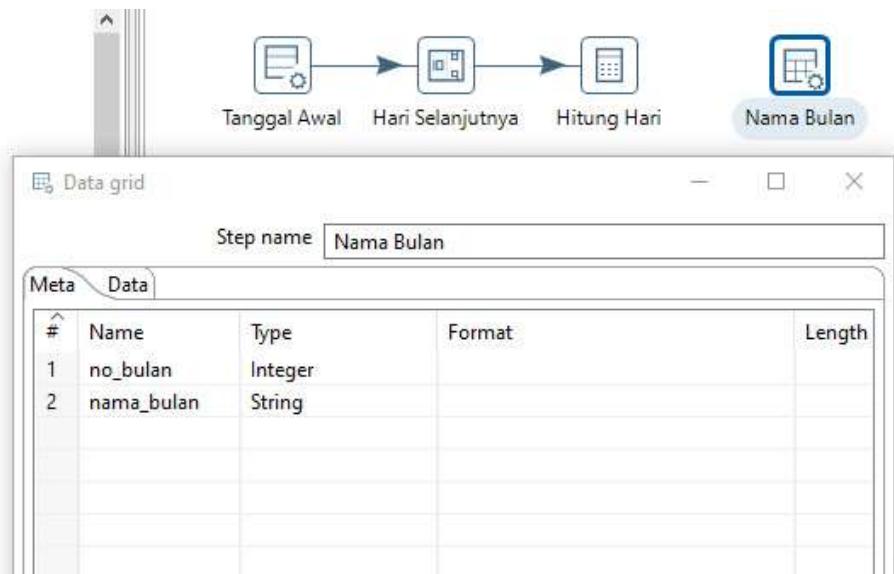
#	tanggal_awal	hari_selanjutnya	tanggal	tahun	quarter_int	quarter_const	kuartal	bulan	hari
1	2000-01-01	0	2000-01-01	2000	1	Q	Q1	1	1
2	2000-01-01	1	2000-01-02	2000	1	Q	Q1	1	2
3	2000-01-01	2	2000-01-03	2000	1	Q	Q1	1	3
4	2000-01-01	3	2000-01-04	2000	1	Q	Q1	1	4
5	2000-01-01	4	2000-01-05	2000	1	Q	Q1	1	5
6	2000-01-01	5	2000-01-06	2000	1	Q	Q1	1	6
7	2000-01-01	6	2000-01-07	2000	1	Q	Q1	1	7
8	2000-01-01	7	2000-01-08	2000	1	Q	Q1	1	8
9	2000-01-01	8	2000-01-09	2000	1	Q	Q1	1	9
10	2000-01-01	9	2000-01-10	2000	1	Q	Q1	1	10
11	2000-01-01	10	2000-01-11	2000	1	Q	Q1	1	11
12	2000-01-01	11	2000-01-12	2000	1	Q	Q1	1	12
13	2000-01-01	12	2000-01-13	2000	1	Q	Q1	1	13
14	2000-01-01	13	2000-01-14	2000	1	Q	Q1	1	14
15	2000-01-01	14	2000-01-15	2000	1	Q	Q1	1	15
16	2000-01-01	15	2000-01-16	2000	1	Q	Q1	1	16
17	2000-01-01	16	2000-01-17	2000	1	Q	Q1	1	17
18	2000-01-01	17	2000-01-18	2000	1	Q	Q1	1	18
19	2000-01-01	18	2000-01-19	2000	1	Q	Q1	1	19
20	2000-01-01	19	2000-01-20	2000	1	Q	Q1	1	20
21	2000-01-01	20	2000-01-21	2000	1	Q	Q1	1	21
22	2000-01-01	21	2000-01-22	2000	1	Q	Q1	1	22
23	2000-01-01	22	2000-01-23	2000	1	Q	Q1	1	23
24	2000-01-01	23	2000-01-24	2000	1	Q	Q1	1	24
25	2000-01-01	24	2000-01-25	2000	1	Q	Q1	1	25
26	2000-01-01	25	2000-01-26	2000	1	Q	Q1	1	26
27	2000-01-01	26	2000-01-27	2000	1	Q	Q1	1	27
28	2000-01-01	27	2000-01-28	2000	1	Q	Q1	1	28
29	2000-01-01	28	2000-01-29	2000	1	Q	Q1	1	29
30	2000-01-01	29	2000-01-30	2000	1	Q	Q1	1	30
31	2000-01-01	30	2000-01-31	2000	1	Q	Q1	1	31

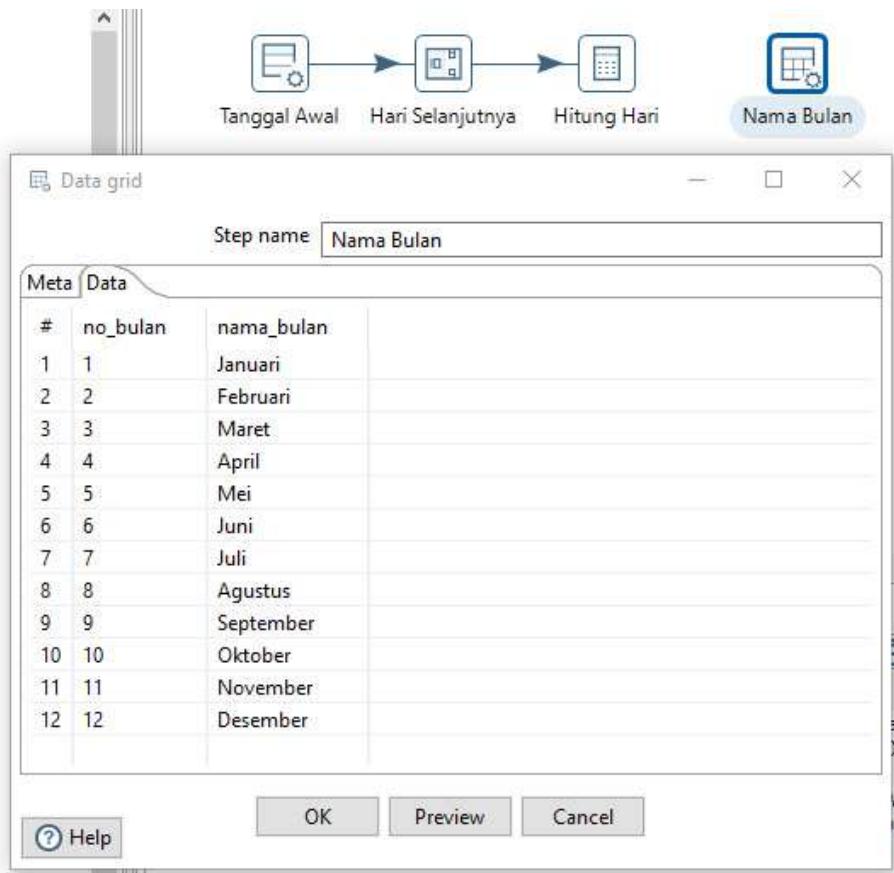
[Close](#)

[Stop](#)

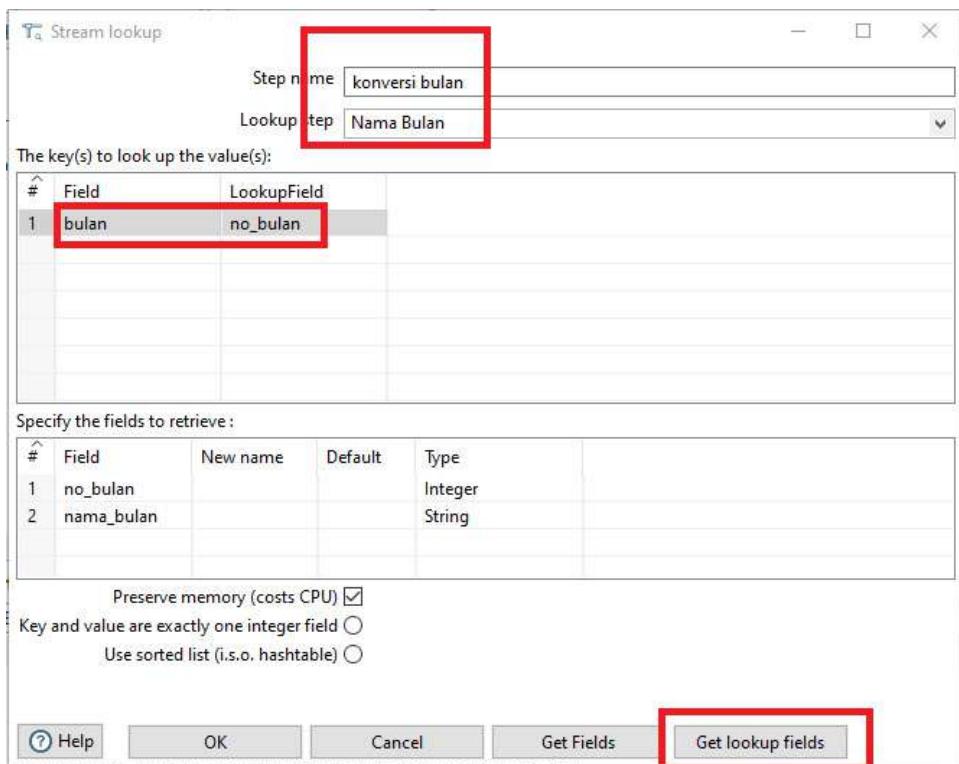
[Get more rows](#)

4. Selanjutnya konversi bulan dari integer menjadi string. Gunakan step **Data grid**.





5. Hubungkan step Hitung Hari dengan Nama Bulan menggunakan step **Stream lookup**.



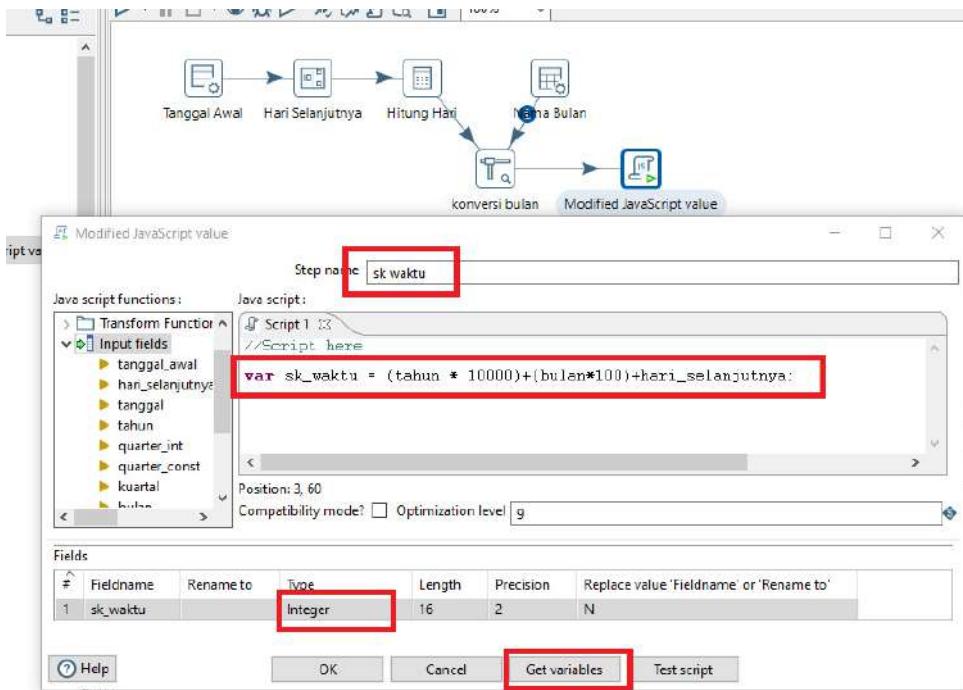
 Examine preview data

Rows of step: konversi_bulan (1000 rows)

#	tanggal_awal	hari_selanjutnya	tanggal	tahun	quarter_int	quarter_const	kuartal	bulan	hari	no_bulan	nama_bulan
22	2000-01-01		21	2000-01-22	2000	1 Q	Q1	1	22	1	Januari
23	2000-01-01		22	2000-01-23	2000	1 Q	Q1	1	23	1	Januari
24	2000-01-01		23	2000-01-24	2000	1 Q	Q1	1	24	1	Januari
25	2000-01-01		24	2000-01-25	2000	1 Q	Q1	1	25	1	Januari
26	2000-01-01		25	2000-01-26	2000	1 Q	Q1	1	26	1	Januari
27	2000-01-01		26	2000-01-27	2000	1 Q	Q1	1	27	1	Januari
28	2000-01-01		27	2000-01-28	2000	1 Q	Q1	1	28	1	Januari
29	2000-01-01		28	2000-01-29	2000	1 Q	Q1	1	29	1	Januari
30	2000-01-01		29	2000-01-30	2000	1 Q	Q1	1	30	1	Januari
31	2000-01-01		30	2000-01-31	2000	1 Q	Q1	1	31	1	Januari
32	2000-01-01		31	2000-02-01	2000	1 Q	Q1	2	1	2	Februari
33	2000-01-01		32	2000-02-02	2000	1 Q	Q1	2	2	2	Februari
34	2000-01-01		33	2000-02-03	2000	1 Q	Q1	2	3	2	Februari
35	2000-01-01		34	2000-02-04	2000	1 Q	Q1	2	4	2	Februari
36	2000-01-01		35	2000-02-05	2000	1 Q	Q1	2	5	2	Februari
37	2000-01-01		36	2000-02-06	2000	1 Q	Q1	2	6	2	Februari
38	2000-01-01		37	2000-02-07	2000	1 Q	Q1	2	7	2	Februari
39	2000-01-01		38	2000-02-08	2000	1 Q	Q1	2	8	2	Februari
40	2000-01-01		39	2000-02-09	2000	1 Q	Q1	2	9	2	Februari
41	2000-01-01		40	2000-02-10	2000	1 Q	Q1	2	10	2	Februari
42	2000-01-01		41	2000-02-11	2000	1 Q	Q1	2	11	2	Februari
43	2000-01-01		42	2000-02-12	2000	1 Q	Q1	2	12	2	Februari
44	2000-01-01		43	2000-02-13	2000	1 Q	Q1	2	13	2	Februari
45	2000-01-01		44	2000-02-14	2000	1 Q	Q1	2	14	2	Februari
46	2000-01-01		45	2000-02-15	2000	1 Q	Q1	2	15	2	Februari
47	2000-01-01		46	2000-02-16	2000	1 Q	Q1	2	16	2	Februari
48	2000-01-01		47	2000-02-17	2000	1 Q	Q1	2	17	2	Februari
49	2000-01-01		48	2000-02-18	2000	1 Q	Q1	2	18	2	Februari
50	2000-01-01		49	2000-02-19	2000	1 Q	Q1	2	19	2	Februari
51	2000-01-01		50	2000-02-20	2000	1 Q	Q1	2	20	2	Februari
52	2000-01-01		51	2000-02-21	2000	1 Q	Q1	2	21	2	Februari

[Close](#) [Stop](#) [Get more rows](#)

6. Selanjutnya membuat kolom **sk_waktu**. Kita membutuhkan step **Modified JavaScript value**.



```
var sk_waktu = (tahun * 10000) + (bulan * 100) + hari_selanjutnya;
```

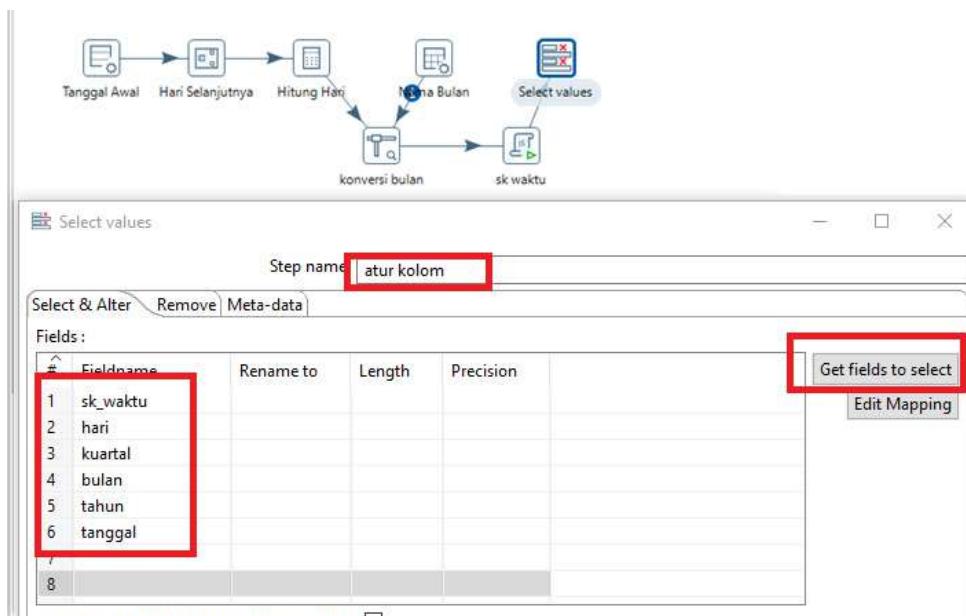
7. Lihat dengan Preview >> Quick Launch.

Examine preview data

Rows of step: sk_waktu (1000 rows)

#	tanggal_awal	hari_selanjutnya	tanggal	tahun	quarter_int	quarter_const	kuartal	bulan	hari	no_bulan	nama_bulan	sk_waktu
1	2000-01-01	0	2000-01-01	2000	1	Q	Q1	1	1	1	Januari	20000100
2	2000-01-01	1	2000-01-02	2000	1	Q	Q1	1	2	1	Januari	20000101
3	2000-01-01	2	2000-01-03	2000	1	Q	Q1	1	3	1	Januari	20000102
4	2000-01-01	3	2000-01-04	2000	1	Q	Q1	1	4	1	Januari	20000103
5	2000-01-01	4	2000-01-05	2000	1	Q	Q1	1	5	1	Januari	20000104
6	2000-01-01	5	2000-01-06	2000	1	Q	Q1	1	6	1	Januari	20000105
7	2000-01-01	6	2000-01-07	2000	1	Q	Q1	1	7	1	Januari	20000106
8	2000-01-01	7	2000-01-08	2000	1	Q	Q1	1	8	1	Januari	20000107
9	2000-01-01	8	2000-01-09	2000	1	Q	Q1	1	9	1	Januari	20000108
10	2000-01-01	9	2000-01-10	2000	1	Q	Q1	1	10	1	Januari	20000109
11	2000-01-01	10	2000-01-11	2000	1	Q	Q1	1	11	1	Januari	20000110
12	2000-01-01	11	2000-01-12	2000	1	Q	Q1	1	12	1	Januari	20000111
13	2000-01-01	12	2000-01-13	2000	1	Q	Q1	1	13	1	Januari	20000112
14	2000-01-01	13	2000-01-14	2000	1	Q	Q1	1	14	1	Januari	20000113
15	2000-01-01	14	2000-01-15	2000	1	Q	Q1	1	15	1	Januari	20000114
16	2000-01-01	15	2000-01-16	2000	1	O	Q1	1	16	1	Januari	

8. Langkah selanjutnya adalah mengatur posisi kolom. Gunakan step **Select values**.

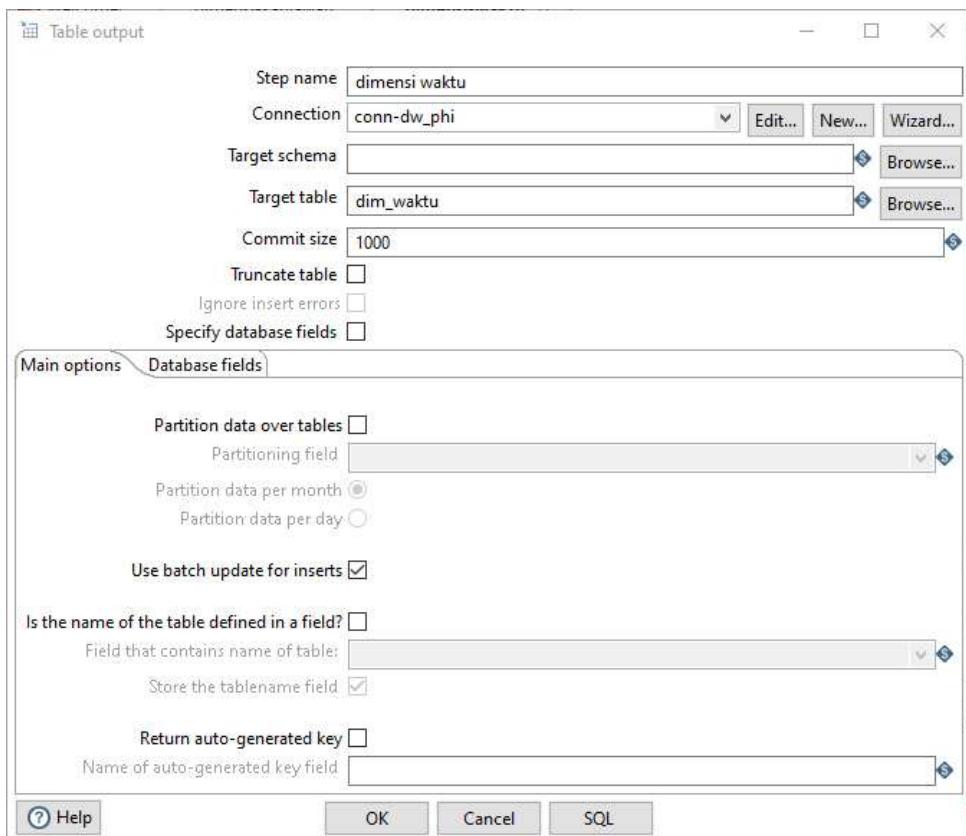




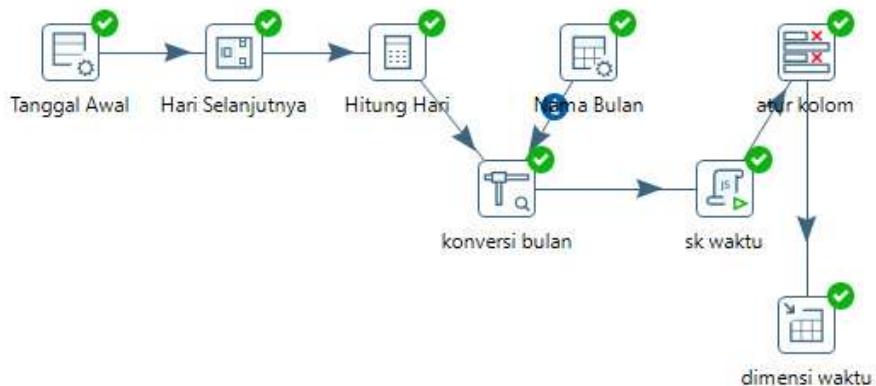
Rows of step: atur kolom (1000 rows)

#	sk_waktu	hari	kuartal	bulan	tahun	tanggal
1	20000100	1	Q1	1	2000	2000-01-01
2	20000101	2	Q1	1	2000	2000-01-02
3	20000102	3	Q1	1	2000	2000-01-03
4	20000103	4	Q1	1	2000	2000-01-04
5	20000104	5	Q1	1	2000	2000-01-05
6	20000105	6	Q1	1	2000	2000-01-06
7	20000106	7	Q1	1	2000	2000-01-07
8	20000107	8	Q1	1	2000	2000-01-08
9	20000108	9	Q1	1	2000	2000-01-09
10	20000109	10	Q1	1	2000	2000-01-10
11	20000110	11	Q1	1	2000	2000-01-11
12	20000111	12	Q1	1	2000	2000-01-12
13	20000112	13	Q1	1	2000	2000-01-13
14	20000113	14	Q1	1	2000	2000-01-14
15	20000114	15	Q1	1	2000	2000-01-15
16	20000115	16	Q1	1	2000	2000-01-16
17	20000116	17	Q1	1	2000	2000-01-17
18	20000117	18	Q1	1	2000	2000-01-18
19	20000118	19	Q1	1	2000	2000-01-19
20	20000119	20	Q1	1	2000	2000-01-20
21	20000120	21	Q1	1	2000	2000-01-21
22	20000121	22	Q1	1	2000	2000-01-22
23	20000122	23	Q1	1	2000	2000-01-23
24	20000123	24	Q1	1	2000	2000-01-24
25	20000124	25	Q1	1	2000	2000-01-25
26	20000125	26	Q1	1	2000	2000-01-26
27	20000126	27	Q1	1	2000	2000-01-27
28	20000127	28	Q1	1	2000	2000-01-28
29	20000128	29	Q1	1	2000	2000-01-29
30	20000129	30	Q1	1	2000	2000-01-30
31	20000130	31	Q1	1	2000	2000-01-31

9. Langkah berikutnya kita simpan dalam database **dw_phi** dengan nama tabel **dim_waktu**, menggunakan step **Table output**.



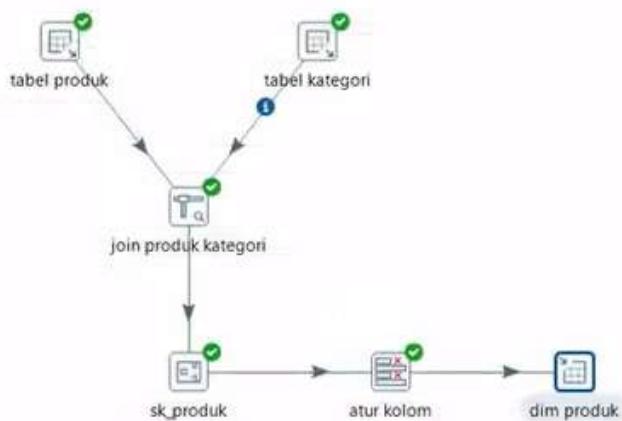
10. Jangan lupa simpan file dan tekan tombol **Run** untuk mengeksekusi.



3.5 Tugas

Selesaikan pembuatan tabel dimensi untuk **dim_produk** dan **dim_cabang** sesuai star schema PHI-minimart di kelas. Jika belum selesai, bisa dilanjutkan di rumah dan dikumpulkan sebelum pertemuan berikutnya sesuai arahan dosen pengampu. Penyelesaian tugas ini sekaligus menjadi prasyarat dalam mengerjakan kegiatan praktikum di modul berikutnya.

1. Rancangan Tabel Dimensi Produk



2. Rancangan Tabel Dimensi Cabang



Modul 4

Tabel Fakta

4.1 Tujuan

1. Mahasiswa mampu membangun tabel fakta yang menjadi tabel pusat transaksi dalam sebuah *data warehouse*.
2. Mahasiswa mampu melakukan proses ETL secara lebih lanjut pada pengembangan sebuah *data warehouse*.

4.2 Landasan Teori

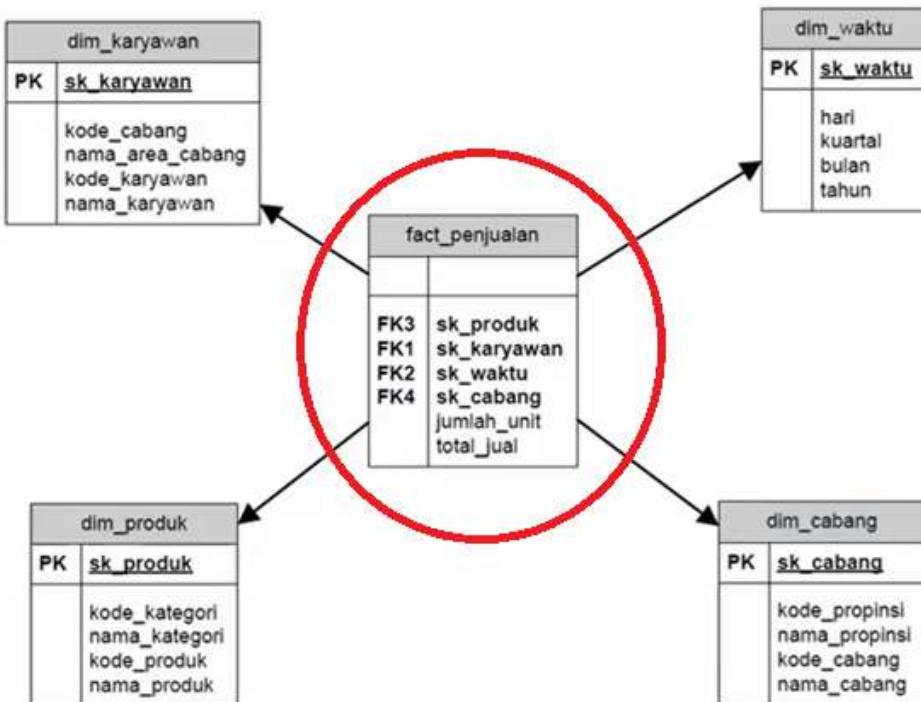
Fact table (tabel fakta) adalah tabel yang umumnya mengandung sesuatu yang dapat diukur (*measure*) seperti harga, jumlah barang dan sebagainya. *Fact table* juga merupakan kumpulan *foreign key* dari *primary key* yang terdapat pada masing-masing *dimension table*. *Fact table* juga mengandung data yang historis.

4.3 Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi XAMPP.
3. Program aplikasi Pentaho Data Integration.
4. Modul Praktikum Data Warehousing dan Data Mining.

4.4 Langkah-langkah Praktikum

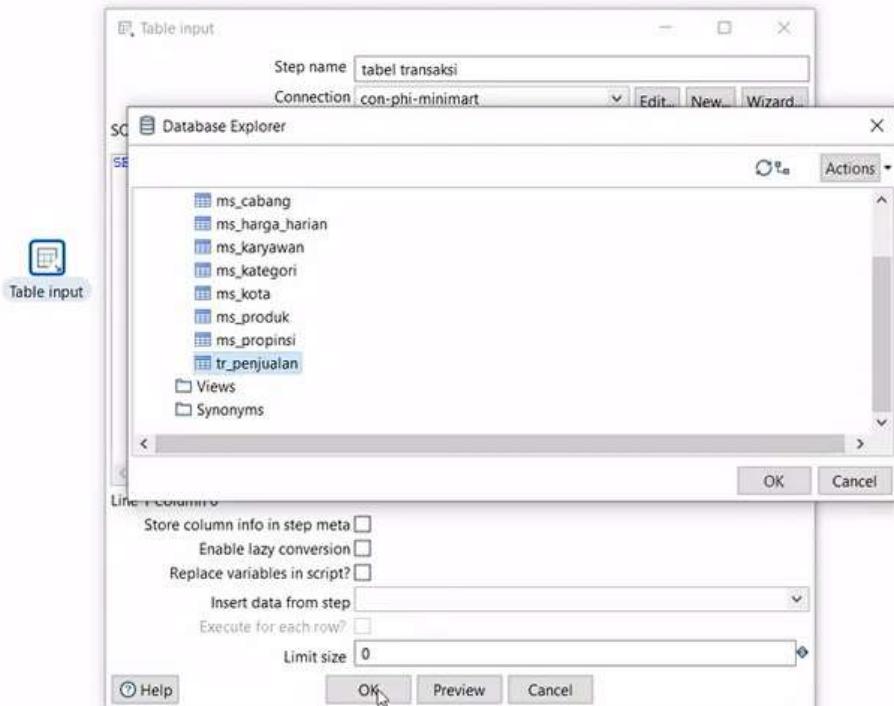
Pada praktikum ini, kita masih menggunakan database **phi_minimart** serta **dw_phi**. Melanjutkan membuat tabel fakta penjualan berdasarkan tabel dimensi yang sudah dibuat pada percobaan di modul sebelumnya.



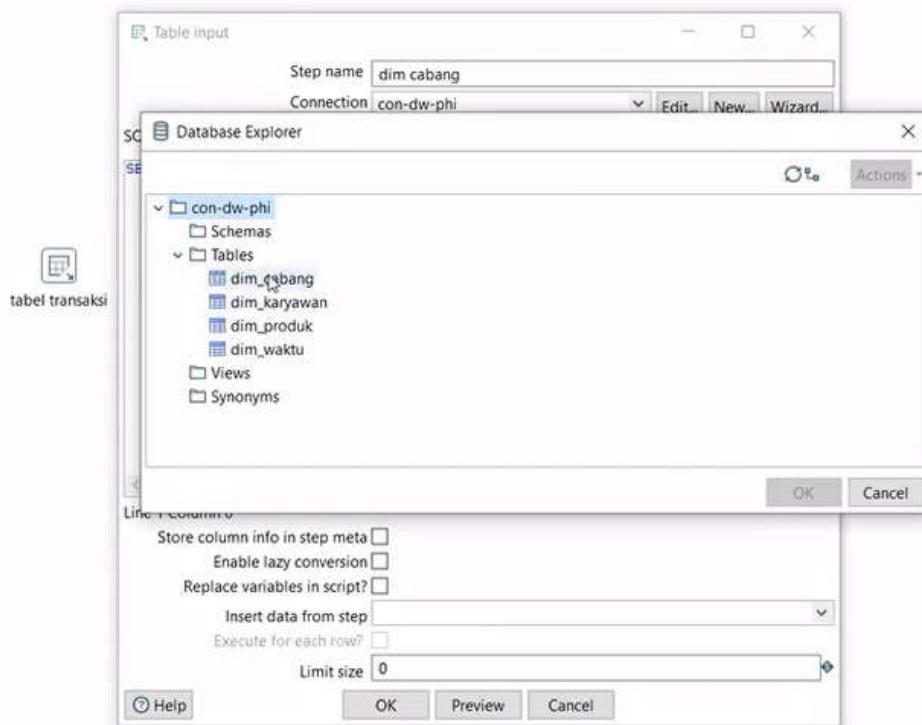
Berdasarkan gambar di atas, isi / *field* dari tabel fakta penjualan antara lain **sk_produk** yang diambil dari tabel dimensi produk, **sk_karyawan** diambil dari tabel dimensi karyawan, **sk_waktu** diambil dari tabel dimensi waktu, **sk_cabang** diambil dari tabel dimensi cabang, ditambah kolom baru yaitu **jumlah_unit** dan **total_jual**.

Berikut ini langkah-langkah dalam membuat tabel fakta penjualan :

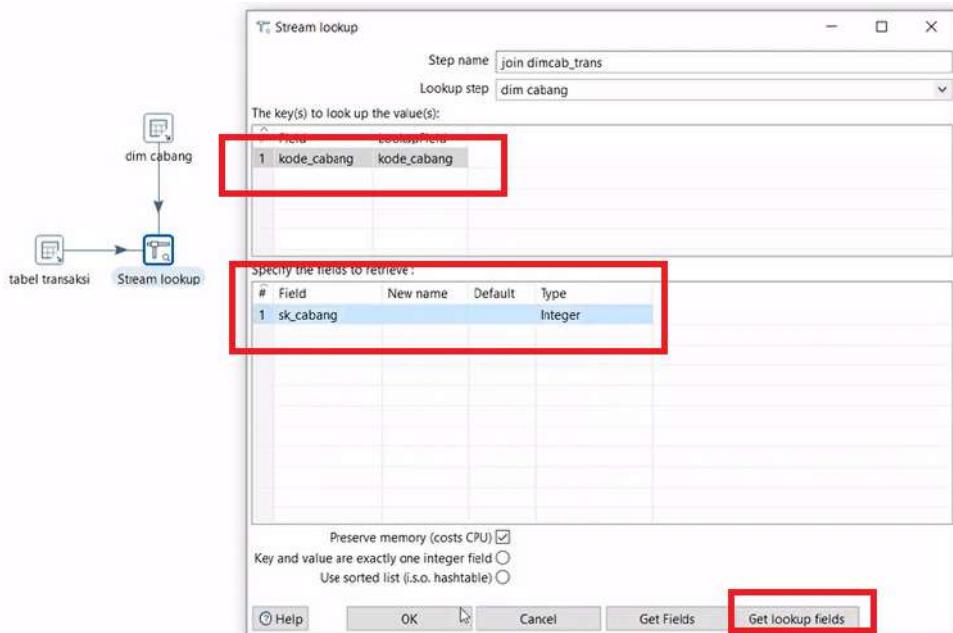
- Buat transformasi baru, tarik step **Table input** ke canvas. Edit seperti berikut ini.



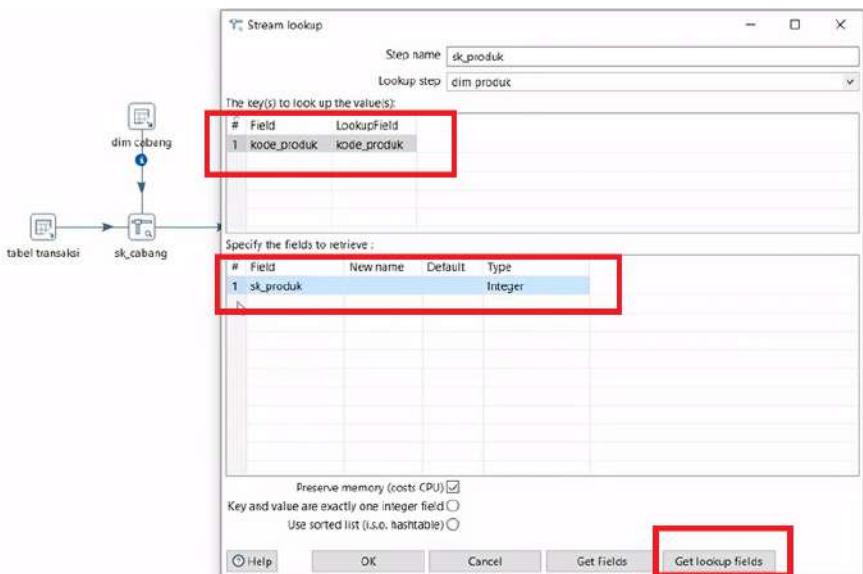
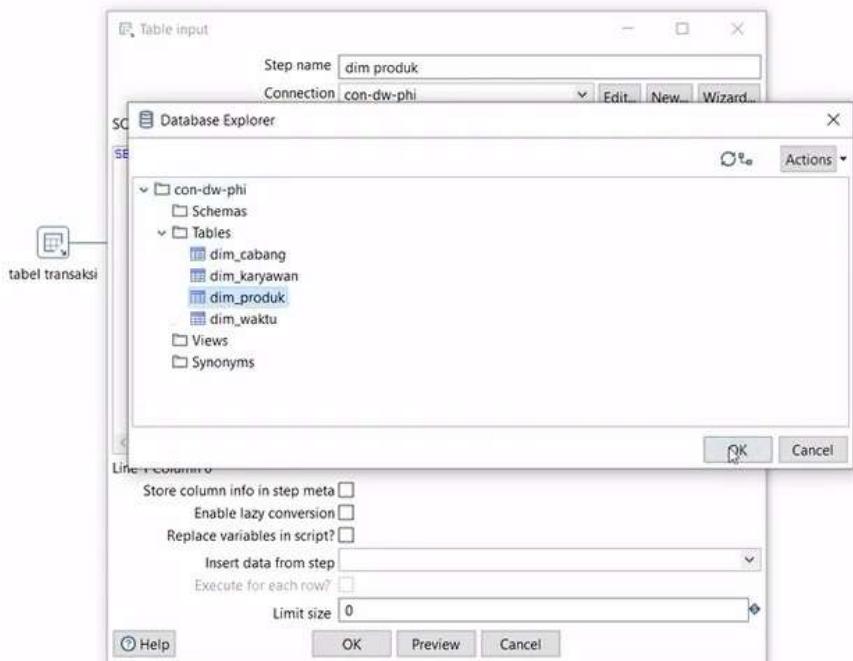
- Selanjutnya hubungkan ke tabel dimensi cabang dengan cara *drag and drop* step **Table input** ke canvas.



3. Tujuannya adalah untuk mengambil kolom **sk_cabang** di tabel **dim_cabang**. Untuk menghubungkannya bisa kita gunakan step **Stream lookup**.



4. Selanjutnya menghubungkan tabel dimensi produk untuk mengambil field **sk_produk**. Langkah-langkahnya sama seperti sebelumnya.





Hasilnya bisa dilihat dengan klik kanan **Preview >> Quick Launch**.

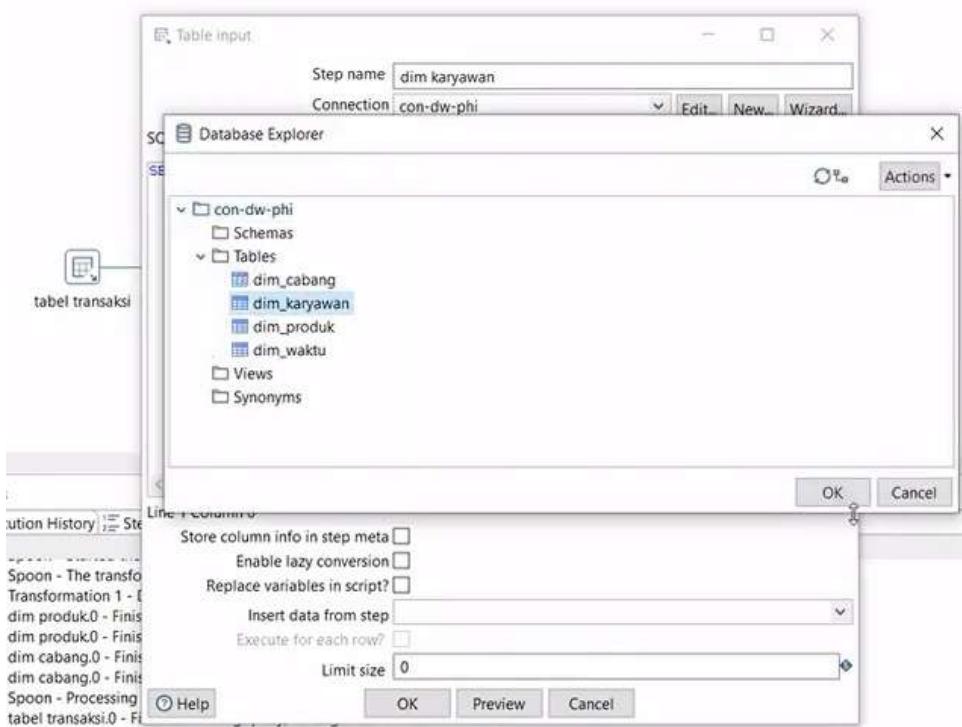
Examine preview data

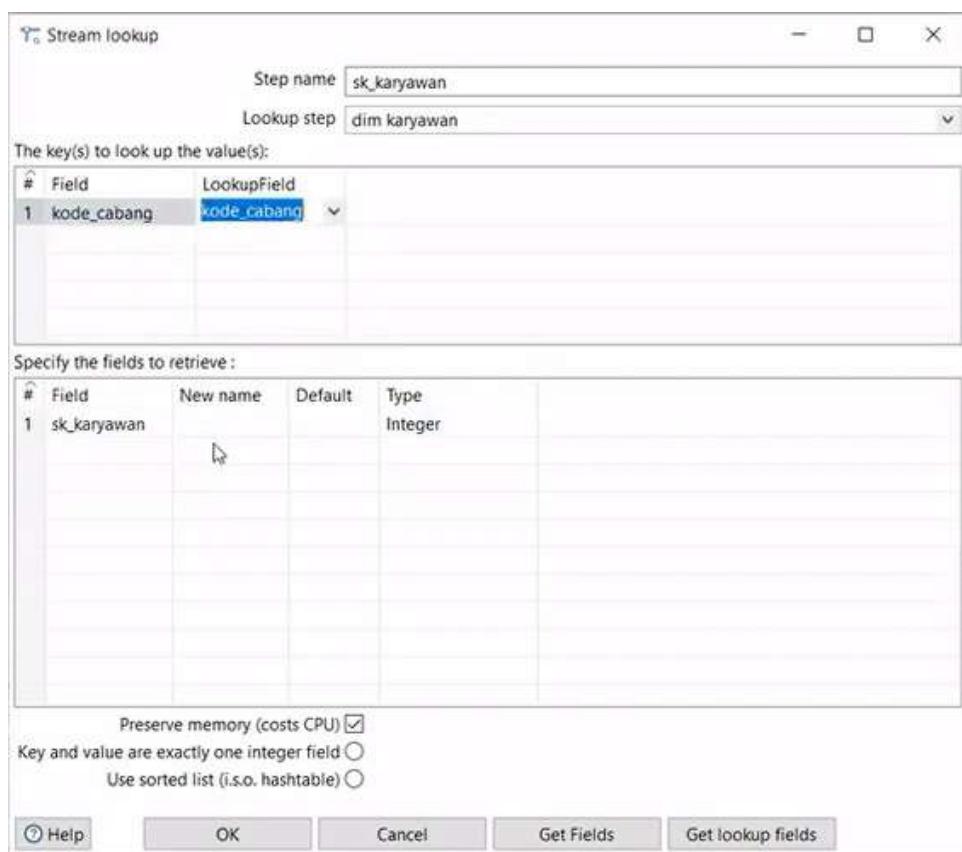
Rows of step: sk_produk (1000 rows)

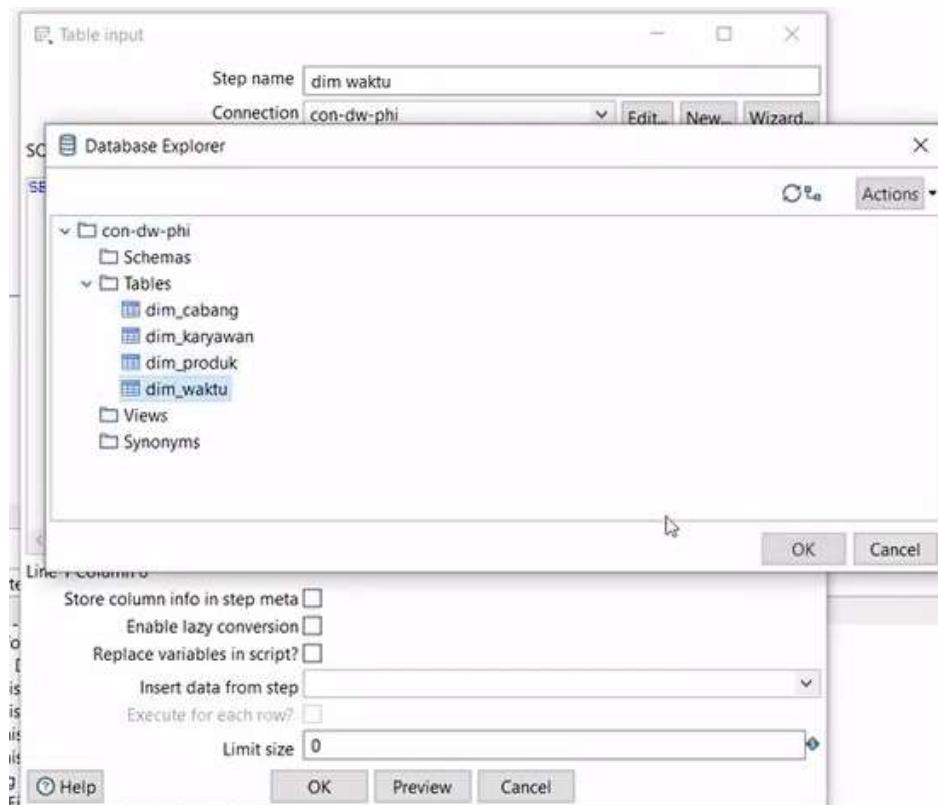
#	rgl_transaksi	kode_cabang	kode_kasir	kode_item	kode_produk	jumlah_pembelian	sk_cabang	sk_produk
1	2008/01/01 00:00:00.000000000	CABANG-039	039-053	ITM-038	PROD-0000040	12	39	40
2	2008/01/01 00:00:00.000000000	CABANG-039	039-127	ITM-020	PROD-0000023	16	39	23
3	2008/01/01 00:00:00.000000000	CABANG-039	039-156	ITM-017	PROD-0000020	12	39	20
4	2008/01/01 00:00:00.000000000	CABANG-039	039-212	ITM-002	PROD-0000002	11	39	2
5	2008/01/01 00:00:00.000000000	CABANG-039	039-044	ITM-034	PROD-0000036	14	39	36
6	2008/01/01 00:00:00.000000000	CABANG-039	039-156	ITM-023	PROD-0000015	9	39	15
7	2008/01/01 00:00:00.000000000	CABANG-039	039-203	ITM-020	PROD-0000023	20	39	23
8	2008/01/01 00:00:00.000000000	CABANG-039	039-053	ITM-021	PROD-0000024	9	39	24
9	2008/01/01 00:00:00.000000000	CABANG-039	039-203	ITM-015	PROD-0000018	3	39	18
10	2008/01/01 00:00:00.000000000	CABANG-039	039-147	ITM-005	PROD-0000006	9	39	6
11	2008/01/01 00:00:00.000000000	CABANG-039	039-084	ITM-035	PROD-0000037	10	39	37
12	2008/01/01 00:00:00.000000000	CABANG-039	039-212	ITM-005	PROD-0000006	18	39	6
13	2008/01/01 00:00:00.000000000	CABANG-039	039-147	ITM-034	PROD-0000036	5	39	36
14	2008/01/01 00:00:00.000000000	CABANG-039	039-203	ITM-022	PROD-0000025	8	39	25
15	2008/01/01 00:00:00.000000000	CABANG-039	039-031	ITM-007	PROD-0000007	14	39	7
16	2008/01/01 00:00:00.000000000	CABANG-039	039-053	ITM-022	PROD-0000025	18	39	25
17	2008/01/01 00:00:00.000000000	CABANG-039	039-212	ITM-038	PROD-0000040	18	39	40
18	2008/01/01 00:00:00.000000000	CABANG-039	039-053	ITM-009	PROD-0000009	5	39	9
19	2008/01/01 00:00:00.000000000	CABANG-039	039-212	ITM-012	PROD-0000012	9	39	12
20	2008/01/01 00:00:00.000000000	CABANG-039	039-212	ITM-023	PROD-0000014	9	39	14
21	2008/01/01 00:00:00.000000000	CABANG-039	039-044	ITM-017	PROD-0000020	6	39	20
22	2008/01/01 00:00:00.000000000	CABANG-039	039-156	ITM-007	PROD-0000007	8	39	7
23	2008/01/01 00:00:00.000000000	CABANG-039	039-147	ITM-009	PROD-0000009	2	39	9
24	2008/01/01 00:00:00.000000000	CABANG-039	039-044	ITM-019	PROD-0000022	20	39	22
25	2008/01/01 00:00:00.000000000	CABANG-039	039-035	ITM-0037	PROD-0000037	6	39	37
26	2008/01/01 00:00:00.000000000	CABANG-039	039-156	ITM-015	PROD-0000018	3	39	18
27	2008/01/01 00:00:00.000000000	CABANG-039	039-127	ITM-015	PROD-0000018	18	39	18
28	2008/01/01 00:00:00.000000000	CABANG-039	039-031	ITM-033	PROD-0000035	7	39	35
29	2008/01/01 00:00:00.000000000	CABANG-039	039-127	ITM-002	PROD-0000002	18	39	2
30	2008/01/01 00:00:00.000000000	CABANG-039	039-053	ITM-035	PROD-0000038	6	39	38
31	2008/01/01 00:00:00.000000000	CABANG-039	039-127	ITM-012	PROD-0000012	18	39	12
32	2008/01/01 00:00:00.000000000	CABANG-039	039-127	ITM-037	PROD-0000039	17	39	39

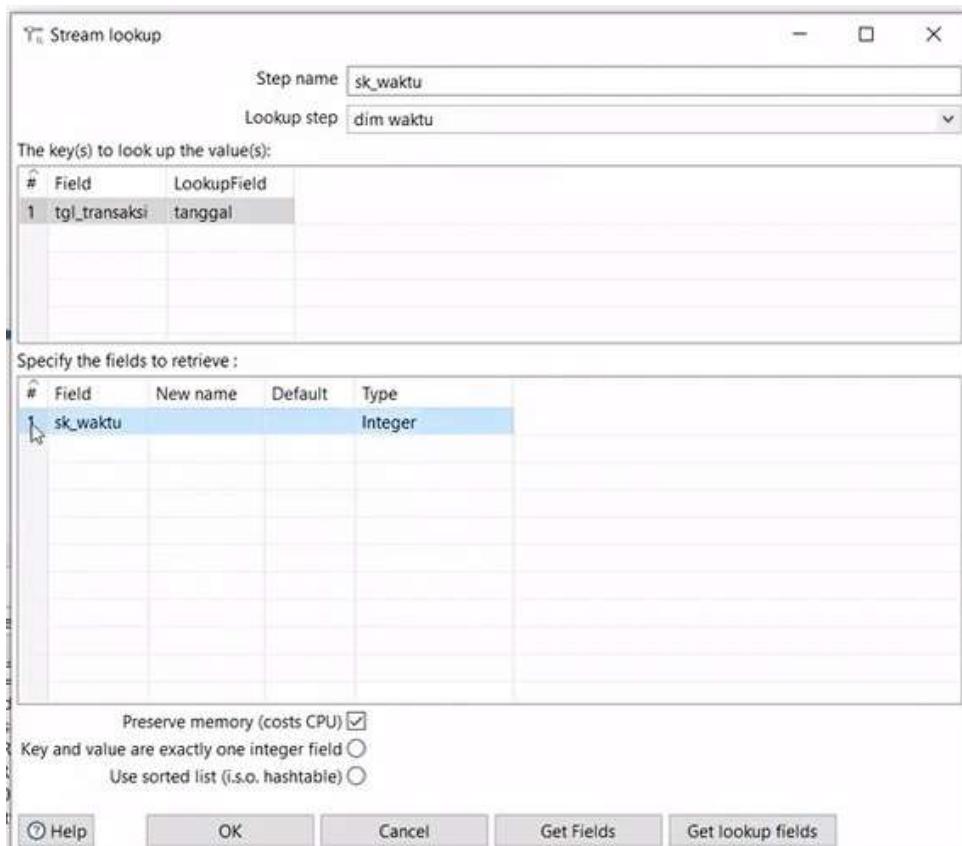
Close Stop Get more rows

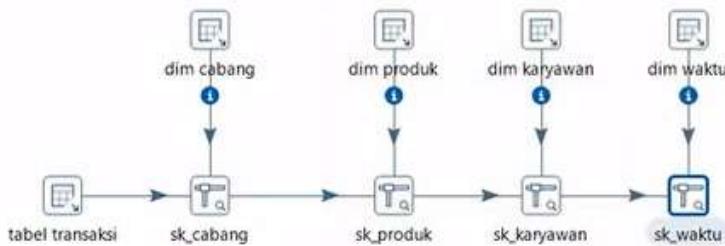
- Langkah yang sama juga berlaku untuk mengambil field **sk_karyawan** dari tabel dimensi karyawan dan field **sk_waktu** dari tabel dimensi waktu.



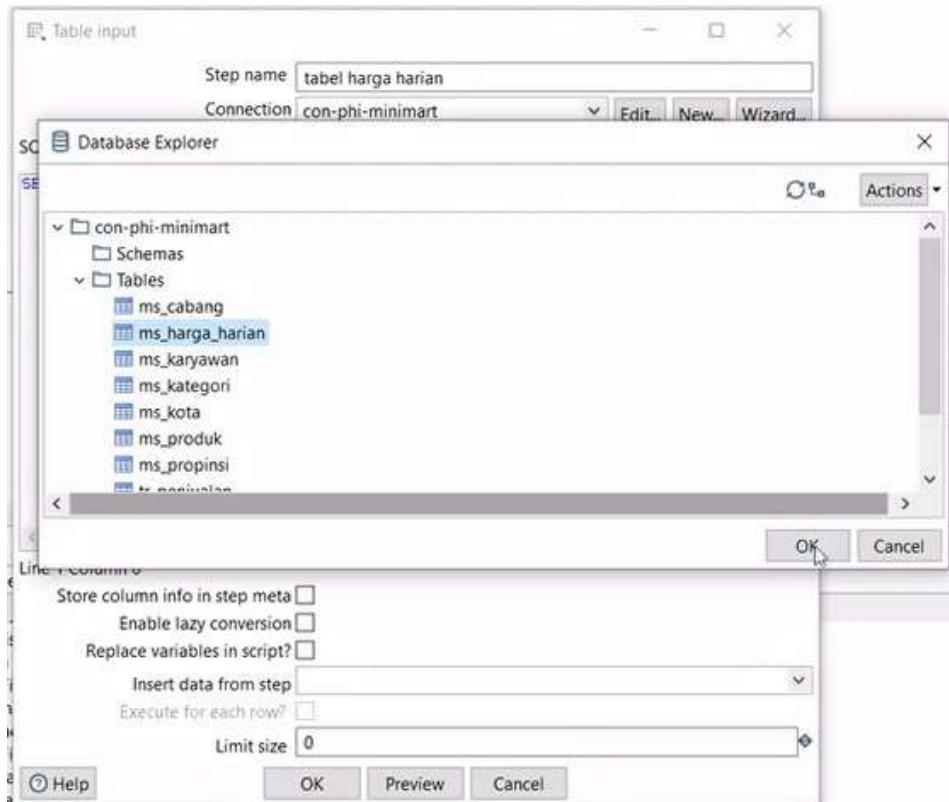




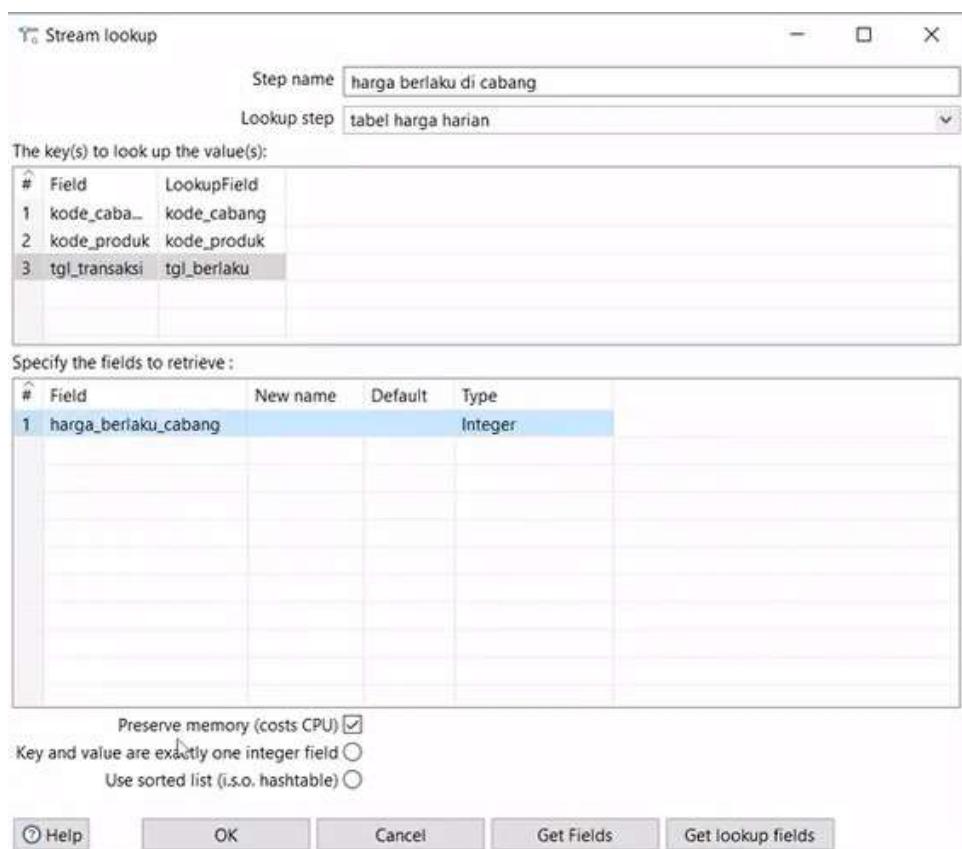




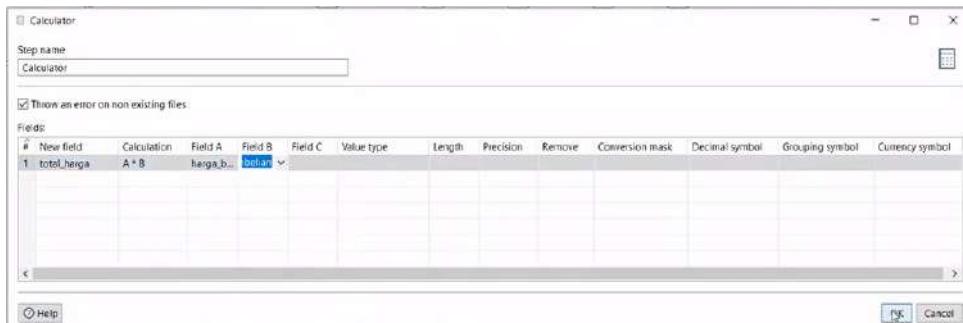
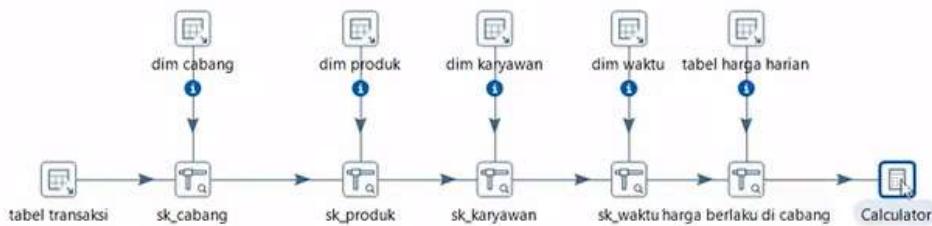
6. Berikutnya adalah menambahkan kolom **jumlah_unit** dan **total_jual**. Tambahkan step **Table input** ke dalam canvas dan Edit seperti berikut ini.



7. Selanjutnya step **Stream lookup** untuk menghubungkan **sk_waktu** dengan **tabel harga harian**.



8. Tarik step **Calculator** ke dalam canvas.



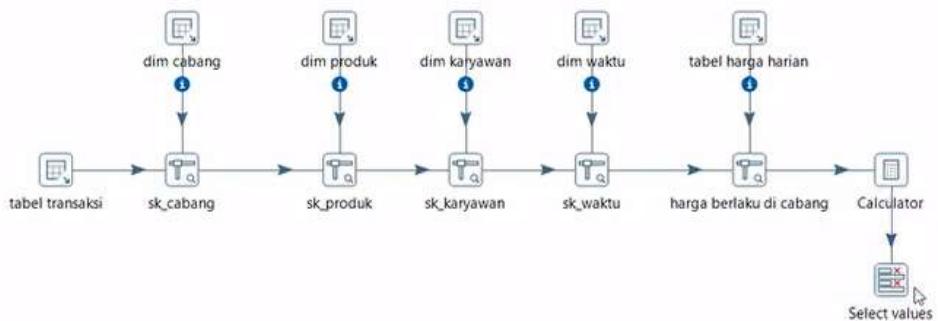
9. Periksa hasilnya dengan klik kanan Preview >> Quick Launch.

Examine preview data

Rows of step: Calculator (1000 rows)

#	tgt_transaksi	kode_cabang	kode_kasir	kode_item	kode_produk	jumlah_pemberian	sk_cabang	sk_produk	sk_karyawan	sk_waktu	harga_berlaku_cabang	total_harga
1	2008/01/01 00:00:00.000000000	CABANG-099	039-053	ITM-038	PROD-0000040	12	39	40	28	20080101	15090	181080.0
2	2008/01/01 00:00:00.000000000	CABANG-099	039-127	ITM-020	PROD-0000023	16	39	23	28	20080101	14170	226720.0
3	2008/01/01 00:00:00.000000000	CABANG-099	039-156	ITM-020	PROD-0000020	12	39	20	28	20080101	5240	62380.0
4	2008/01/01 00:00:00.000000000	CABANG-099	039-212	ITM-002	PROD-0000002	11	39	2	28	20080101	4220	46420.0
5	2008/01/01 00:00:00.000000000	CABANG-099	039-044	ITM-034	PROD-0000036	14	39	36	28	20080101	30360	420940.0
6	2008/01/01 00:00:00.000000000	CABANG-099	039-156	ITM-023	PROD-0000015	9	39	15	28	20080101	18880	169920.0
7	2008/01/01 00:00:00.000000000	CABANG-099	039-203	ITM-020	PROD-0000023	20	39	23	28	20080101	14170	283400.0
8	2008/01/01 00:00:00.000000000	CABANG-099	039-053	ITM-021	PROD-0000024	9	39	24	28	20080101	14940	134460.0
9	2008/01/01 00:00:00.000000000	CABANG-099	039-203	ITM-013	PROD-0000018	3	39	18	28	20080101	8960	26880.0
10	2008/01/01 00:00:00.000000000	CABANG-099	039-147	ITM-006	PROD-0000006	9	39	6	28	20080101	8110	72990.0
11	2008/01/01 00:00:00.000000000	CABANG-099	039-084	ITM-035	PROD-0000037	10	39	37	28	20080101	4990	40900.0
12	2008/01/01 00:00:00.000000000	CABANG-099	039-212	ITM-006	PROD-0000006	18	39	6	28	20080101	8110	145980.0
13	2008/01/01 00:00:00.000000000	CABANG-099	039-147	ITM-034	PROD-0000036	5	39	36	28	20080101	30360	150300.0
14	2008/01/01 00:00:00.000000000	CABANG-099	039-203	ITM-022	PROD-0000025	8	39	25	28	20080101	10160	81292.0
15	2008/01/01 00:00:00.000000000	CABANG-099	039-031	ITM-007	PROD-0000007	14	39	7	28	20080101	4640	64960.0
16	2008/01/01 00:00:00.000000000	CABANG-099	039-053	ITM-022	PROD-0000025	18	39	25	28	20080101	10160	162380.0
17	2008/01/01 00:00:00.000000000	CABANG-099	039-212	ITM-038	PROD-0000040	18	39	40	28	20080101	15090	271620.0
18	2008/01/01 00:00:00.000000000	CABANG-099	039-053	ITM-009	PROD-0000009	5	39	9	28	20080101	4680	23450.0
19	2008/01/01 00:00:00.000000000	CABANG-099	039-212	ITM-012	PROD-0000012	9	39	12	28	20080101	3440	30960.0
20	2008/01/01 00:00:00.000000000	CABANG-099	039-212	ITM-023	PROD-0000014	9	39	14	28	20080101	11520	109880.0
21	2008/01/01 00:00:00.000000000	CABANG-099	039-044	ITM-017	PROD-0000020	6	39	20	28	20080101	5240	31440.0
22	2008/01/01 00:00:00.000000000	CABANG-099	039-156	ITM-007	PROD-0000007	8	39	7	28	20080101	4640	37120.0
23	2008/01/01 00:00:00.000000000	CABANG-099	039-147	ITM-009	PROD-0000009	2	39	9	28	20080101	4680	9380.0
24	2008/01/01 00:00:00.000000000	CABANG-099	039-044	ITM-019	PROD-0000022	20	39	22	28	20080101	70280	1405600.0
25	2008/01/01 00:00:00.000000000	CABANG-099	039-156	ITM-035	PROD-0000037	6	39	37	28	20080101	4990	29940.0
26	2008/01/01 00:00:00.000000000	CABANG-099	039-151	ITM-015	PROD-0000018	3	39	18	28	20080101	8960	26880.0
27	2008/01/01 00:00:00.000000000	CABANG-099	039-127	ITM-013	PROD-0000018	18	39	18	28	20080101	8960	162380.0
28	2008/01/01 00:00:00.000000000	CABANG-099	039-031	ITM-033	PROD-0000035	7	39	35	28	20080101	6050	42350.0
29	2008/01/01 00:00:00.000000000	CABANG-099	039-127	ITM-002	PROD-0000002	18	39	2	28	20080101	4220	75960.0
30	2008/01/01 00:00:00.000000000	CABANG-099	039-053	ITM-036	PROD-0000038	6	39	38	28	20080101	5990	35940.0
31	2008/01/01 00:00:00.000000000	CABANG-099	039-127	ITM-012	PROD-0000012	18	39	12	28	20080101	3440	61920.0
32	2008/01/01 00:00:00.000000000	CABANG-099	039-127	ITM-037	PROD-0000039	17	39	35	28	20080101	5480	93160.0

10. Berikutnya sortir field dengan step **Select values**.

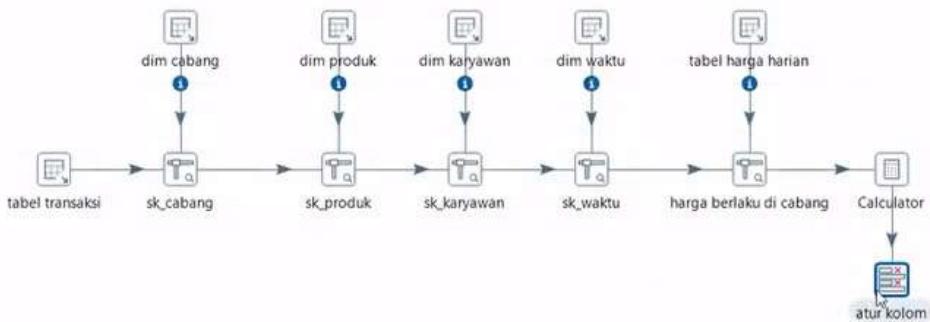


Select values

Select & Alter Remove Meta-data

Step name

#	Fieldname	Rename to	Length	Precision
1	sk_produk			
2	sk_karyawan			
3	kode_cabang			
4	sk_waktu			
5	jumlah_pembelian			
6	total_harga			



11. Periksa hasilnya dengan klik kanan Preview >> Quick Launch.

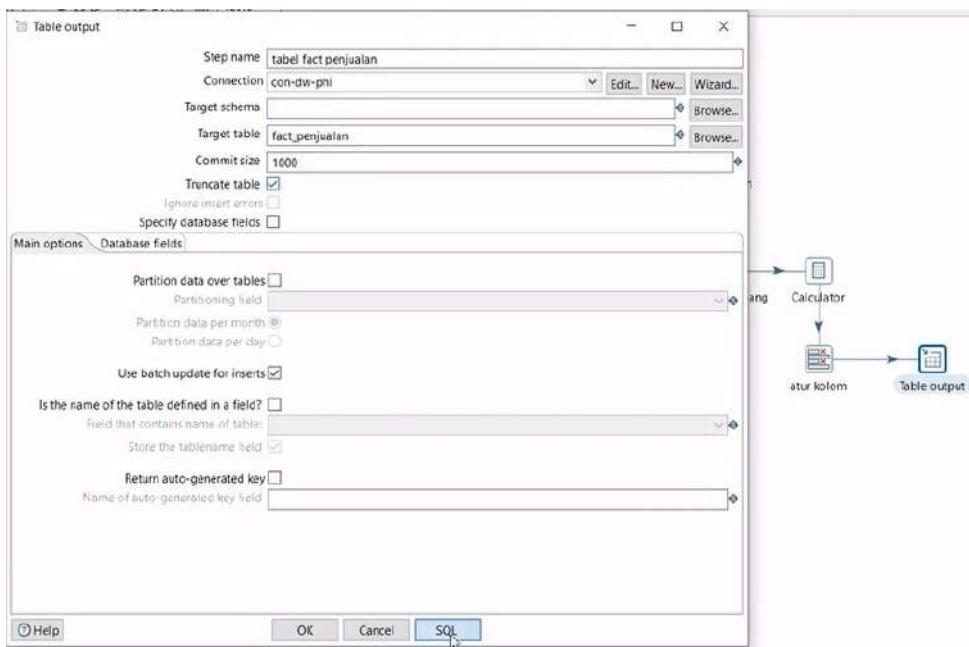
Examine preview data

Rows of step: atur kolom (1000 rows)

#	sk_produk	sk_karyawan	kode_cabang	sk_waktu	jumlah_pembelian	total_harga
1	40	28	CABANG-039	20080101	12	181080.0
2	23	28	CABANG-039	20080101	16	226720.0
3	20	28	CABANG-039	20080101	12	62880.0
4	2	28	CABANG-039	20080101	11	46420.0
5	36	28	CABANG-039	20080101	14	420840.0
6	15	28	CABANG-039	20080101	9	169920.0
7	23	28	CABANG-039	20080101	20	283400.0
8	24	28	CABANG-039	20080101	9	134460.0
9	18	28	CABANG-039	20080101	3	26880.0
10	6	28	CABANG-039	20080101	9	72990.0
11	37	28	CABANG-039	20080101	10	49900.0
12	6	28	CABANG-039	20080101	18	145980.0
13	36	28	CABANG-039	20080101	5	150300.0
14	25	28	CABANG-039	20080101	8	81280.0
15	7	28	CABANG-039	20080101	14	64960.0
16	25	28	CABANG-039	20080101	18	182880.0
17	40	28	CABANG-039	20080101	18	271620.0
18	9	28	CABANG-039	20080101	5	23450.0
19	12	28	CABANG-039	20080101	9	30960.0
20	14	28	CABANG-039	20080101	9	103680.0
21	20	28	CABANG-039	20080101	6	31440.0
22	7	28	CABANG-039	20080101	8	37120.0
23	9	28	CABANG-039	20080101	2	9380.0
24	22	28	CABANG-039	20080101	20	1405600.0
25	37	28	CABANG-039	20080101	6	29940.0
26	18	28	CABANG-039	20080101	3	26880.0
27	18	28	CABANG-039	20080101	18	161280.0
28	35	28	CABANG-039	20080101	7	42350.0
29	2	28	CABANG-039	20080101	18	75960.0
30	38	28	CABANG-039	20080101	6	35940.0
31	12	28	CABANG-039	20080101	18	61920.0
32	39	28	CABANG-039	20080101	17	93160.0

Close

12. Simpan data pada tabel fakta penjualan dengan step **Table output**.

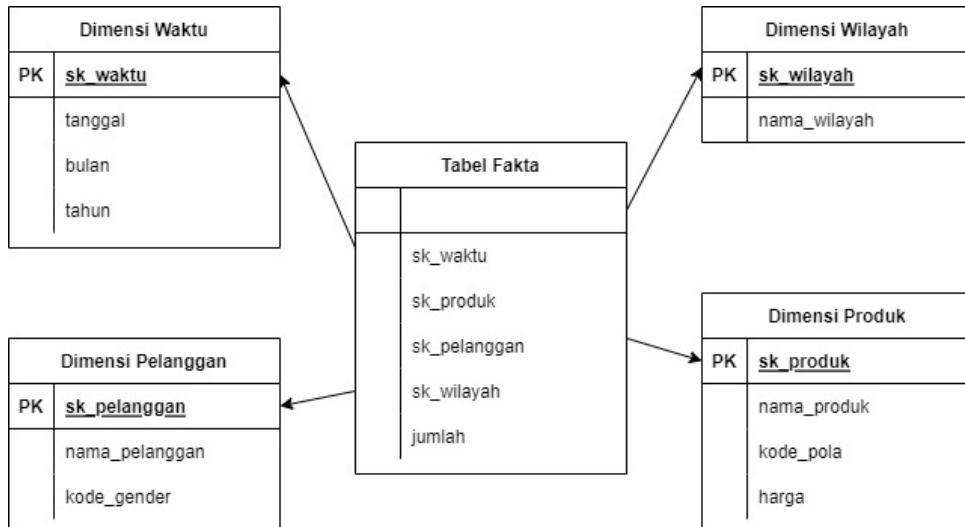


13. Jangan lupa simpan file dan tekan tombol **Run** untuk mengeksekusi.

4.5 Tugas

Buatlah tabel dimensi serta tabel fakta berdasarkan data excel serta star schema di bawah ini!

WAKTU	NAMA BARANG	HARGA	JUMLAH	PEMBELI	DAERAH
2010-03-26	Celana Standar Print Lasem	55000	17	Ibu Hadi Sukarni	Jawa Barat
2010-06-14	Bahan Beludru Cap Mahkota	500000	1	Ibu Tyas	Jawa Tengah
2010-11-21	Hem Sutra Print Rama	100000	5	Ibu Tyas	Jawa Tengah
2011-01-05	Kaos Katun Print Bola	60000	1	Bapak Imron	Jawa Barat
2011-03-27	Bahan Standar Cap Lasem	120000	8	Ibu Siti Arya	Jawa Barat
2011-04-09	Hem Katun Print Kawung	70000	3	Ibu Harini	Jawa Timur
2011-08-19	Hem Standar Tulis Madura	550000	5	Ibu Atik	Jawa Tengah
2011-10-13	Sarimbit Stadar Print Lukis	150000	1	Ibu Hatamah	Jawa Timur
2011-12-28	Jarik Standar Print Sogan	225000	2	Bapak Ketut	Bali
2011-12-30	Bolero Standar Cap Sidomukti	225000	1	Ibu Hatamah	Jawa Timur
2012-01-04	Kaos Batik Cap Lukis	30000	14	Ibu Harini	Jawa Timur
2012-01-09	Jam Standar Print Lukis	80000	44	Ibu Siti Arya	Jawa Barat
2012-02-14	Celana Standar Cap Warna	55000	17	Ibu Hadi Sukarni	Jawa Barat
2012-04-05	Bahan Standar Cap Garis	135000	7	Ibu Tyas	Jawa Tengah
2012-04-06	Jarik Standar Tulis Sarimbit	40000	4	Ibu Harini	Jawa Timur
2012-05-21	Hem Katun Print Kelenggan	299000	3	Bapak Totok	Jawa Timur
2012-06-22	Bahan Lawasan Tulis Tolet	130000	1	Ibu Niken	Jawa Tengah
2012-09-18	Batik Standar Cap Tumpal	150000	1	Bapak Heru	Jawa Timur
2012-09-28	Hem Standar Cap Tumpal	100000	1	Ibu Aini Kasmaji	Jawa Tengah
2012-12-15	Rok Batik Print Kombinasi	225000	1	Ibu Siti Arya	Jawa Barat



Modul 5

Pivot Table dan Chart

5.1 Tujuan

Mahasiswa mampu melakukan menampilkan *data warehouse* secara multidimensi dengan *pivot table* dan *chart*.

5.2 Landasan Teori

Sebuah *pivot table* sangat berguna untuk menyimpulkan, menganalisa, mengeksplorasi dan menyajikan data yang mudah dibaca dan dimengerti.

Pivot table adalah fitur dari Microsoft Excel yang sangat memudahkan dalam merangkum sejumlah besar data. Biasanya pivot table digunakan untuk menganalisa data numerik secara rinci, dari sini akan diperoleh jawaban dari pertanyaan-pertanyaan yang biasanya tidak terduga dari suatu data.

Pivot table berfungsi antara lain untuk:

1. Melakukan query sejumlah data yang sangat besar dengan cara yang mudah.
2. Proses kalkulasi subtotal dan menjumlahkan data numerik, meringkas data dengan sebuah kategori dan subkategori, serta membuat perhitungan dengan formula dan rumusan yang dapat dibuat sendiri.
3. Memperluas dan mempersempit tingkatan tampilan data yang berguna untuk fokus terhadap apa yang ingin dicari, dan menampilkan secara rinci dari ringkasan data (yang menjadi titik fokus perhatian).
4. Melihat data dari dimensi yang diinginkan.

5.3 Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi Ms. Office.
3. Modul Praktikum Data Warehousing dan Data Mining.

5.4 Langkah-langkah Praktikum

5.4.1 Membuat Pivot Table

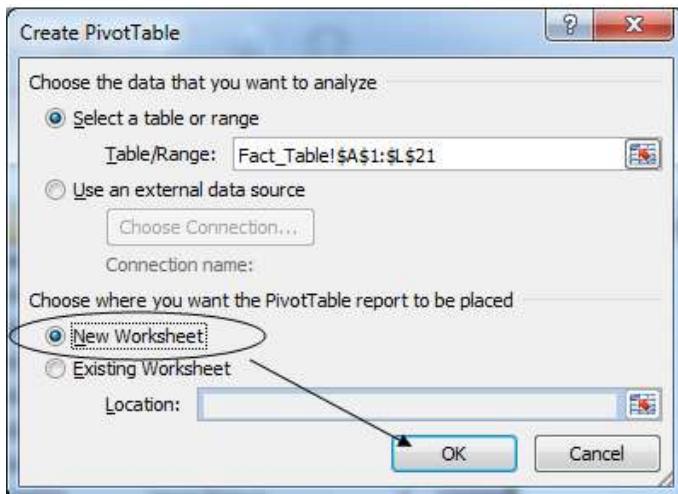
1. Gunakan file dengan nama “**Fakta Penjualan.xlsx**” yang merupakan hasil penggerjaan tugas pada modul sebelumnya. Anda bisa mengunduhnya pada URL berikut: https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM/src/branch/master/Data/ETL/Fakta%20Penjualan.xlsx. Buka sheet **Fact_Table**, dimana datanya terlihat seperti pada gambar berikut.

bulan	kuartal	tahun	nama_produk	nama_kategori	nama_subkategori	nama_pola	nama_pelanggan	jenis_kelamin	nama_wilayah	jumlah	harga
2	12	4	2011 Jarik Standar Print Standar	Jarik	Print	Bapak Ketut	PRIA	Bali		2	225000
3	1	1	2012 Kaos Batik Cap Lu Batik	Kaos	Cap	Ibu Harini	WANITA	Jawa Timur		14	30000
4	4	2	2012 Jarik Standar Tulis Standar	Jarik	Tulis	Ibu Harini	WANITA	Jawa Timur		4	40000
5	4	2	2011 Hem Katun Print IKatun	Hem	Print	Ibu Harini	WANITA	Jawa Timur		3	70000
6	9	3	2012 Batik Standar Cap Standar	Batik	Cap	Bapak Heru	PRIA	Jawa Timur		1	150000
7	5	2	2012 Hem Katun Print IKatun	Hem	Print	Bapak Totok	PRIA	Jawa Timur		3	299000
8	12	4	2011 Bolero Standar C/S Standar	Bolero	Cap	Ibu Hatamah	WANITA	Jawa Timur		1	225000
9	10	4	2011 Sarimbit Standar I Standar	Sarimbit	Print	Ibu Hatamah	WANITA	Jawa Timur		1	150000
10	1	1	2011 Kaos Katun Print IKatun	Kaos	Print	Bapak Imron	PRIA	Jawa Barat		1	60000
11	2	1	2012 Celana Standar C/S Standar	Celana	Cap	Ibu Hadi Sukarni	WANITA	Jawa Barat		17	55000
12	3	1	2010 Celana Standar Pr Standar	Celana	Print	Ibu Hadi Sukarni	WANITA	Jawa Barat		17	55000
13	3	1	2011 Bahan Standar Ca Standar	Bahan	Cap	Ibu Siti Arya	WANITA	Jawa Barat		8	120000
14	12	4	2012 Rok Batik Print K/C Batik	Rok	Print	Ibu Siti Arya	WANITA	Jawa Barat		1	225000
15	1	1	2012 Jam Standar Print Standar	Jam	Print	Ibu Siti Arya	WANITA	Jawa Barat		44	80000
16	9	3	2012 Hem Standar Cap Standar	Hem	Cap	Ibu Aini Kasmaji	WANITA	Jawa Tengah		1	100000
17	6	2	2012 Bahan Lawasan T/Lawasan	Bahan	Tulis	Ibu Niken	WANITA	Jawa Tengah		1	130000
18	8	3	2011 Hem Standar Tulis Standar	Hem	Tulis	Ibu Atik	WANITA	Jawa Tengah		5	550000
19	4	2	2012 Bahan Standar Ca Standar	Bahan	Cap	Ibu Tyas	WANITA	Jawa Tengah		7	135000
20	6	2	2010 Bahan Beludru Ca Beludru	Bahan	Cap	Ibu Tyas	WANITA	Jawa Tengah		1	500000
21	11	4	2010 Hem Sutra Print R/Sutra	Hem	Print	Ibu Tyas	WANITA	Jawa Tengah		5	300000

2. Pilih range data A1:L21 atau tekan tombol **CTRL + SHIFT + ***.
3. Klik tab **Insert** pada Ribbon, pilih menu **PivotTable** | **Insert PivotTable**.

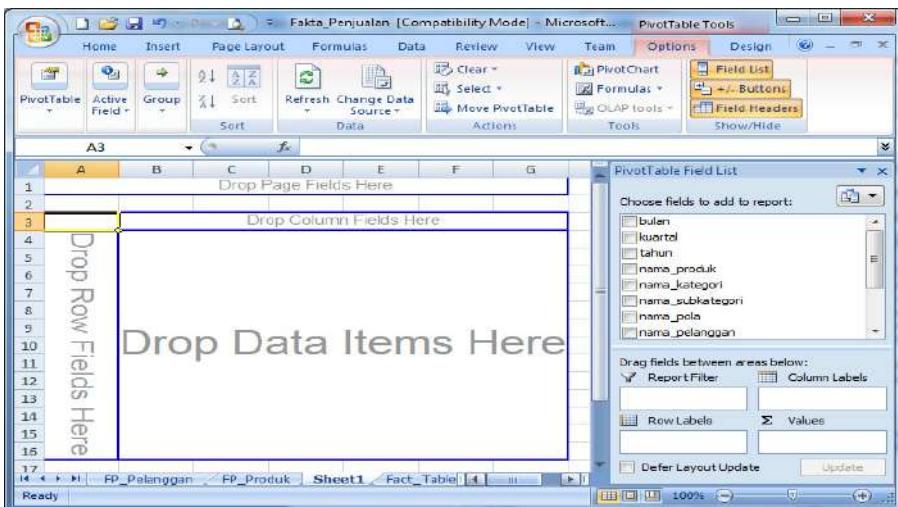


4. Pada dialog Create PivotTable yang muncul, pilih **New Worksheet**, klik tombol **OK**.



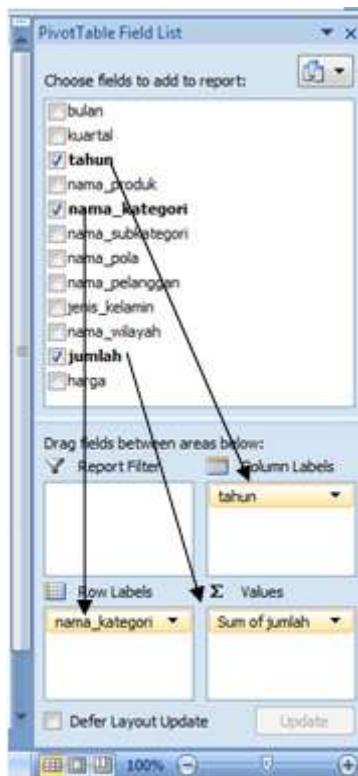
5. Sheet baru akan muncul disertai suatu kotak / placeholder PivotTable (*PivotTable Box*). Selain itu terdapat panel daftar field (**PivotTable**

Field List) pada posisi sebelah kanan worksheet. Terlihat pada daftar tersebut 10 field heading dari range data yang dipilih sebelumnya.



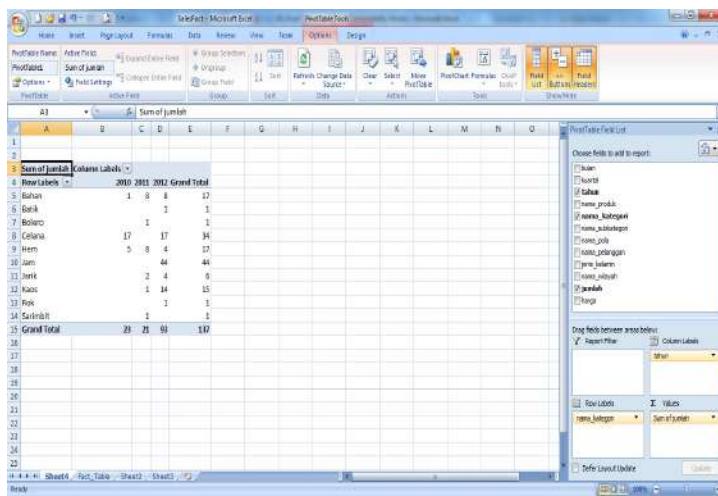
6. Pada bagian bawah panel sebelah kanan terdapat 4 kotak area. Tiap kotak tersebut dapat ditambahkan field-field yang terdapat pada field list. Adapun fungsi dari 4 kotak tersebut adalah sebagai berikut:
 - a. **Report Filter:** pada kotak ini field akan digunakan sebagai filter yang mempengaruhi hasil data pada PivotTable namun tidak akan terlihat sebagai isi dari PivotTable itu sendiri.
 - b. **Column Labels:** data dari field akan ditempatkan pada bagian kolom dari tabel dengan level sesuai urutan susunan pada area ini.
 - c. **Row Labels:** data dari field akan ditempatkan pada bagian baris dari tabel dengan level sesuai urutan susunan pada area ini.
 - d. **Values:** nilai field yang terdapat pada kotak ini akan dijadikan sebagai basis perhitungan *summary*. Tipe *summary* yang bisa digunakan adalah *count*, *sum*, *average*, *max*, *min* dan lain-lain.
7. Cobalah berbagai kombinasi penempatan field dalam kotak area tersebut. Susunlah layout field dengan urutan berikut :
 - a. Field **nama_kategori** ke kotak **Row Labels**.

- b. Field **tahun** ke kotak **Column Labels**.
- c. Field **jumlah** ke kotak **Values**.



Perhatikan pada saat ditempatkan di kotak **Values**, nama field **jumlah** akan berubah menjadi **Sum of jumlah**. Ini menandakan bahwa field tersebut merupakan kalkulasi **sum (penjumlahan)** dari nilai-nilai field **jumlah**.

- 8. Perhatikan hasil pengaturan ini pada area PivotTable. Area ini akan berisi suatu tabel dengan grouping field **nama_kategori** pada bagian baris, field **tahun** pada kolom. Sedangkan nilai total jumlah_unit ditempatkan pada cell-cell hasil perpotongan item grouping baris dan kolom tersebut.



Salah satu contoh perpotongan adalah total jumlah yang terjual dengan kategori **Jam** selama tahun **2012**, adalah sebesar **44** unit.

9. Simpan file dengan nama yang sama.

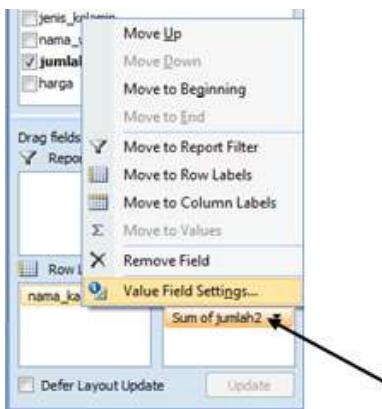
5.4.2 Menambahkan Tipe Summary Baru

1. Masih bekerja menggunakan file “**Fakta_Penjualan.xlsx**” pada kegiatan sebelumnya dengan Sheet1 PivotTable.
- 2.Tambahkan field **jumlah** kembali ke kotak **Value** dengan cara klik dan drag, sehingga muncul field baru dengan nama **Sum of jumlah2**.

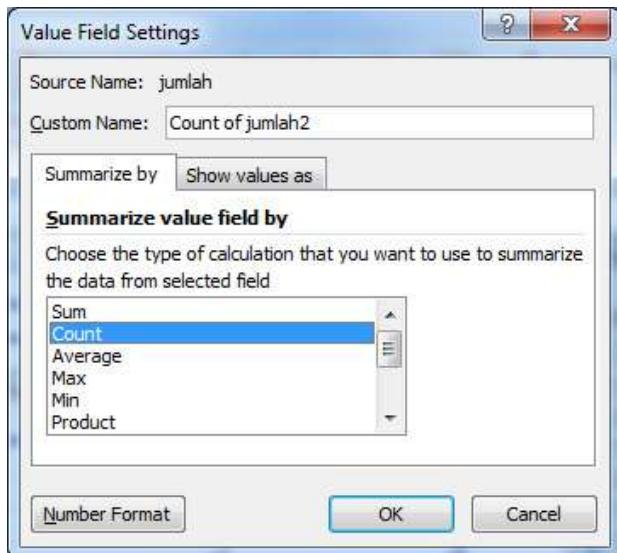


- Akan diperoleh tambahan satu kolom perhitungan baru yang sama dengan hasil sebelumnya pada masing-masing tahun. Namun tentunya bukan ini yang diinginkan.

- Kembali ke area **Values**, dan klik tombol panah ke bawah pada field **Sum of jumlah2**. Pilih item **Value Field Settings**.



5. Pada dialog Value Field Settings, ubah **Sum** menjadi **Count**. Perhatikan nama field akan berubah menjadi **Count of jumlah2**.



6. Klik tombol **OK**.
7. Pada area PivotTable, didapatkan dua *summary* yaitu:
 - a) nilai jumlah unit penjualan yang terjadi (**sum**).
 - b) jumlah transaksi yang terjadi (**count**).

The diagram shows a Pivot Table with data for various items across three years (2010, 2011, 2012). Arrows point from the 'Total jumlah' and 'jumlah2' values in the 'Grand Total' row to two boxes below it.

Total jumlah produk yang terjual

Jumlah transaksi yang terjadi (jumlah baris data penjualan)

	Column Labels								
	2010		2011		2012		Total Sum of jumlah		Total Count of jumlah2
Row Labels	Sum of jumlah	Count of jumlah2	Sum of jumlah	Count of jumlah2	Sum of jumlah	Count of jumlah2	Total Sum of jumlah	Total Count of jumlah2	
6 Bahan	1	1	8	1	8	2	17	4	
7 Batik					1	1	1	1	
8 Bolero			1	1			1	1	
9 Celana	17	1			17	1	34	2	
10 Hem	5	1	8	2	4	2	17	5	
11 Jam					44	1	44	1	
12 Jarik			2	1	4	1	6	2	
13 Kaos			1	1	14	1	15	2	
14 Rok					1	1	1	1	
15 Sarimbit			1	1			1	1	
16 Grand Total	23	3	21	7	93	10	137	20	

- Simpan kembali dengan nama file yang sama.

5.4.3 Calculated Field dan Calculated Item di Pivot Table

Pada **PivotTable** terdapat fasilitas yang bisa digunakan untuk menambahkan perhitungan dengan nama **Calculated Field** dan **Calculated Item** untuk membantu analisa lebih lanjut. Perbedaan dari kedua fasilitas ini yaitu:

- Calculated Field** digunakan jika ingin menambahkan field / kolom baru pada daftar field yang ada.
- Calculated Item** digunakan jika ingin menambahkan daftar nilai dari suatu field, dengan ini otomatis menambah item grouping baru. Sebagai catatan, formula tidak boleh menggunakan item dari field lain.

Berikut penggunaan Calculated Field dan Calculated Item.

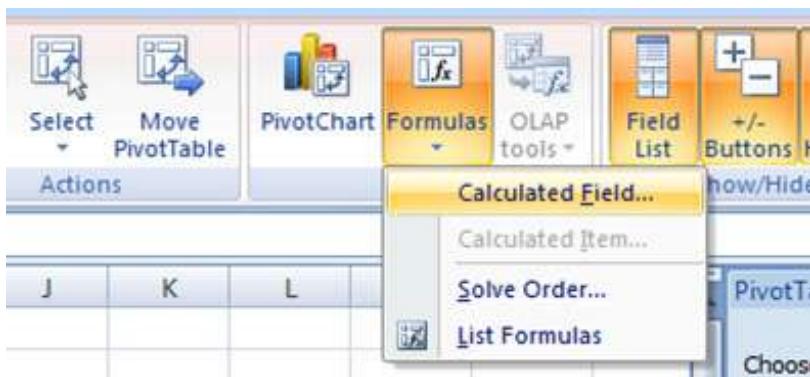
A. Calculated Field

Misalkan diinginkan untuk menambahkan sebuah field, yaitu jumlah pendapatan yang diperoleh berdasarkan jumlah produk yang terjual

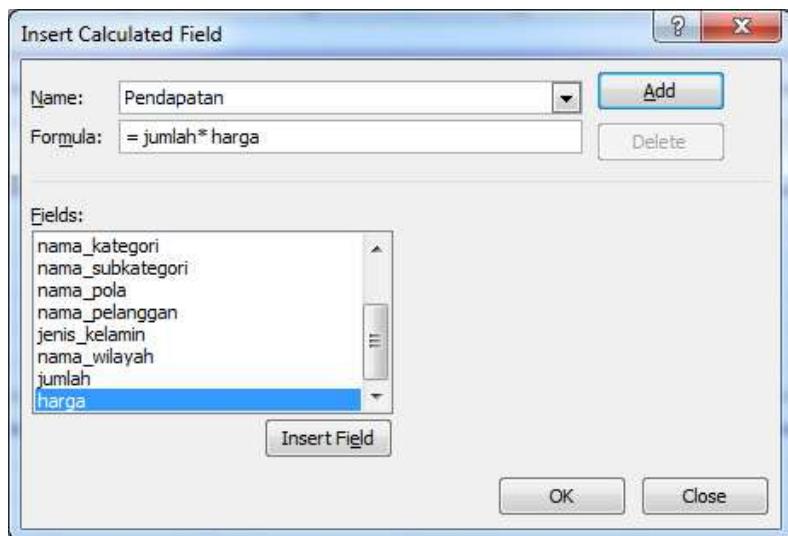
dikalikan dengan harga produk menggunakan Pivot Table yang terdapat pada file “**Fakta_Penjualan.xlsx**” pada Sheet **Fact_Table**.

Berikut adalah langkah-langkah untuk melakukan hal tersebut:

1. Buka Sheet1 dalam file **Fakta_Penjualan.xlsx**, dan letakkan kursor ke area PivotTable.
2. Pada menu ribbon **PivotTable Tools | Options**, klik button **Formulas** dan pilih **Calculated Field**.



3. Pada kotak dialog **Insert Calculated Field** yang muncul, masukkan nilai berikut kemudian klik tombol **OK**.
 - a) Name : Pendapatan
 - b) Formula : = jumlah * harga (Pilih field **jumlah** kemudian klik Insert Field kemudian ketikkan tanda "*" dan masukkan field **harga**)



4. Field baru, “**Sum of Pendapatan**” akan muncul pada Pivot Table.

				Total	Total	
				Sum of jumlah	Count of jumlah2	Total Sum of Pendapatan
Row Labels	Sum of jumlah	Count of jumlah2	Sum of Pendapatan			
6 Bahan	8	2	2.120.000	17	4	15.045.000
7 Batik	1	1	150.000	1	1	150.000
8 Bolero			-	1	1	225.000
9 Celana	17	1	935.000	34	2	3.740.000
10 Hem	4	2	1.596.000	17	5	19.023.000
11 Jam	44	1	3.520.000	44	1	3.520.000
12 Jarik	4	1	160.000	6	2	1.590.000
13 Kaos	14	1	420.000	15	2	1.350.000
14 Rok	1	1	225.000	1	1	225.000
15 Sarimbit			-	1	1	150.000
16 Grand Total	93	10	115.692.000	137	20	451.963.000

B. Calculated Item

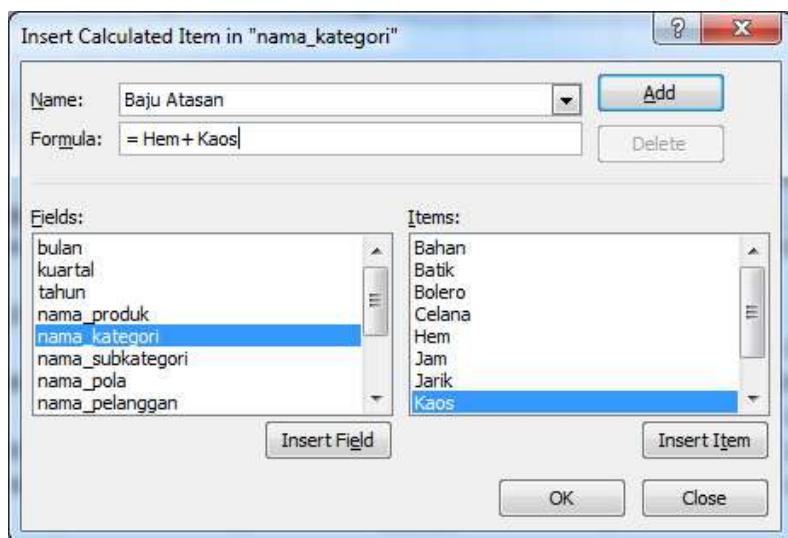
Misalkan diinginkan untuk menambahkan satu nilai pada field **nama_kategori**, yaitu **Baju Atasan** yang mewakili jumlah produk yang terjual untuk kategori **Hem** dan **Kaos**. Berikut adalah langkah-langkah untuk melakukan hal tersebut:

- Buka Sheet1 dan arahkan cursor ke area nilai nama_kategori pada Pivot Table. Sebagai contoh dengan memilih kategori **Bahan**.



			Total Sum of jumlah	Total Count of jumlah2	Total Sum of Pendapatan
Row Labels	2012		jumlah	jumlah2	Total Sum of Pendapatan
6 Bahan	8	2	2.120.000	17	15.045.000
7 Batik	1	1	150.000	1	150.000
8 Bolero			-	1	225.000
9 Celana	17	1	935.000	34	3.740.000
10 Hem	4	2	1.596.000	17	19.023.000
11 Jam	44	1	3.520.000	44	3.520.000
12 Jarik	4	1	160.000	6	1.590.000
13 Kaos	14	1	420.000	15	1.350.000
14 Rok	1	1	225.000	1	225.000
15 Sarimbit			-	1	150.000
16 Grand Total	93	10	115.692.000	137	451.963.000

- Pada ribbon **PivotTable Tools | Options**, klik button “Formulas” dan pilih “Calculated Item”.
- Pada kotak dialog **Insert Calculated Item in “nama_kategori”** yang muncul, masukkan nilai berikut di bawah ini kemudian klik tombol **OK**.
 - Name : Baju Atasan
 - Formula : = Hem + Kaos (Pilih item **Hem** kemudian klik **Insert Item**, ketikkan tanda “+” dan pilih **Kaos** kemudian klik **Insert Item** lagi)



4. Item baru pada nama_kategori yaitu **Baju Atasan** dan juga total penjumlahan unit dan Pendapatan akan muncul pada Pivot Table.

		2011		2012		Total Sum of jumlah	Total Sum of Pendapatan
Row Labels	Sum of jumlah	Sum of Pendapatan	Sum of jumlah	Sum of Pendapatan			
6 Bahan	8	960.000	8	2.120.000	17	15.045.000	
7 Batik		-	1	150.000	1	150.000	
8 Bolero	1	225.000		-	1	225.000	
9 Celana		-	17	935.000	34	3.740.000	
10 Hem	8	4.960.000	4	1.596.000	17	19.023.000	
11 Jam		-	44	3.520.000	44	3.520.000	
12 Jarik	2	450.000	4	160.000	6	1.590.000	
13 Kaos	1	60.000	14	420.000	15	1.350.000	
14 Rok		-	1	225.000	1	225.000	
15 Sarimbit	1	150.000		-	1	150.000	
16 Baju Atasan	9	6.120.000	18	7.722.000	32	38.688.000	
17 Grand Total	30	62.400.000	111	185.703.000	169	761.852.000	

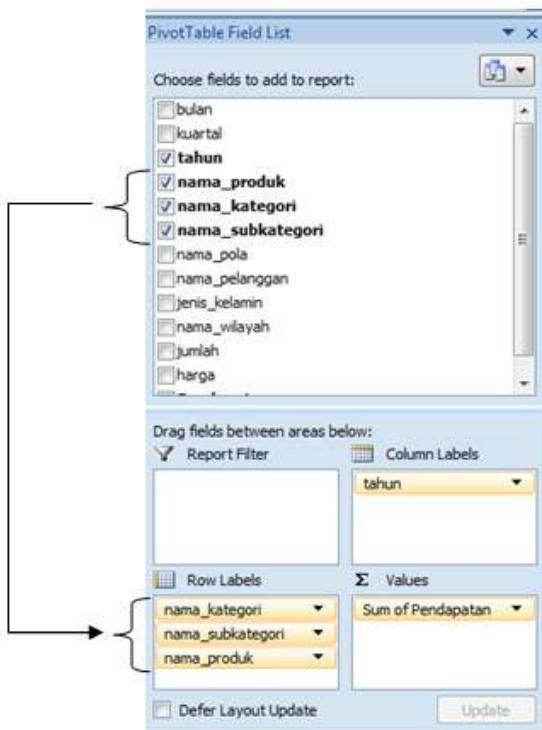
Nilai Baju Atasan diperoleh dari penjumlahan Hem dan Kaos

5.4.4 Operasi Roll Up dan Drill Down

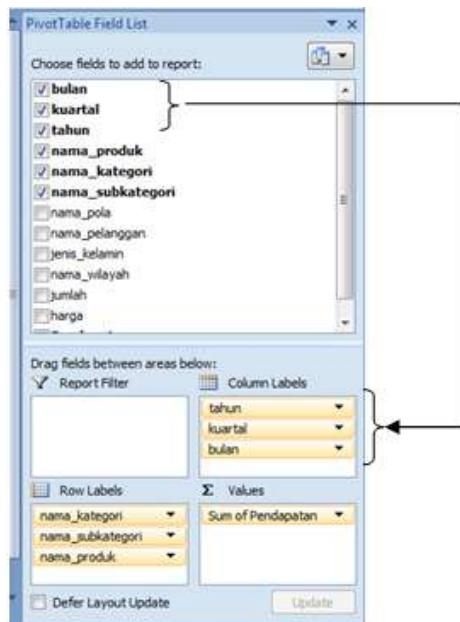
Operasi Roll Up dan Drill Down digunakan untuk melihat data secara lebih rinci dan secara lebih umum berdasarkan kategori tertentu pada sebuah data warehouse yang disajikan dalam bentuk *cube* (multidimensi). Secara khusus, operasi **Roll Up** berfungsi untuk melihat data secara lebih umum, sedangkan operasi **Drill Down** untuk melihat data secara lebih spesifik dan terperinci.

Berikut adalah langkah-langkah untuk melakukan operasi tersebut :

1. Buka Sheet1 (hasil pivot table) dan letakkan kursor pada area pivot table.
2. Pada kotak **PivotTable Field List**, hilangkan tanda cek pada field **Jumlah** (field ini sementara tidak digunakan), dan beri tanda cek pada field (kolom) yang akan ditampilkan ke dalam *cube*.
3. Beri tanda cek dan letakkan field-field berikut pada kotak **Row Labels** atau **Column Labels** sesuai dengan kebutuhan tampilan *cube*. Urutan field dalam kotak ini menentukan urutan rincian kategori data. Field yang terletak pada urutan teratas merupakan field dengan kategori paling umum, sedangkan field yang terletak pada urutan terbawah adalah field dengan kategori paling spesifik (paling rinci).
4. Misalkan pada Row Labels akan ditampilkan data berdasarkan urutan **nama_kategori**, **nama_subkategori**, dan **nama_produk**. Beri tanda cek pada field tersebut (bisa *drag and drop*) dan letakkan pada kotak **Row Labels**.



5. Pada Column Labels akan ditampilkan data berdasarkan urutan **tahun**, **kuartal**, dan **bulan**. Beri tanda cek pada field tersebut (*drag and drop*) dan letakkan pada kotak **Column Labels**.



6. Lihat kembali pada *cube* setelah ditambahkan field-field untuk operasi roll up dan drill down.
7. Pada masing-masing **Row Labels** dan **Column Labels** telah bertambah field-field yang bisa diperinci dan diringkas sesuai urutan kategori data yang lebih spesifik.

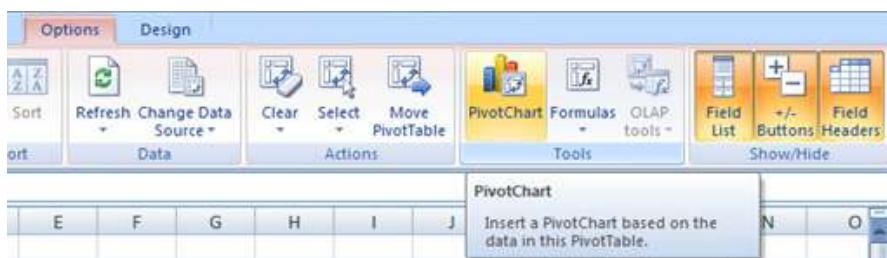
3 Sum of Pendapatan		Column Labels												2012 Total																	
		2012																													
		+1		+2		+3		+4		+5		+6		+7		+8		+9		+10		+11		+12		2 Total		+3		+4	
6 Row Labels																															
7 Bahan	kategori	0	0	0	945000	0	130000	0	0	0	0	2120000	0	0	2.120.000																
8 Lawasan	sub-kategori	0	0	0	0	0	130000	0	0	0	0	0	130000	0	0	130.000															
9 Bahan Lawasan Tulis Telepon		0	0	0	0	0	0	0	0	0	0	0	130000	0	0	130.000															
10 Standar		0	0	0	945000	0	0	0	0	0	0	945000	0	0	945.000																
11 Bahan Standar Cap Garis		0	0	0	945000	0	0	0	0	0	0	945000	0	0	945.000																
12 Batik		0	0	0	0	0	0	0	0	0	0	150000	0	0	150.000																
13 Celana	nama_produk	935000	0	0	0	0	0	0	0	0	0	0	0	0	0	935.000															
14 Hem		0	0	0	0	0	897000	0	0	0	0	0	897000	100000	0	1.596.000															
15 Jam		3520000	0	0	0	0	0	0	0	0	0	0	0	0	0	3.520.000															
16 Jarik		0	0	0	0	160000	0	0	0	0	0	0	160000	0	0	160.000															
17 Kaos		420000	0	0	0	0	0	0	0	0	0	0	0	0	0	420.000															
18 Rok		0	0	0	0	0	0	0	0	0	0	0	0	0	0	225000	225.000														
19 Baju Atasan		420000	0	0	0	0	897000	0	0	0	0	0	897000	100000	0	7.722.000															
20 Grand Total		17355000	0	0	1925000	3588000	130000	0	0	0	0	0	16254000	1050000	225000	185.703.000															

8. Klik tanda untuk melakukan operasi Roll Up dan klik tanda untuk melakukan operasi Drill Down.

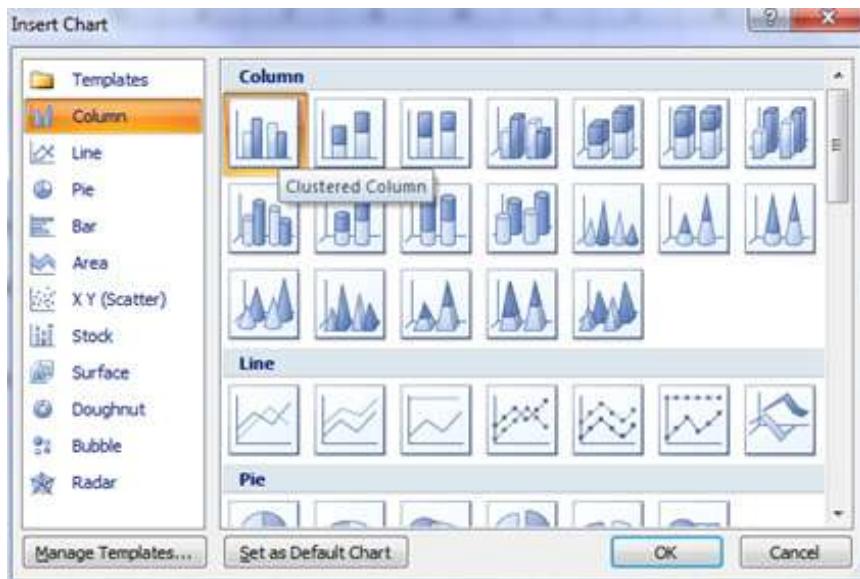
5.4.5 Menggunakan Pivot Chart

PivotChart merupakan sebuah cara untuk menampilkan cube dalam bentuk grafik. Dengan menggunakan grafik, sebuah pola atau statistik dari transaksi dalam waktu tertentu dapat dilihat dengan mudah dan dapat diketahui secara cepat. Selain itu, laporan-laporan dalam bentuk grafik sangat diperlukan untuk sebagai bahan dasar penentuan suatu kebijakan bagi para pengambil keputusan. Berikut adalah langkah-langkah untuk melakukan operasi tersebut:

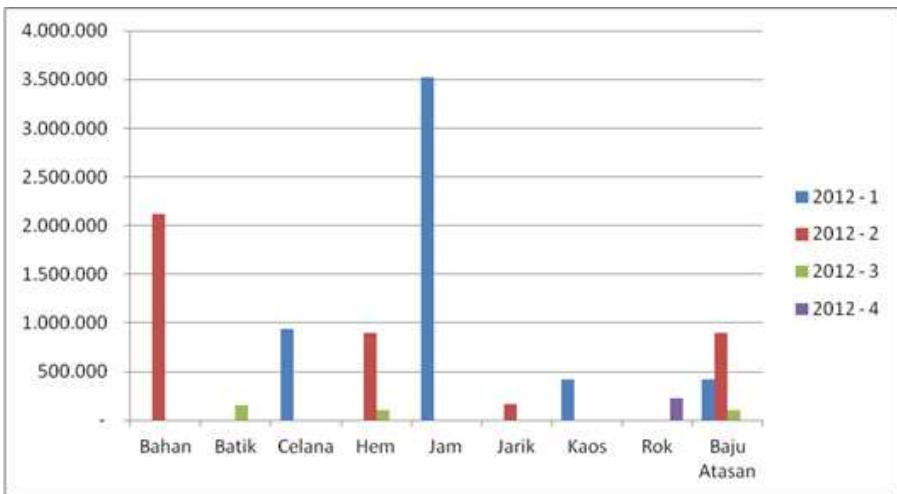
1. Arahkan kursor pada area pivot table dalam Sheet1 (Hasil PivotTable).
2. Pada menu Option, klik PivotChart.



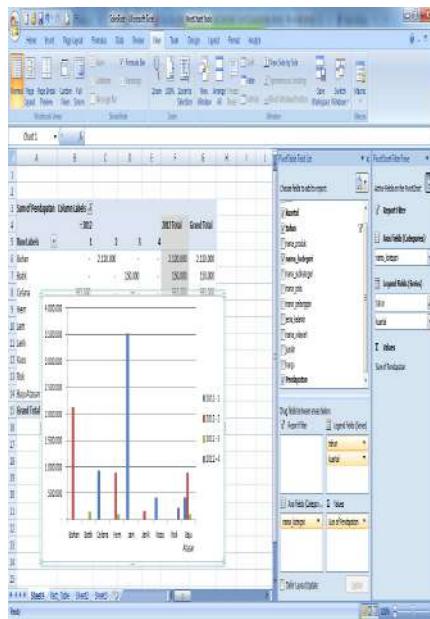
3. Pada jendela Insert Chart, pilih bentuk grafik yang diinginkan. Misalkan pilih grafik dalam bentuk batang, maka klik Clustered Column. Kemudian Klik OK.



4. Grafik akan ditampilkan dengan sumbu X dan sumbu Y menyesuaikan dengan Row Labels dan Column Labels.
5. Jika grafik terlalu rinci, maka bisa dibuat secara lebih umum dengan menghilangkan kembali tanda cek pada field dalam **PivotTable Field List**. Misalkan hilangkan tanda cek pada field **nama_produk**, **nama_subkategori**, dan **bulan**.



6. Dengan melihat grafik PivotChart, pola transaksi dari kuartal pertama hingga kuartal keempat dapat dilihat dengan mudah apakah terjadi kenaikan, penurunan atau stabil untuk masing-masing kategori produk.
7. Jendela PivotChart Filter Pane berfungsi untuk menyaring (*filter*) data-data khusus yang akan ditampilkan saja.



5.5 Tugas

1. Dengan menggunakan **PivotTable** pada file **Fakta_Penjualan.xls** tambahkan 2 buah field, yaitu :
 - a. **PPN** (Pajak Pertambahan Nilai) sebesar 10% dari tiap pendapatan pada Pivot Table.
 - b. **Total Penghasilan** yang dihitung dari pendapatan dikurangi dengan PPN tersebut.

	Column Labels					
	2012			Total Sum of Pendapatan	Total Sum of PPN (10%)	Total Sum of Total Penghasilan
Row Labels	Sum of Pendapatan	Sum of PPN (10%)	Sum of Total Penghasilan			
Bahan	2.120.000	212.000	1.908.000	2.120.000	212.000	1.908.000
Batik	150.000	15.000	135.000	150.000	15.000	135.000
Celana	935.000	93.500	841.500	935.000	93.500	841.500
Hem	1.596.000	159.600	1.436.400	1.596.000	159.600	1.436.400
Jam	3.520.000	352.000	3.168.000	3.520.000	352.000	3.168.000
Jarik	160.000	16.000	144.000	160.000	16.000	144.000
Kaos	420.000	42.000	378.000	420.000	42.000	378.000
Rok	225.000	22.500	202.500	225.000	22.500	202.500
Baju Atasan	7.722.000	772.200	6.949.800	7.722.000	772.200	6.949.800
Grand Total	185.703.000	18.570.300	167.132.700	185.703.000	18.570.300	167.132.700

2. Buatlah **PivotTable** dan **PivotChart** untuk melihat PPN dan Total Penghasilan tersebut selama tahun 2010 – 2012. Kategori produk apakah yang memberikan nilai penghasilan terbanyak selama 3 tahun tersebut?

Modul 6

Pengenalan Aplikasi Data Mining

6.1 Tujuan

1. Mahasiswa dapat memahami tujuan data mining.
2. Mahasiswa memahami dan mengenal beberapa software aplikasi yang dapat digunakan untuk data mining.

6.2 Landasan Teori

6.2.1 Pengertian Data Mining

Data mining adalah serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari suatu basisdata. Informasi yang dihasilkan diperoleh dengan cara mengekstraksi dan mengenali pola yang penting atau menarik dari data yang terdapat dalam basisdata.

Hal-hal yang melatarbelakangi data mining:

1. Melimpahnya data (*overload data*) yang dialami oleh berbagai institusi, perusahaan atau organisasi. Merlimpahnya data ini merupakan akumulasi data transaksi yang terekam bertahun-tahun.
2. Data-data tersebut merupakan data transaksi yang umumnya diproses menggunakan aplikasi komputer yang biasa disebut dengan OLTP (*On Line Transaction Processing*).
3. Adanya ledakan informasi (*explosion information*) dari berbagai media terutama internet yang secara umum informasi tersebut tidak terstruktur (*unstructured information*).

6.2.2 Manfaat Penggunaan Data Mining

1. Sudut Pandang Komersial

Data mining dapat digunakan dalam menangani meledaknya volume data. Bagaimana menyimpannya, mengestraknya serta memanfaatkannya. Berbagai teknik komputasi dapat digunakan untuk menghasilkan informasi yang dibutuhkan. Informasi yang dihasilkan menjadi aset untuk meningkatkan daya saing suatu institusi. Data mining tidak hanya digunakan untuk menangani persoalan menumpuknya data/informasi dan bagaimana menyimpannya tanpa kehilangan informasi yang penting (*warehousing*). Data mining juga diperlukan untuk menyelesaikan permasalahan atau menjawab kebutuhan bisnis itu sendiri, sebagai contoh:

- a. Bagaimana mengetahui hilangnya pelanggan karena pesaing
- b. Bagaimana mengetahui item produk atau konsumen yang memiliki kesamaan karakteristik
- c. Bagaimana mengidentifikasi produk-produk yang terjual bersamaan dengan produk lain.
- d. Bagaimana memprediski tingkat penjualan
- e. Bagaimana menilai tingkat resiko dalam menentukan jumlah produksi.
- f. Bagaimana memprediksi perilaku bisnis di masa yang akan datang.

2. Sudut Pandang Keilmuan

Data mining dapat digunakan untuk meng-*capture*, menganalisis serta menyimpan data yang bersifat *real-time* dan sangat besar, misalnya:

- a. Remote sensor yang ditempatkan pada suatu satelit
- b. Telescope yang digunakan untuk memindai langit
- c. Simulasi saintifik yang membangkitkan data dalam ukuran terabytes

Data mining merupakan salah satu metode alternatif yang dapat digunakan untuk mengolah data mentah, ketika metode konvensional tidak mungkin untuk dilakukan karena besarnya volume data yang diolah.

Hal ini dapat terjadi karena data mining memiliki kemampuan mereduksi data baik melalui teknik katalogisasi, klasifikasi maupun segementasi.

6.2.3 Proses Data Mining

Data mining sebenarnya merupakan salah satu rangkaian dari proses pencarian pengetahuan dalam database (*Knowledge Discovery in Database (KDD)*). KDD berhubungan dengan teknik integrasi dan penemuan ilmiah, interpretasi dan visualisasi dari pola-pola sejumlah kumpulan data. KDD adalah keseluruhan proses non-trivial untuk mencari dan mengidentifikasi pola (*pattern*) dalam data, dimana pola yang ditemukan bersifat sah, baru, dapat bermanfaat dan dapat dimengerti. Serangkaian proses tersebut memiliki tahap sebagai berikut:

1. Pembersihan data dan integrasi data (*cleaning and integration*). Proses ini digunakan untuk membuang data yang tidak konsisten dan bersifat *noise* dari data yang terdapat di berbagai basisdata yang mungkin berbeda format maupun platform yang kemudian diintegrasikan dalam satu database *data warehouse*.
2. Seleksi dan transformasi data (*selection and transformation*). Data yang terdapat dalam database *data warehouse* kemudian direduksi dengan berbagai teknik. Proses reduksi diperlukan untuk mendapatkan hasil yang lebih akurat dan mengurangi waktu komputasi terutama untuk masalah dengan skala besar (*large scale problem*). Beberapa cara seleksi, antara lain:
 - a. *Sampling*, yaitu seleksi subset representatif dari populasi data yang besar.
 - b. *Denoising*, yaitu proses menghilangkan noise dari data yang akan ditransformasikan.
 - c. *Feature extraction*, yaitu proses membuka spesifikasi data yang signifikan dalam konteks tertentu.

Transformasi data diperlukan sebagai tahap *pre-processing*, dimana data yang diolah siap untuk ditambang. Beberapa cara transformasi, antara lain:

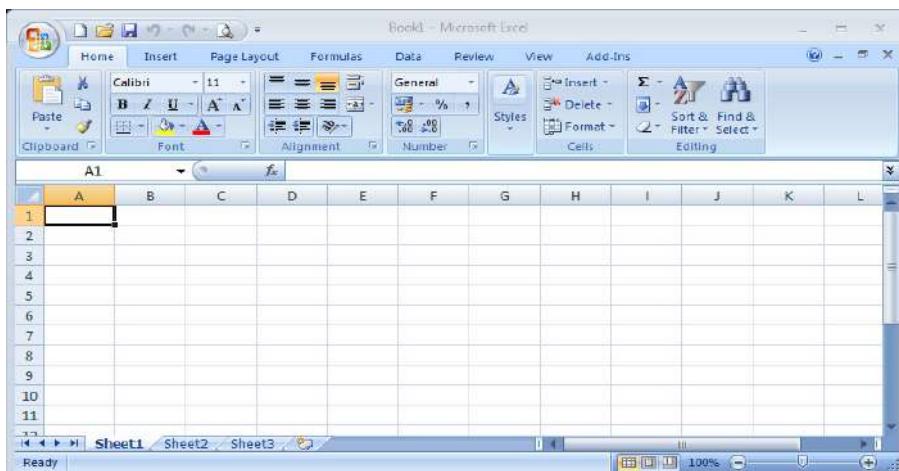
- a. *Centering*, mengurangi setiap data dengan rata-rata dari setiap atribut yang ada.
 - b. *Normalisation*, membagi setiap data yang di *centering* dengan standar deviasi dari atribut bersangkutan.
 - c. *Scaling*, mengubah data sehingga berada dalam skala tertentu.
3. Penambangan data (*data mining*)
Data-data yang telah diseleksi dan ditransformasi kemudian ditambah dengan berbagai teknik. Proses data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan fungsi-fungsi tertentu. Fungsi atau algoritma dalam data mining sangat bervariasi. Pemilihan fungsi atau algoritma yang tepat sangat bergantung pada tujuan dan proses pencarian pengetahuan secara keseluruhan.
4. Evaluasi pola dan presentasi pengetahuan
Tahap ini merupakan bagian dari proses pencarian pengetahuan yang mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya. Langkah terakhir KDD adalah mempresentasikan pengetahuan dalam bentuk yang mudah dipahami oleh pengguna.

6.3 Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Ms. Excel, Weka, RapidMiner, Python 3, Anaconda Navigator, Jupyter Notebook.
3. Dataset yang bisa diunduh dari folder /Data/Bab 06/ di repository:
https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM
4. Modul Praktikum Data Warehousing dan Data Mining.

6.4 Pengenalan Perangkat Lunak Data Mining

6.4.1 Perangkat Lunak 1: Microsoft Excel



Gambar 6.1 Lembar Kerja Microsoft Excel

Microsoft Excel atau Microsoft Office Excel adalah sebuah program aplikasi lembar kerja spreadsheet yang dibuat dan didistribusikan oleh Microsoft Corporation untuk sistem operasi Microsoft Windows dan Mac OS. Seperti yang ditunjukkan pada Gambar 6.1, aplikasi ini memiliki fitur kalkulasi dan pembuatan grafik. Excel menawarkan penghitungan kembali terhadap sel-sel secara cerdas, di mana hanya sel yang berkaitan dengan sel tersebut saja yang akan diperbarui nilainya (di mana program-program *spreadsheet* lainnya akan menghitung ulang keseluruhan data atau menunggu perintah khusus dari pengguna). Selain itu, Excel juga menawarkan fitur pengolahan grafik yang sangat baik.

6.4.2 Perangkat Lunak 2: Weka

WEKA adalah sebuah paket *tools machine learning* praktis. "WEKA" merupakan singkatan dari *Waikato Environment for Knowledge Analysis*, yang dibuat di Universitas Waikato, New Zealand untuk penelitian,

pendidikan dan berbagai aplikasi. WEKA mampu menyelesaikan masalah-masalah *data mining* di dunia-nyata, khususnya klasifikasi yang mendasari pendekatan-pendekatan *machine learning*. Perangkat lunak ini ditulis dalam hierarki *class Java* dengan metode berorientasi objek dan dapat berjalan hampir di semua *platform*.



Gambar 6.2 Halaman Depan Weka

Gambar 6.2 menunjukkan ada 4 tombol pada halaman depan aplikasi Weka yang dapat digunakan untuk menjalankan aplikasi, sebagai berikut:

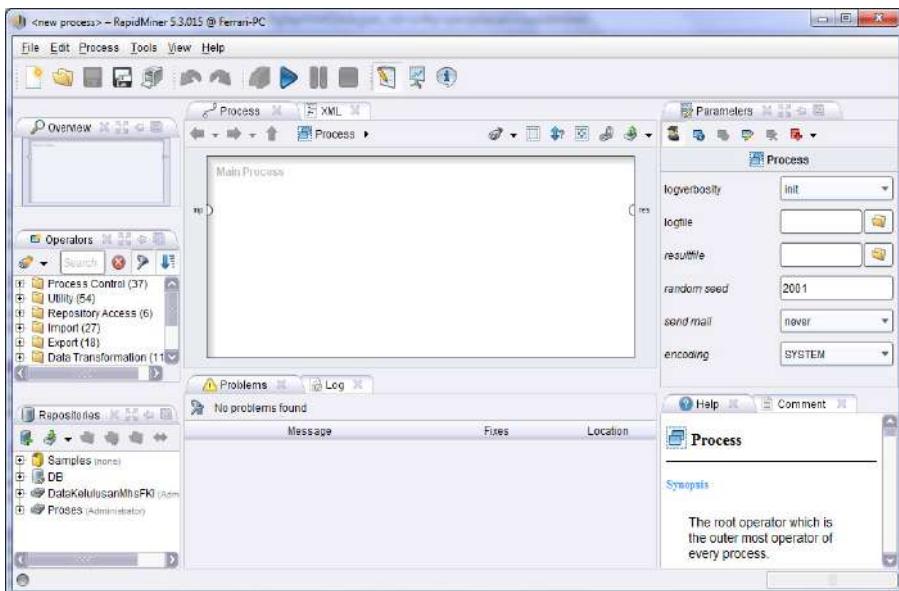
1. **Explorer** digunakan untuk menggali lebih jauh data dengan aplikasi WEKA.
2. **Experimenter** digunakan untuk melakukan percobaan dengan pengujian statistik skema belajar.
3. **Knowledge Flow** digunakan untuk pengetahuan pendukung.
4. **Simple CLI** antara muka dengan menggunakan tampilan command-line yang memungkinkan langsung mengeksekusi perintah weka untuk Sistem Operasi yang tidak menyediakan secara langsung.

6.4.3 Perangkat Lunak 3: RapidMiner

RapidMiner adalah sebuah platform perangkat lunak yang dikembangkan oleh perusahaan yang menyediakan lingkungan secara terintegrasi untuk *machine learning*, *data mining*, *text mining*, analisis

prediktif dan analisis bisnis. Aplikasi ini digunakan untuk kepentingan bisnis dan industri serta untuk penelitian, pendidikan, pelatihan, prototyping secara cepat, dan pengembangan aplikasi dan mendukung proses data mining termasuk hasil visualisasi, validasi dan optimasi. RapidMiner dikembangkan dari versi sebelumnya yang tersedia di bawah lisensi open source OSI-certified, seperti yang dapat dilihat pada Gambar 6.3.

RapidMiner menyediakan prosedur data mining dan *machine learning* termasuk untuk proses ETL (*extraction, transformation, loading*), *data preprocessing*, visualisasi, modeling dan evaluasi. Proses data mining tersusun atas operator-operator yang *nestable*, yang dideskripsikan dengan XML dan dibuat dengan GUI. Aplikasi ini ditulis dalam bahasa pemrograman Java dan mengintegrasikan proyek data mining Weka dan statistika R.

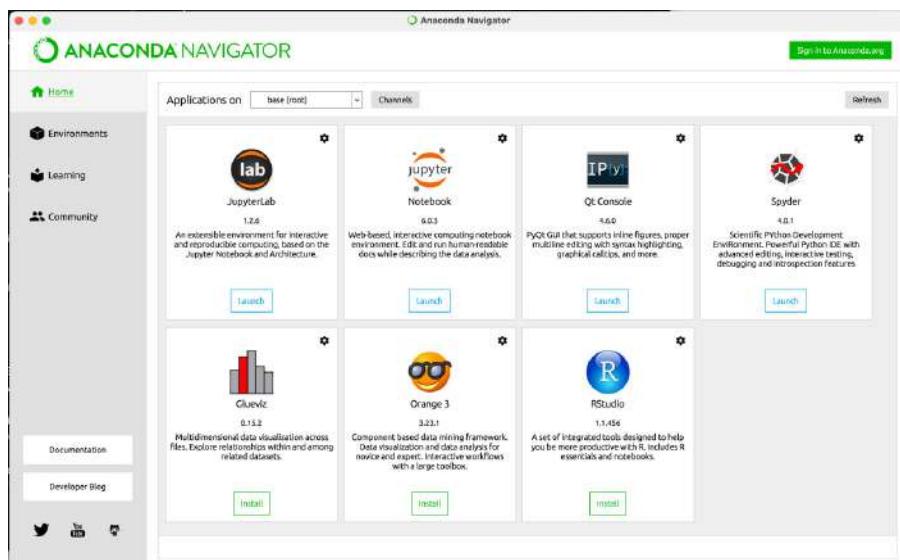


Gambar 6.3 Perspektif RapidMiner

6.5 Penggunaan Python untuk Data Mining

Python merupakan bahasa *scripting* yang berorientasi objek. Bahasa pemrograman ini dapat digunakan untuk pengembangan perangkat lunak dan bisa dijalankan melalui berbagai sistem operasi. Saat ini, Python juga merupakan bahasa yang populer di bidang *data science* dan analisis. Hal ini karena adanya dukungan bahasa Python terhadap *library-library* yang menyediakan fungsi analisis data dan fungsi *machine learning*, *data preprocessing tools*, serta visualisasi data.

Tugas data mining dengan Python dapat dilakukan dengan bantuan aplikasi Anaconda Navigator, seperti ditunjukkan pada Gambar 6.4. Anaconda Navigator telah menyediakan berbagai kelengkapan Python yang dikhususkan untuk kebutuhan analisis data. IDE (*Integrated Development Environment*) yang dapat digunakan antara lain *Jupyter Notebook* dengan extension .ipynb yang sudah merupakan bawaan dari Anaconda Navigator atau dengan text editor seperti *Sublime*, *Notepad*, *Notepad++* dengan extension .py.



Gambar 6.4 Tampilan Awal Aplikasi Anaconda Navigator

Berikut adalah contoh proses yang dilakukan oleh Python untuk proyek data mining sederhana:

Data mentah, dapat diimpor ke Python dengan menggunakan library. Library ini mempunyai fungsi untuk mengimpor data dengan format csv ke Python.

Eksplorasi data dan *data preprocessing* dapat dilakukan dengan lebih mudah, karena Python telah memiliki fungsi untuk melihat persebaran data dan melakukan manipulasi data untuk mengatasi data yang tidak sesuai. Dalam kasus ini *library* yang berfungsi adalah *sklearn* (Scikit learn) dan NumPy. Untuk visualisasi data, salah satu *library* yang terkenal adalah Matplotlib, dimana dapat membuat visualisasi dari persebaran data termasuk *plot* dan *chart*.

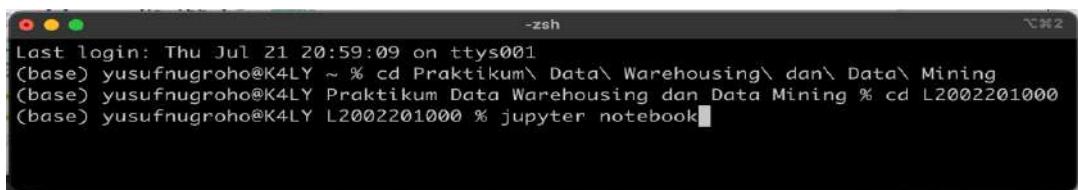
Pembuatan model data mining dapat dilakukan dengan cepat dengan tersedianya metode-metode dalam *machine learning* yang cukup lengkap dari Scikit learn. Contoh metode yang tersedia adalah *neural network*, *decision tree*, *SVM (Support Vector Machine)*, *random forest*, *regression tree*, *logistic regression*. Tidak hanya Scikit learn, ada beberapa *library* lain yang dapat digunakan seperti Keras, TensorFlow. Pembagian data testing dan training juga dapat dilakukan dengan *library* mengikuti metode yang tersedia. Contohnya adalah *cross-validation*.

Evaluasi model yang telah dibuat, seperti menghitung akurasi, sensitivitas, presisi, *error rate* dapat ditampilkan dengan mudah dengan menggunakan fungsi *classification_score* yang tersedia pada Scikit learn. Fungsi pada Scikit learn juga dapat menampilkan *confusion matrix* yang berisi nilai prediksi dan aktual yang dilakukan dari data *testing*.

Merepresentasikan hasil dari model dapat divisualisasikan dalam bentuk plot ataupun hasil dalam bentuk web. Ketika permintaan penampilan hasil dalam bentuk web, maka hasil dari model dapat disinkronkan karena Python juga mendukung untuk pembuatan web. Selain itu, untuk menampilkan model, dapat menggunakan library-library lain, contohnya adalah graphviz untuk menampilkan *decision tree*.

6.6 Langkah-langkah Praktikum

1. Buka Windows Explorer dan arahkan pada folder Praktikum Data Warehousing dan Data Mining. Buat sebuah folder sesuai dengan NIM mahasiswa di dalam folder Praktikum Data Warehousing dan Data Mining.
2. Gunakan data pelatihan dari dataset Titanic dengan nama *train.csv* yang telah diunduh dari repository [gitea](#).
3. Simpan file data pelatihan tersebut pada folder NIM yang telah dibuat pada langkah 1.
4. Buka aplikasi Anaconda Navigator untuk menjalankan Jupyter Notebook. Atau jika menggunakan “command prompt”, mula-mula arahkan drive pada folder yang telah dibuat sesuai dengan NIM. Ketikkan perintah `jupyter notebook` kemudian tekan Enter.

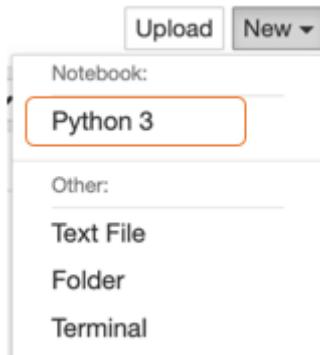


```
zsh
Last login: Thu Jul 21 20:59:09 on ttys001
(base) yusufnugroho@K4LY ~ % cd Praktikum\ Data\ Warehousing\ dan\ Data\ Mining
(base) yusufnugroho@K4LY Praktikum Data Warehousing dan Data Mining % cd L2002201000
(base) yusufnugroho@K4LY L2002201000 % jupyter notebook
```

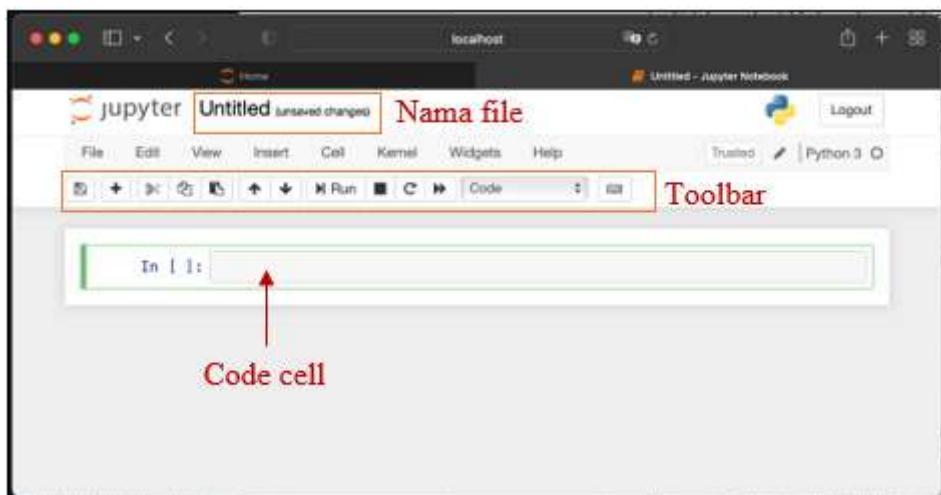
5. Aplikasi Jupyter Notebook akan dijalankan pada sebuah browser yang terinstal di komputer.



- Untuk membuat file kerja baru untuk menulis kode-kode program, klik tombol New -> Python 2 atau 3. Dalam contoh ini, kita menggunakan Python 3.

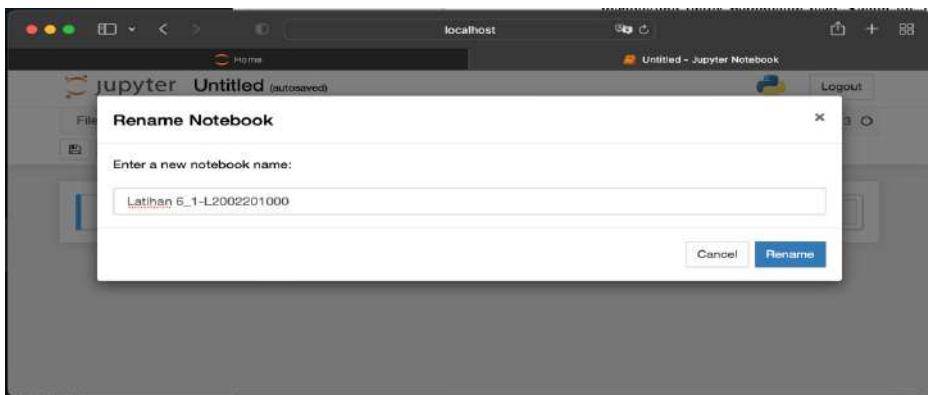


- Sebuah tab baru pada browser akan ditampilkan.



Tab baru ini menampilkan halaman *code editor* yang berfungsi sebagai tempat menulis kode-kode Python. Beberapa komponen penting dalam halaman ini antara lain:

- a. Nama file: digunakan untuk menentukan nama file yang akan memiliki ekstensi .ipynb.
 - b. Toolbar: berisi beragam tombol untuk melakukan tugas-tugas tertentu, misalnya *save*, *insert cell*, *cut cell*, *copy*, *paste*, *move cell up*, *move cell down*, *run cell*, *interrupt kernel*, *restart kernel*, *restart and rerun kernel*, and *style*.
 - c. Code cell: digunakan untuk menuliskan kode-kode program.
8. Dengan mengklik nama file “Untitled”, ubahlah nama file dengan format “Latihan 6_1-NIM”, misalnya “**Latihan_6_1-L2002201000**”. File ini akan tersimpan di komputer dalam folder *C:/Praktikum Data Warehousing dan Data Mining/Latihan 6_1-L2002201000.ipynb*. Setelah klik Rename, maka nama file akan berubah sesuai dengan yang diinputkan.



9. Pada percobaan pertama dalam Bab 6 ini, kita akan belajar bagaimana menampilkan data dari file CSV. Untuk mengeksekusi dengan Python, kita membutuhkan library pandas.
10. Pada sel pertama, ketikkan baris perintah berikut untuk mengimpor library *pandas* untuk mengolah dataframe. Klik tombol Run atau bisa dengan menekan tombol SHIFT+ENTER pada keyboard untuk mengeksekusi kode.

```
1 import pandas as pd
```

11. Tunggu proses *import library* selesai. Jika library tidak dikenali oleh Python, maka library tersebut perlu diinstal terlebih dahulu menggunakan perintah pip, selama terhubung dengan internet. Ketikkan baris kode berikut untuk menginstal library yang dibutuhkan.

```
In [ ]: !pip install <nama library>
```

Setelah proses instal *library* selesai, ulangi langkah 10 kemudian dilanjutkan ke langkah 12.

12. Ketik *method* `read_csv()` pada *pandas* untuk menampilkan file dataset Titanic dengan nama *train.csv*. Untuk menampilkan 5 data pertama, tambahkan *method* `head()` pada *dataframe*.

```
1 train_data = pd.read_csv("train.csv")
2 train_data.head()
```

Setelah dieksekusi, maka ditampilkan 5 data pertama dataset *train* Titanic yang terdiri dari 12 kolom.

Jika ingin melihat 10 data pertama, tambahkan angka 10 dalam tanda kurung, misalnya `train_data.head(10)`.

Atribut atau kolom tabel merupakan komponen penting dalam sebuah dataset. Untuk melihat nama-nama atribut yang digunakan dalam dataset, ketikkan perintah `train_data.columns`, kemudian eksekusi. Sehingga akan ditampilkan nama dari semua atribut dalam dataset.

```
train_data.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

Sedangkan untuk mengetahui informasi detil dari dataset terkait tipe data masing-masing atribut adalah dapat menggunakan perintah `train_data.info()`.

```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
 --- 
 0   PassengerId 891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object  
 4   Sex          891 non-null    object  
 5   Age          714 non-null    float64 
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object  
 9   Fare          891 non-null    float64 
 10  Cabin        204 non-null    object  
 11  Embarked     889 non-null    object  
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

13. Untuk mengetahui jumlah data yang terdapat pada dataset `train.csv`, ketikkan kode `print(len(train_data))`.

```
print("Jumlah data train:", len(train_data))
```

Jumlah data train: 891

Berdasarkan hasil eksekusi, dapat diketahui bahwa jumlah data yang terdapat pada dataset *train.csv* adalah sebanyak 891 baris.

14. Untuk melihat salah satu data pada baris tertentu dalam dataset, maka bisa menggunakan nomor *index* untuk menentukan barisnya, dimana nomor *index* diawali dari angka 0 (nol) dengan menggunakan *method iloc[]*. Misalnya untuk melihat detil dari data pada baris pertama, maka nomor *index* yang digunakan adalah 0. Ketikkan perintah berikut pada sel *train_data.iloc[0]*, kemudian eksekusi kode tersebut.

```
train_data.iloc[0]
```

```
PassengerId          1
Survived             0
Pclass                3
Name      Braund, Mr. Owen Harris
Sex                  male
Age                 22
SibSp                 1
Parch                 0
Ticket           A/5 21171
Fare                  7.25
Cabin                NaN
Embarked              S
Name: 0, dtype: object
```

15. Jika ingin melakukan filtering, yaitu hanya untuk menampilkan data berdasarkan nilai tertentu pada kolom tertentu, maka bisa menggunakan perintah *loc[]* yang ditambahkan pada *dataframe*. Contohnya jika hanya ingin menampilkan penumpang Titanic yang berjenis kelamin wanita (female), maka bisa menuliskan kode berikut:

```
1 | gender = train_data['Sex']=='female'
2 | train_data.loc[gender]
```

Sehingga jika dieksekusi akan menghasilkan 314 data berdasarkan jenis kelamin wanita (female).

```
1 gender = train_data['Sex']=='female'
2 train_data.loc[gender]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
...
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)	female	25.0	0	1	230433	26.0000	NaN	S
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552	10.5167	NaN	S
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382852	29.1250	NaN	Q
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W/C. 6607	23.4500	NaN	S

314 rows × 12 columns

16. Jika filtering berdasarkan nilai tertentu dari 2 kolom atau lebih, maka bisa menggunakan operator AND (&) dan/atau OR (|). Misalnya akan menampilkan data penumpang dengan jenis kelamin pria (male) dan yang menempati kelas penumpang 3.

```
1 | gender = train_data['Sex']=='male'
2 | pclass = train_data['Pclass']==3
3 | train_data.loc[gender & pclass]
```

17. Sedangkan jika ingin menyaring data berdasarkan string yang terkandung dalam suatu nilai kolom, maka bisa menggunakan method str.contains(). Misalnya data yang ditampilkan adalah penumpang wanita yang namanya mengandung kata "Miss". Jika hasil filtering mengabaikan huruf besar kecil, maka tambahkan case=False pada

perintah contains().

```
1 | name = train_data['Name'].str.contains("miss", case=False)
2 | train_data.loc[name]
```

```
1 | name = train_data['Name'].str.contains("miss", case=False)
2 | train_data.loc[name]
```

6.7 Tugas

Dikerjakan saat ini.

Dengan menggunakan dataset *test.csv* yang sudah diunduh dari repository, kerjakan perintah dan jawablah pertanyaan berikut ini.

1. Bukalah data *test.csv* dengan kode python. Ada berapa data yang terdapat dalam dataset tersebut?
2. Dengan menggunakan kode python, tampilkan nama-nama kolom dan tipe datanya pada dataset tersebut, serta tampilkan jumlah kolomnya!
3. Tampilkan dan tentukan jumlah data penumpang yang berjenis kelamin pria (male)!
4. Dalam satu tabel, tampilkan datanya dan tentukan jumlah data penumpang wanita (female) DAN namanya mengandung kata "James" ATAU penumpang pria (male) DAN namanya mengandung kata "Samuel"!

Modul 7

Data Preprocessing

7.1 Tujuan

1. Mahasiswa mampu menyebutkan tipe-tipe data yang digunakan dalam data mining.
2. Mahasiswa mampu menjelaskan permasalahan kualitas data dan penyelesaiannya.
3. Mahasiswa mampu melakukan data preprocessing.

7.2 Landasan Teori

Sebelum diproses data mining sering kali diperlukan preprocessing. Data preprocessing menerangkan tipe-tipe proses yang melaksanakan data mentah untuk mempersiapkan proses prosedur yang lainnya. Tujuan preprosesing dalam data mining adalah mentransformasi data ke suatu format yang prosesnya lebih mudah dan efektif untuk kebutuhan pemakai, dengan indikator sebagai berikut:

1. Mendapatkan hasil yang lebih akurat.
2. Pengurangan waktu komputasi untuk *large scale problem*.
3. Membuat nilai data menjadi lebih kecil tanpa merubah informasi yang dikandungnya.

Terdapat beberapa alat dan metode yang berbeda yang digunakan untuk preprocessing seperti:

1. *Sampling*, menyeleksi subset representatif dari populasi data yang besar.
2. *Transformation*, memanipulasi data mentah untuk menghasilkan input tunggal.
3. *Denoising*, menghilangkan noise dari data.
4. *Normalization*, mengorganisasi data untuk pengaksesan yang lebih spesifik.
5. *Feature extraction*, membuka spesifikasi data yang signifikan.

Menurut Susanto (2007), alasan-alasan harus dilakukan data preprocessing adalah antara lain:

1. Data mentah yang ada sebagian besar kotor, seperti:
 - a. Tidak lengkap, yaitu berisi data yang hilang / kosong, kekurangan atribut yang sesuai, atau hanya berisi data aggregate.
 - b. Banyak noise, yaitu berisi data yang *outlier*, atau berisi data error.
- c. Tidak konsisten, yaitu berisi nilai yang berbeda dalam suatu kode atau nama.
2. Data yang tidak berkualitas, akan menghasilkan kualitas mining yang tidak baik pula.
3. Data preprocessing, cleaning, dan transformasi merupakan pekerjaan mayoritas dalam aplikasi data mining.

Untuk mengatasi masalah data tersebut, dilakukan preprocessing terhadap data sebelum diolah dengan data mining. Preprocessing dapat dilakukan dengan beberapa teknik yaitu:

1. **Cleaning**, memperkecil jumlah data yang hilang atau berbeda, dapat dilakukan dengan cara:
 - a. Mengisi data yang hilang dengan *default value*.
 - b. Mengisi data secara manual, misal: trace ulang transaksi untuk mengetahui data yang hilang.
 - c. Mengisi dengan rata-rata atribut tersebut, misal: gaji pegawai

- yang kosong diisi dengan rata-rata gaji pegawai.
- d. Mengisi dengan rata-rata suatu atribut untuk kelas yang sama, misal: gaji pegawai yang kosong diisi dengan rata-rata gaji pegawai yang memiliki jabatan yang sama.
 - e. Menggunakan regresi, prediksi berdasarkan dua variabel yang lain, misal: mengisi gaji pegawai yang kosong dengan nilai prediksi dengan regresi berdasarkan jabatan dan lama masa kerja.
 - f. Menghilangkan baris yang mengandung data yang hilang. misal: menghilangkan data pegawai yang gaji pegawainya kosong.
 - g. *Binning by means*, menggunakan rata-rata pengelompokan. misal: sorted data dibagi menjadi beberapa kelompok, dan dicari rata-rata masing-masing kelompok untuk mengganti setiap data yang ada, sesuai dengan kelompoknya. Misal data dari kelompok A diganti dengan rata-rata kelompok A.
 - h. *Binning by range boundaries*, menggunakan batas terdekat suatu kelompok data, misal: sorted data dibagi menjadi beberapa kelompok, dicari nilai minimum dan maximum dari masing-masing kelompok, lalu gantikan tiap nilai di suatu kelompok dengan batas atas atau batas bawah kelompoknya, sesuai dengan yang paling dekat.
 - i. Mencari dan menghilangkan outlier dengan pengelompokan atau regresi
 - j. *Binning*, mengganti suatu nilai outlier dengan nilai yang lebih sesuai dengan data lain yang ada di sekitar data outlier tersebut (*local smoothing*).
2. **Integrasi**, menggabungkan beberapa sumber data sehingga dapat saling melengkapi. Data perlu digabungkan dengan key yang sesuai. Key ini mungkin memiliki nama yang berbeda di sumber data yang berbeda. Misal di tabel a menggunakan nama atribut 'id', di tabel b menggunakan nama atribut 'nomor', atau satuan yang digunakan untuk konsep yang sama (misal harga) disimpan dalam juta dan ribu.

3. **Transformasi**, mengubah data yang kompleks dengan tidak menghilangkan isi, sehingga lebih mudah diolah, dilakukan dengan cara:
 - a. *Smoothing (binning, clustering dan regresi)*.
 - b. Agregasi (*summarize*, menggunakan dimensi yang lebih general (*cube construction*)).
 - c. Generalisasi, misal menggunakan dimensi propinsi daripada kabupaten atau *grouping* (hirarki konsep).
 - d. Normalisasi, mengelompokkan data sesuai skala tertentu, misal IPK.
 - 1) Normalisasi min-max, standarisasi data dengan menempatkan data dalam range 0 sampai 1, nilai terkecil sebagai 0, dan nilai terbesar sebagai 1. $\text{nilai baru} = ((\text{nilai lama} - \text{nilai minimal}) / (\text{nilai maksimal} - \text{nilai minimal})) (\text{range maksimal} - \text{range minimal}) + \text{range minimal}$. range minimal = 0, range maksimal = 1.
 - 2) Normalisasi z-index, nilai baru = $(\text{nilai lama} - \text{rata-rata}) / \text{standar deviasi}$.
 - 3) Normalisasi skala desimal, nilai baru = $\text{nilai lama} / 10^{\wedge} x$.
4. **Diskretisasi**, membagi nilai data menjadi beberapa range data, dilakukan dengan cara:
 - a. Binning, seperti prosedur **Cleaning** 1.10.
 - b. Hirarki konsep, misal mengelompokkan harga produk menjadi, mahal, biasa, murah.
5. **Reduksi**, mengurangi jumlah data sehingga resource yang digunakan lebih sedikit, sehingga prosesnya dapat lebih cepat dilakukan dengan cara:
 - a. Sampling/generalisasi,
 - b. Agregasi, seperti agregasi pada transformasi. data ribuan memiliki volume byte yang lebih kecil daripada data jutaan
 - c. Mengurangi atribut yang tidak perlu (korelasi yang rendah terhadap keseluruhan data), misal nomor telepon, nama ibu atau nama jalan. Jika data set memiliki atribut sejumlah n,

maka ada 2^n (2 pangkat n) kemungkinan korelasi antar atribut kompresi data.

7.3 Alat dan Bahan

1. Python 3, Anaconda Navigator, Jupyter Notebook.
2. Dataset yang bisa diunduh dari folder /Data/Bab 07/ di repository: https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM
3. Modul Praktikum Data Warehousing dan Data Mining.

7.4 Langkah-langkah Praktikum

Pada praktikum Modul 7 ini, kita akan belajar bagaimana melakukan data preprocessing dan memvisualisasikan data.

1. Buka Windows Explorer dan arahkan pada folder Praktikum Data Warehousing dan Data Mining. Buat sebuah folder sesuai dengan NIM mahasiswa di dalam folder Praktikum Data Warehousing dan Data Mining.
2. Unduh semua data dari dataset Titanic dari repository gitea Bab 07 dan disimpan pada folder NIM yang telah dibuat pada langkah 1.
3. Buka aplikasi Anaconda Navigator untuk menjalankan Jupyter Notebook. Atau jika menggunakan “command prompt”. Ketikkan perintah jupyter notebook kemudian tekan Enter.
4. Aplikasi Jupyter Notebook akan dijalankan pada sebuah browser yang terinstal di komputer.
5. Buat file kerja baru untuk menulis kode-kode program, klik tombol New -> Python 3.
6. Dengan mengklik nama file “Untitled”, ubahlah nama file dengan format “Latihan 7_1-NIM”, misalnya “**Latihan_7_1-L2002201000**”. File ini akan tersimpan di komputer dalam folder *C:/Praktikum Data Warehousing dan Data Mining/NIM/Latihan 7_1-L2002201000.ipynb*.
7. Pada sel pertama, kita melakukan *import* beberapa *library* yang dibutuhkan dalam data preprocessing, antara lain *numpy*, *pandas*,

matplotlib, *seaborn*, dan *sklearn*. Eksekusi kode pada sel pertama dengan menekan tombol **Run** atau shift+Enter pada keyboard.

```
1 # Import libraries
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
```

8. Selanjutnya, kita akan mengakses tiga dataset yang telah kita unduh sesuai pada Langkah 2 dengan menggunakan *library pandas* untuk disimpan dalam bentuk *dataframe*.

```
1 #Load data
2 train_data = pd.read_csv('train.csv')
3 test_data = pd.read_csv('test.csv')
4 gender_submission = pd.read_csv('gender_submission.csv')
```

9. Untuk menampilkan *dataframe* dari masing-masing dataset, gunakan *method head()*, misalnya *train_data.head()* yang secara default akan menampilkan 5 data pertama.
10. Informasi deskripsi dari masing-masing *dataframe*, seperti nilai rata-rata (*mean*), *standard deviation (std)*, nilai minimum (*min*), nilai maksimum (*max*), dan lain-lainnya dapat dilihat dengan menambahkan *method describe()* pada *dataframe*.

```
1 train_data.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Data preprocessing merupakan salah satu tahap awal dalam data mining untuk melakukan pembersihan data dari berbagai kesalahan misalnya data yang kosong, menghapus atribut yang tidak diperlukan dalam data mining, maupun mengubah data yang ada menjadi data yang berbeda misalnya menggabungkan nilai dari 2 kolom menjadi 1, mengubah data teks menjadi data angka atau sebaliknya.

Number of Missing Values

11. Pada tahap ini, kita akan melihat jumlah data dari semua atribut/kolom yang tidak memiliki nilai. Ketikkan perintah berikut ini pada sel jupyter kemudian dieksekusi.

```
1 column_names = train_data.columns
2 for column in column_names:
3     print(column + ' - ' + str(train_data[column].isnull().sum()))
```

```
PassengerId - 0
Survived - 0
Pclass - 0
Name - 0
Sex - 0
Age - 177
SibSp - 0
Parch - 0
Ticket - 0
Fare - 0
Cabin - 687
Embarked - 2
```

Hasil ini menunjukkan bahwa ada beberapa atribut yang masih memiliki data yang kosong, misalnya kolom *Age* terdapat 177 data kosong, *Cabin* memiliki 687 data kosong, dan *Embarked* memiliki 2 data kosong.

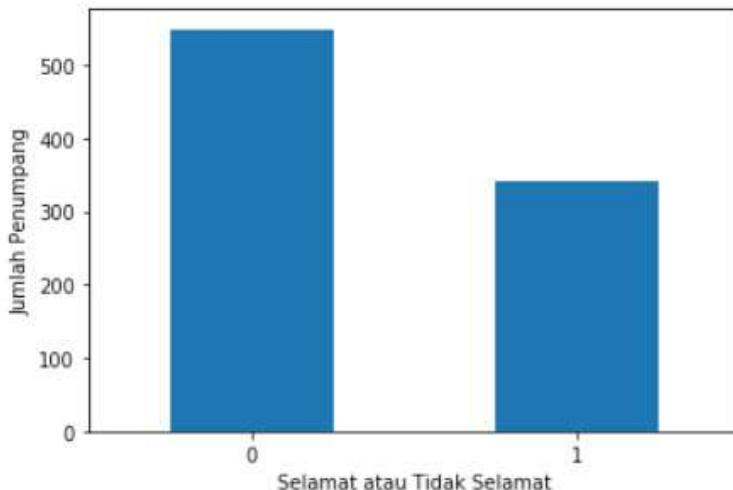
12. Pada dataset *Titanic*, atribut **Survived** adalah atribut yang menjadi target dalam analisis, misalnya untuk keperluan prediksi keselamatan (*survival*) para penumpang. Untuk melihat data jumlah penumpang yang selamat dan tidak selamat dalam data pelatihan (*train.csv*), kita bisa menggunakan *method* *value_counts()* terhadap kolom dalam *dataframe*.

```
1 train_data['Survived'].value_counts()  
0    549  
1    342  
Name: Survived, dtype: int64
```

Dapat dilihat bahwa data yang bernilai 0 (**survived**) ada 549 orang, sedangkan yang bernilai 1 (**not survived**) ada 342 orang.

13. Untuk menampilkan jumlah orang yang selamat maupun tidak selamat dalam bentuk grafik batang, maka data dihitung berdasarkan kolom **Survived**. Kemudian untuk menampilkan grafik batang bisa menggunakan *method* *plot(kind='bar')*.

```
1 dt = train_data['Survived'].value_counts()  
2 plt = dt.plot(kind = 'bar', rot=0)  
3 plt.set_xlabel('Selamat atau Tidak Selamat')  
4 plt.set_ylabel('Jumlah Penumpang')  
  
Text(0, 0.5, 'Jumlah Penumpang')
```



Dapat dilihat dari grafik bahwa penumpang yang tidak selamat (nilai=0) lebih banyak dibandingkan dengan penumpang yang selamat (nilai=1).

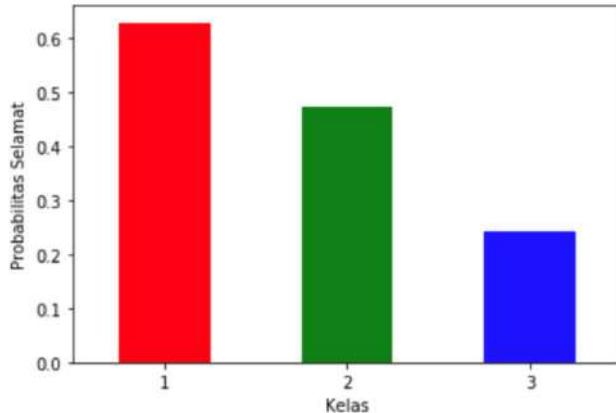
14. Tahap berikutnya adalah melihat tingkat kemungkinan keselamatan (*Survived*) berdasarkan kelas penumpang (*Pclass*).

```

1 dt = train_data[['Pclass', 'Survived']].groupby(
2     'Pclass').mean().Survived
3 plt = dt.plot(kind='bar', rot=0, color=['red', 'green', 'blue'])
4 plt.set_xlabel('Kelas')
5 plt.set_ylabel('Probabilitas Selamat')

Text(0, 0.5, 'Probabilitas Selamat')

```



Dapat dilihat dari grafik bahwa penumpang yang berada di Kelas 1 memiliki probabilitas keselamatan paling tinggi dibandingkan dengan kelas yang lainnya.

15. Dataset Titanic juga bisa dilihat secara multidimensi dengan kode python, yaitu dengan menggunakan *seaborn* (sns) dan method *catplot()*. Misalnya untuk menampilkan kelas penumpang (*Pclass*) vs. jenis kelamin (*Sex*). Maka tuliskan kode berikut:

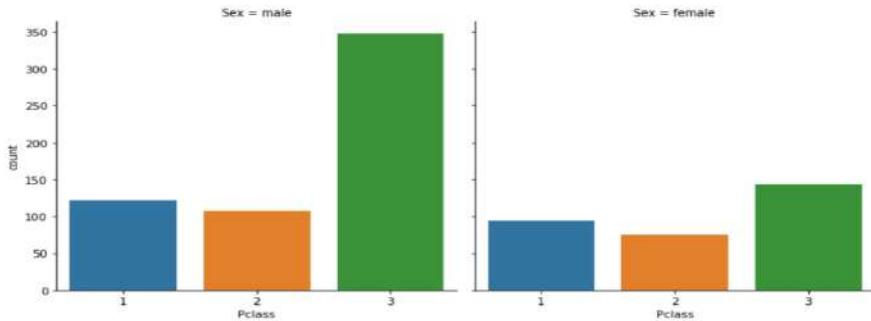
```

1 sns.catplot('Pclass', col = 'Sex', data = train_data,
2             kind = 'count')

```

Jika kode tersebut dieksekusi, maka akan ditampilkan grafik berikut ini:

```
1 sns.catplot('Pclass', col = 'Sex', data = train_data,
2             kind = 'count')
<seaborn.axisgrid.FacetGrid at 0x7fed40b90e50>
```



16. Dalam proses data mining, beberapa atribut terkadang tidak diperlukan karena dianggap tidak penting untuk dianalisis. Pada contoh dataset *Titanic*, kolom Id Penumpang (**PassengerId**) dan Nomor Tiket (**Ticket**) tidak penting untuk dianalisis sehingga bisa saja dihapus dari dataset. Begitu juga atribut yang memiliki data kosong banyak, juga bisa dihapus dari dataset karena justru akan mengganggu proses data mining jika tetap digunakan, misalnya kolom **Cabin**.

Untuk menghapus 3 kolom tersebut, maka gunakan *method drop()* terhadap *dataframe* yang digunakan, kemudian kode dieksekusi.

```

1 train_data = train_data.drop(columns=[
2     'Ticket', 'PassengerId', 'Cabin'])
3 train_data.head()

```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Th...	female	38.0	1	0	71.2833	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S

Hasil eksekusi kode menunjukkan bahwa 3 kolom yang telah dihapus sudah hilang dari dataset.

17. Pada tahap data preprocessing, penambahan fitur (kolom) baru berdasarkan atribut yang telah ada terkadang diperlukan untuk analisis lebih lanjut. Dalam contoh ini, jumlah penumpang, jumlah saudara/pasangan, dan jumlah orang tua/anak bisa digabung menjadi 1 kolom misalnya kolom *FamilySize* (Jumlah keluarga) yang dihitung dari jumlah saudara/pasangan ditambah dengan jumlah orang tua/anak dan termasuk si pemilik tiket dengan formula SibSp + Parch + 1.

Ketikkan kode berikut dan kemudian dieksekusi.

```

1 train_data['FamilySize'] = train_data['SibSp'] + train_data['Parch'] + 1
2 train_data.head()

```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	FamilySize
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	2
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Th...	female	38.0	1	0	71.2833	C	2
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	1
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	2
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	1

Hasil penambahan fitur baru bisa dilihat dari kolom *FamilySize* yang berada di paling kanan.

18. Pengolahan data mining, beberapa algoritma hanya bisa dilakukan pada data yang bertipe angka (*numeric*). Sehingga jika ada data yang dimiliki bertipe teks padahal penting untuk digunakan dalam data mining, maka perlu diubah (*transform*) menjadi tipe angka. Pada contoh dataset Titanic, atribut jenis kelamin (*Sex*) dan lokasi keberangkatan (*Embarked*) akan diubah menjadi tipe angka dengan ketentuan berikut:

Sex: male=0; female=1

Embarked: C=0; Q=1; S=2

Tuliskan kode berikut untuk mengubah tipe data kemudian dieksekusi:

```
1 train_data['Sex'] = train_data['Sex'].map({'male':0, 'female':1})  
2 train_data['Embarked'] = train_data['Embarked'].map({'C':0, 'Q':1, 'S':2})  
3 train_data.head()
```

```
1 train_data['Sex'] = train_data['Sex'].map({'male':0, 'female':1})  
2 train_data['Embarked'] = train_data['Embarked'].map({'C':0, 'Q':1, 'S':2})  
3 train_data.head()
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	FamilySize
0	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	7.2500	2.0	2
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	1	38.0	1	0	71.2833	0.0	2
2	-1	3			26.0	0	0	7.9250	2.0	1
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	53.1000	2.0	2
4	0	3	Allen, Mr. William Henry	0	35.0	0	0	8.0500	2.0	1

19. Berikutnya adalah melakukan preprocess terhadap nama penumpang. Di dataset Titanic, nama penumpang mengandung sebutan/gelar misalnya *Mr*, *Miss*, *Mrs*, *Dr*, *Lady*, dan lain-lain. Sebutan/gelar ini diperlukan dalam analisis data mining, namun nama penumpang tidak diperlukan. Sehingga gelar penumpang akan diekstraksi terlebih dahulu kemudian dikategorikan sebelum proses lebih lanjut. Untuk mengekstraksi data sebutan/gelar, maka diperlukan sebuah *Regular Expression (Regex)*, yaitu sebuah teks (string) yang mendefinisikan sebuah pola pencarian sehingga dapat membantu untuk melakukan *matching* (pencocokan), *locate* (pencarian), dan manipulasi teks.

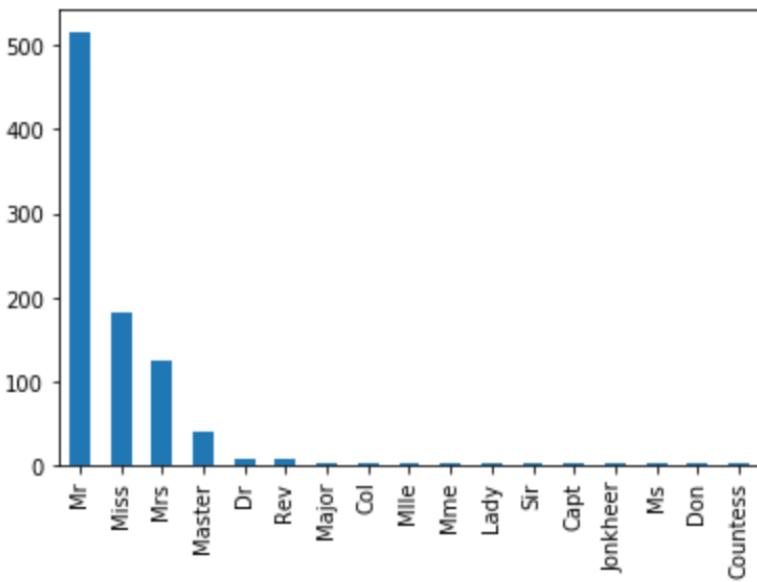
Regex yang diperlukan untuk mengekstraksi sebutan/gelar pada kolom *Name* adalah '([A-Za-z]+)\.' yang dimasukkan pada *method extract()*.

```
1 train_data['Title'] = train_data['Name'].str.extract(  
2   '([A-Za-z]+)\.', expand=False)  
3 train_data = train_data.drop(columns='Name')  
4 train_data.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	FamilySize	Title
0	0	3	0	22.0	1	0	7.2500	2.0	2	Mr
1	1	1	1	38.0	1	0	71.2833	0.0	2	Mrs
2	1	3	1	26.0	0	0	7.9250	2.0	1	Miss
3	1	1	1	35.0	1	0	53.1000	2.0	2	Mrs
4	0	3	0	35.0	0	0	8.0500	2.0	1	Mr

Untuk melihat sebaran data penumpang berdasarkan kategori sebutan/ gelar (*Title*) penumpang, maka bisa dilihat dengan grafik batang.

```
1 train_data['Title'].value_counts().plot(kind='bar')  
<matplotlib.axes._subplots.AxesSubplot at 0x7fed42c39310>
```



20. Berdasarkan grafik pada langkah 19, dapat dilihat bahwa terdapat banyak sebutan/gelar nama penumpang yang tidak biasa atau unik, seperti *Rev*, *Major*, *Col*, *Mlle*, dan lain-lainnya. Pada kasus ini, misalnya gelar nama hanya akan dikategorikan menjadi 5 jenis, yaitu *Master*, *Mr*, *Mrs*, *Miss*, dan *Others*. Sehingga diperlukan mengganti gelar-gelar yang unik berikut ini dengan kategori yang telah ditentukan:
1. *Dr*, *Rev*, *Col*, *Major*, *Countess*, *Sir*, *Jonkheer*, *Lady*, *Capt*, *Don* diganti dengan *Others*.
 2. *Ms* diganti dengan *Miss*.
 3. *Mme* diganti dengan *Mrs*.
 4. *Mlle* diganti dengan *Miss*.

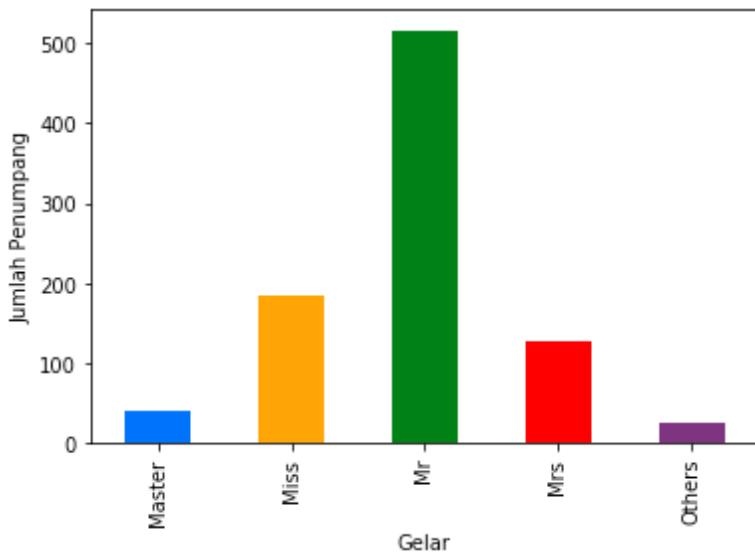
Sedangkan jika gelar nama penumpang sudah berupa *Master*, *Mr*, *Mrs*, dan *Miss*, maka tidak perlu diganti. Gunakan *method replace()* untuk mengganti gelar nama penumpang sesuai dengan kategori yang ditentukan.

```
1 train_data['Title'] = train_data['Title'].replace(  
2     ['Dr', 'Rev', 'Col', 'Major', 'Countess', 'Sir', 'Jonkheer',  
3      'Lady', 'Capt', 'Don'], 'Others')  
4 train_data['Title'] = train_data['Title'].replace('Ms', 'Miss')  
5 train_data['Title'] = train_data['Title'].replace('Mme', 'Mrs')  
6 train_data['Title'] = train_data['Title'].replace('Mlle', 'Miss')
```

21. Untuk mengetahui sebaran data penumpang berdasarkan sebutan/gelarnya, maka bisa dilihat dengan grafik batang dari 5 kategori sebutan tersebut.

```
1 plt = train_data['Title'].value_counts().sort_index().plot(  
2     kind='bar', color=['blue','orange', 'green', 'red', 'purple'])  
3 plt.set_xlabel('Gelar')  
4 plt.set_ylabel('Jumlah Penumpang')
```

Jika dieksekusi maka akan tampil grafik batang berikut ini.



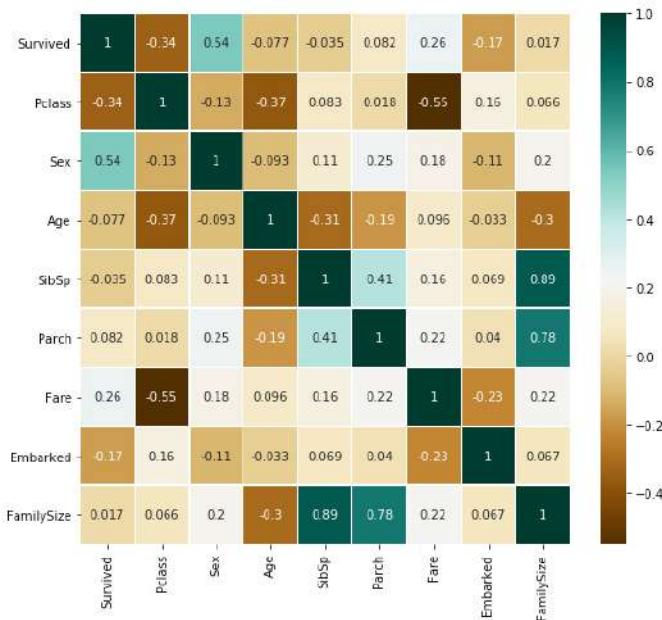
22. Sebelum melakukan analisis data lebih lanjut, para peneliti terkadang mencari korelasi antar atribut. Korelasi antar atribut bisa dicari dengan menerapkan *method corr()* pada dataframe yang dianalisis.

```
1 corr_matrix = train_data.corr()
```

Kemudian gambarkan dengan grafik *heatmap* menggunakan *library matplotlib*.

```
1 import matplotlib.pyplot as plt
2
3 plt.figure(figsize=(9, 8))
4 sns.heatmap(data = corr_matrix, cmap='BrBG', annot=True,
5               linewidths=0.2)
```

Sehingga akan ditampilkan grafik *heatmap*.



23. Berdasarkan hasil langkah 11, kita ketahui terdapat 3 atribut yang memiliki data kosong, yaitu *Age*, *Cabin*, dan *Embarked*. Namun atribut *Cabin* sudah dihilangkan dari dataset pada langkah 16, maka hanya tinggal 2 atribut yang memiliki data kosong. Untuk menangani data yang kosong, kita bisa melakukan beberapa pendekatan misalnya mengisi data kosong menggunakan nilai mayoritas, nilai rerata, nilai median, dan lain-lain tergantung datanya. Pada langkah ini, kita akan menangani data kosong pada atribut *Embarked* terlebih dahulu.

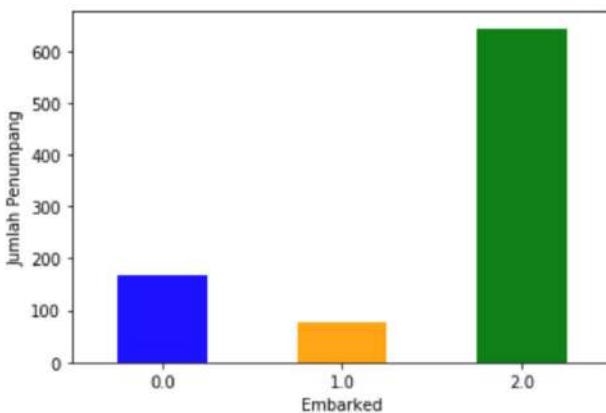
```
1 missing = train_data['Embarked'].isnull().sum()  
2 print("Jumlah data kosong pada atribut Embarked: ",missing)
```

Jumlah data kosong pada atribut Embarked: 2

24. Berdasarkan langkah 23, atribut ini hanya memiliki 2 data kosong, sehingga kita bisa mengisikannya berdasarkan nilai mayoritas pada atribut tersebut.

```
1 plt = train_data['Embarked'].value_counts().sort_index().plot(  
2     kind='bar', color=['blue','orange', 'green'], rot=0)  
3 plt.set_xlabel('Embarked')  
4 plt.set_ylabel('Jumlah Penumpang')
```

Text(0, 0.5, 'Jumlah Penumpang')



25. Nilai mayoritas pada atribut *Embarked* adalah "S" (*Southampton*). yaitu data dengan nilai 2 (berdasarkan langkah 18), sehingga data yang kosong bisa kita isikan dengan nilai 2 menggunakan method *fillna()*.

```
1 train_data['Embarked'] = train_data['Embarked'].fillna(2)
2 train_data.head()
```

26. Sedangkan untuk menangani data kosong pada atribut *Age*, kita akan menggunakan nilai median yang diambilkan dari semua data yang memiliki nilai yang sama dari 3 atribut lainnya, yaitu *SibSp*, *Parch*, dan *Pclass*. Jadi, mula-mula jika ada data yang bernilai kosong pada kolom *Age*, maka kita perlu melihat nilai atribut *SibSp*, *Parch*, dan *Pclass* pada data tersebut. Kemudian kita cari data lainnya dalam dataset yang memiliki nilai sama persis dalam atribut *SibSp*, *Parch*, dan *Pclass*. Pencarian ini bisa saja menghasilkan beberapa data lainnya yang memiliki nilai yang sama pada ketiga atribut tersebut. Data-data yang bernilai sama pada ketiga atribut tersebut, kemudian dihitung nilai mediannya yang kemudian nilai median ini digunakan untuk mengisi data kosong dalam atribut *Age*. Namun jika tidak ditemukan data yang sama dari ketiga atribut tersebut dalam dataset, maka nilai *Age* yang kosong akan diisi dengan nilai median atribut *Age* secara keseluruhan. Langkah ini diulang beberapa kali sebanyak jumlah data *Age* yang kosong dalam dataset.
27. Mula-mula kita identifikasi indeks data *Age* yang bernilai kosong.

```
1 NaN_indexes = train_data['Age'][train_data['Age'].isnull()].index
2 print(NaN_indexes)

Int64Index([ 5, 17, 19, 26, 28, 29, 31, 32, 36, 42,
             ...
            832, 837, 839, 846, 849, 859, 863, 868, 878, 888],
            dtype='int64', length=177)
```

28. Sesuai langkah 27, kita akan menggantikan data kosong pada kolom *Age* dengan nilai mediannya.

```
1 for i in NaN_indexes:
2     pred_age = train_data['Age'][((train_data.SibSp == train_data.iloc[i]["SibSp"]) &
3                                     (train_data.Parch == train_data.iloc[i]["Parch"]) &
4                                     (train_data.Pclass == train_data.iloc[i]["Pclass"]))].median()
5
6     if np.isnan(pred_age):
7         train_data['Age'].iloc[i] = train_data['Age'].median()
8     else:
9         train_data['Age'].iloc[i] = pred_age
```

/Users/yusufnugroho/opt/anaconda3/lib/python3.7/site-packages/pandas/core/indexing.py:670:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
self._setitem_with_indexer(indexer, value)

Jika muncul informasi terkait dengan: *SettingWithCopyWarning*, maka abaikan saja.

29. Untuk melihat jumlah data yang kosong pada atribut *Age* setelah preprocess, ketikkan kode berikut:

```
1 train_data.isnull().sum()
```

```
Survived      0
Pclass        0
Sex           0
Age           0
SibSp         0
Parch         0
Fare          0
Embarked      0
FamilySize    0
Title          0
dtype: int64
```

Dapat dilihat bahwa sekarang semua atribut tidak memiliki data yang kosong (*missing values*).

30. Untuk melihat dataset versi terakhir setelah melakukan preprocessing, bisa dengan cara menuliskan nama *dataframe*-nya, yaitu `train_data`. Data *train* dalam versi final setelah melalui serangkaian preprocessing. Data final inilah yang kemudian siap untuk diolah dalam data mining.

1 `train_data`

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	FamilySize	Title
0	0	3	0	22.0	1	0	7.2500	2.0	2	Mr
1	1	1	1	38.0	1	0	71.2833	0.0	2	Mrs
2	1	3	1	26.0	0	0	7.9250	2.0	1	Miss
3	1	1	1	35.0	1	0	53.1000	2.0	2	Mrs
4	0	3	0	35.0	0	0	8.0500	2.0	1	Mr
...
886	0	2	0	27.0	0	0	13.0000	2.0	1	Others
887	1	1	1	19.0	0	0	30.0000	2.0	1	Miss
888	0	3	1	13.5	1	2	23.4500	2.0	4	Miss
889	1	1	0	26.0	0	0	30.0000	0.0	1	Mr
890	0	3	0	32.0	0	0	7.7500	1.0	1	Mr

891 rows × 10 columns

7.5 Tugas

Dengan menggunakan dataset *train.csv*, kerjakan tugas berikut ini:

- Lakukan kembali langkah 13 pada prosedur praktikum untuk melihat data atribut lainnya dengan grafik batang, misalnya *Pclass*, *Sex*, dan *Embarked*!
- Lakukan kembali langkah 14 pada prosedur praktikum untuk melihat **probabilitas keselamatan** (*Survived*) berdasarkan jenis kelamin (*Sex*), lokasi keberangkatan (*Embarked*), jumlah saudara/pasangan yang ikut (*SibSp*), dan jumlah orang tua/anak yang ikut (*Parch*)!

3. Ulangi kembali langkah 15 pada prosedur praktikum untuk melihat multidimensi terhadap atribut kelas penumpang (*Pclass*) vs lokasi keberangkatan (*Embarked*), dan jenis kelamin (*Sex*) vs lokasi keberangkatan (*Embarked*)!
4. Ubahlah data sebutan/gelar penumpang (*Title*) menjadi data angka dengan ketentuan sebagai berikut:
Master: 0, Miss: 1, Mr: 2, Mrs: 3, Others: 4
5. Carilah nilai korelasi antar atribut termasuk atribut *Title* setelah diubah menjadi data angka dengan menggunakan *heatmap*!

Modul 8

Algoritma Naïve Bayes

8.1 Tujuan

1. Mahasiswa mampu menggunakan dan membuat model klasifikasi dengan teorema Naïve Bayes.
2. Mahasiswa mampu menerapkan algoritma Naïve Bayes terhadap studi kasus tertentu.

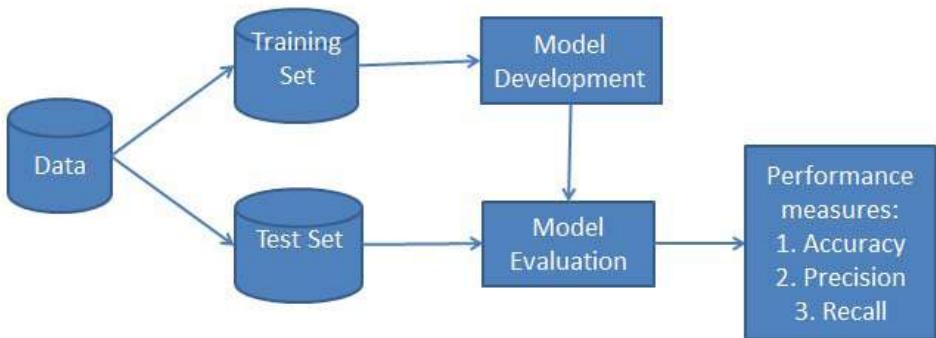
8.2 Landasan Teori

Algoritma Naive Bayes merupakan sebuah metode klasifikasi menggunakan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes. Algoritma Naive Bayes memprediksi peluang kejadian di masa depan berdasarkan pengalaman di masa sebelumnya, sehingga algoritma Naive Bayes sangat cocok digunakan untuk klasifikasi biner maupun multiclass. Metode yang juga dikenal sebagai Naive Bayes Classifier ini menerapkan teknik supervised learning yang mampu mengklasifikasi objek di masa depan dengan menetapkan label kelas ke instance/catatan menggunakan probabilitas bersyarat. Probabilitas bersyarat adalah ukuran peluang suatu peristiwa yang terjadi berdasarkan peristiwa lain yang telah (dengan asumsi, praduga, pernyataan, atau terbukti) terjadi.

Istilah supervised merujuk pada klasifikasi training data yang sudah diberi label dengan kelas. Misalnya, sebuah transaksi penipuan telah ditandai sebagai data transaksional penipuan, kemudian kita bisa mengklasifikasikan transaksi di masa depan menjadi penipuan/non-

penipuan dengan data yang sudah diberi label tersebut, maka jenis klasifikasi itu akan disebut sebagai supervised learning.

Ada dua proses penting yang dilakukan saat melakukan klasifikasi. Proses yang pertama adalah **learning (training)** yaitu proses pembelajaran menggunakan training set. Pada kasus *Naïve Bayesian Classifier*, perhitungan probabilitas dari data berdasarkan data pembelajaran dilakukan. Proses yang kedua adalah proses **testing** yaitu menguji model menggunakan data testing. Gambar 8.1 memperlihatkan alur dari kedua proses tersebut.



Gambar 8.1 Tahapan Proses Klasifikasi

Metode Bayes menggunakan probabilitas bersyarat sebagai dasarnya. Dalam ilmu probabilitas bersyarat dinyatakan sebagai:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Keterangan:

$P(A|B)$: Probabilitas A terjadi dengan bukti bahwa B telah terjadi (probabilitas superior)

$P(B|A)$: Probabilitas B terjadi dengan bukti bahwa A telah terjadi

$P(A)$: Peluang terjadinya A

$P(B)$: Peluang terjadinya B

Salah satu manfaat algoritma naïve bayes adalah untuk melakukan prediksi terhadap data-data tertentu. Prediksi terhadap data yang akan datang bisa dilakukan berdasarkan hasil pembelajaran terhadap data training. Data training diambil dari data yang terdahulu, sedangkan data uji (testing) bisa diambil dari data-data yang sedang atau akan terjadi.

8.3 Alat dan Bahan

1. Jupyter Notebook.
2. Dataset iris.csv
3. Modul Praktikum Data Warehousing dan Data Mining.
4. Bahasa pemrograman python dan beberapa library python.

8.4 Langkah-langkah Praktikum

Langkah-langkah menggunakan algoritma naïve bayes dengan bahasa pemrograman python adalah sebagai berikut:

1. Buka Jupyter notebook yang sudah terinstall pada komputer yang digunakan.
2. Buat file baru pada jupyter notebook dengan mengklik menu New-Python 3, secara otomatis akan keluar window baru untuk melakukan coding pada jupyter notebook. Pastikan file yang dibuat dengan dataset yang digunakan berada dalam satu folder.



3. Pada baris pertama tuliskan beberapa library yang akan digunakan dalam praktikum. Ada beberapa library seperti numpy, pandas, matplotlib, seaborn dan lain sebagainya.
-

PRAKTIKUM DWDM NAIVE BAYES

```
In [1]: #Import Library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
```

4. Setelah itu lakukan read dataset menggunakan library pandas, disini kita menggunakan dataset iris dengan ekstensi .csv. Kita akan melakukan beberapa hal untuk mengekplorasi data yang digunakan. Pertama kita akan melihat isi dataset yang dipakai menggunakan fungsi head, fungsi tersebut akan menampilkan isi data beserta atribut yang ada. Terdapat 6 atribut yang digunakan pada dataset tersebut yaitu Id, SepalLengthCm, SepalWidthCm, PetalLengthCM, PetalWidthCm dan Species yang nanti digunakan sebagai label klasifikasi.

```
In [2]: iris=pd.read_csv('iris.csv')
```

```
In [3]: iris.head(5)
```

```
Out[3]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

5. Kita juga bisa mengecek bentuk dari data yang digunakan menggunakan fungsi shape, bisa juga melihat value apa saja yang ada pada atribut Species.

```
In [4]: # Cek jumlah baris and kolom  
iris.shape
```

```
Out[4]: (150, 6)
```

```
In [5]: iris['Species'].unique()
```

```
Out[5]: array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

6. Hal lain yang harus kita lakukan adalah memeriksa apakah dataset memiliki nilai kosong (null) atau tidak, hal ini bisa dilakukan dengan menggunakan fungsi info(). Terlihat tidak ada data yang kosong pada semua atribut, ini menunjukkan dataset yang digunakan sangat baik.

```
In [6]: iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 150 entries, 0 to 149  
Data columns (total 6 columns):  
 #   Column           Non-Null Count  Dtype     
---  --     
 0   Id               150 non-null    int64    
 1   SepalLengthCm  150 non-null    float64  
 2   SepalWidthCm   150 non-null    float64  
 3   PetalLengthCm  150 non-null    float64  
 4   PetalWidthCm   150 non-null    float64  
 5   Species         150 non-null    object    
dtypes: float64(4), int64(1), object(1)  
memory usage: 7.2+ KB
```

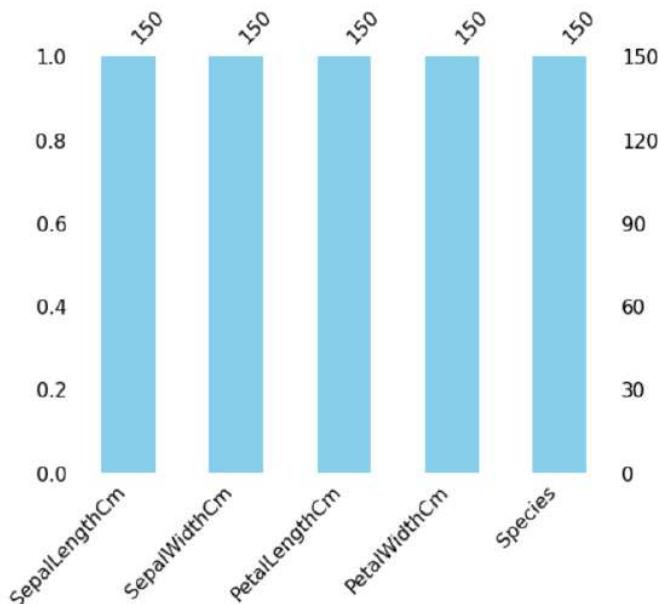
7. Dari semua atribut yang ada kita akan memakai lima fitur saja yaitu SepalLengthCm, SepalWidthCm, PetalLengthCM, PetalWidthCm dan Species. Sehingga atribut Id akan dihapus karena fitur tersebut tidak memiliki pengaruh dalam proses klasifikasi, penghapusan atribut bisa menggunakan fungsi drop.

```
In [7]: iris.drop(columns="Id",inplace=True)  
iris.isnull().sum()
```

```
Out[7]: SepalLengthCm      0  
SepalWidthCm      0  
PetalLengthCm     0  
PetalWidthCm      0  
Species           0  
dtype: int64
```

8. Kita juga bisa mengecek data yang kosong menggunakan grafik.

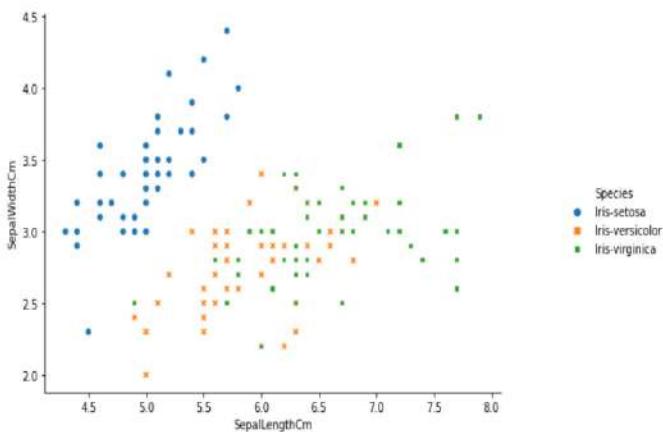
```
In [8]: import missingno as msno  
msno.bar(iris,figsize=(8,6),color='skyblue')  
plt.show()
```



9. Proses selanjutnya adalah melakukan visualisasi data, proses ini sangat penting dilakukan untuk mengetahui sebaran data yang ada dalam dataset.

Data Visualization

```
In [9]: # Scatterplot
g=sns.relplot(x='SepalLengthCm',y='SepalWidthCm',data=iris,hue='Species',style='Species')
g.fig.set_size_inches(10,5)
plt.show()
```



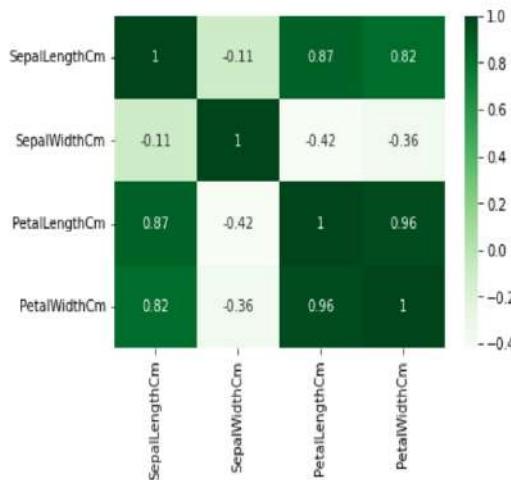
10. Kemudian kita bisa mencari korelasi pada setiap fitur yang digunakan untuk melakukan klasifikasi.

```
In [10]: # Cek korelasi antar fitur  
iris.corr()
```

Out[10]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000

```
In [11]: # Draw heatmap  
sns.heatmap(iris.corr(), annot=True, cmap='Greens')  
plt.show()
```



11. Hal penting lainnya yang harus dilakukan adalah memisahkan antara fitur dan label, atribut Species akan digunakan sebagai label dan atribut lainnya digunakan sebagai fitur pada proses klasifikasi.

Data Preprocessing

```
In [12]: # Pisahkan antara fitur dan Label  
X = iris.drop('Species', axis = 1)  
y = iris['Species']
```

12. Kita bisa mengecek atribut yang sudah terpisah dengan menampilkan variabel X yang merupakan fitur dan variabel y sebagai label.

```
In [13]: X
```

Out[13]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

```
In [14]: y
```

Out[14]:

```
0      Iris-setosa  
1      Iris-setosa  
2      Iris-setosa  
3      Iris-setosa  
4      Iris-setosa  
      ...  
145    Iris-virginica  
146    Iris-virginica  
147    Iris-virginica  
148    Iris-virginica  
149    Iris-virginica  
Name: Species, Length: 150, dtype: object
```

13. Pada dataset iris, label yang digunakan bukan merupakan data numerical, padahal mesin hanya bisa memahami angka, sehingga kita harus merubah isi dari label menjadi angka, proses ini dinamakan dengan encoding.

```
In [15]: # Label encoding
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
y

Out[15]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
```

14. Sebelum memasuki proses klasifikasi data perlu dibagi menjadi dua bagian yaitu data training dan data testing. Data training akan digunakan untuk pembuatan model, sedangkan data testing digunakan untuk melakukan evaluasi pada model yang sudah dibuat, proses pembagian dataset ini menggunakan library sklearn. Terlihat pada gambar dataset terbagi menjadi 120 untuk data training dan 30 untuk data testing.

```
In [16]: # Splitting data menjadi train data dan test data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size= 0.2, random_state=0)

print('The shape of X_train is: {}'.format(X_train.shape))
print('The shape of X_test is: {}'.format(X_test.shape))
print('The shape of y_train is: {}'.format(y_train.shape))
print('The shape of y_test is: {}'.format(y_test.shape))

The shape of X_train is: (120, 4)
The shape of X_test is: (30, 4)
The shape of y_train is: (120,)
The shape of y_test is: (30,)
```

15. Tahap selanjutnya adalah pembuatan model menggunakan algoritma Naive Bayes. Kita bisa memanggilnya dari library sklearn dan menaruhnya pada variabel model.

Membuat Model

```
# Model Training
model = GaussianNB()
model.fit(X_train,y_train)
```

▼ GaussianNB
GaussianNB()

16. Kita lakukan training model menggunakan algoritma naive bayes pada data training. Disini kita mendapatkan akurasi pada saat training memperoleh skor 95%.

```
In [18]: # Prediksi pada data train
pred_train = model.predict(X_train)

cm = confusion_matrix(y_train, pred_train)

# confusion matrix
print('Confusion matrix Naive Bayes\n',cm)
print('')

# Akurasi
print('Akurasi pada saat training: {}' .format(accuracy_score(y_train,pred_train)))# confusion matrix

Confusion matrix Naive Bayes
[[39  0  0]
 [ 0 34  3]
 [ 0  3 41]]

Akurasi pada saat training: 0.95
```

17. Langkah terakhir adalah menguji model yang dibuat dengan data testing yang sudah disiapkan. Pada saat evaluasi model kita mendapatkan nilai yang sama pada akurasi, presisi, recall dan f1-score, yaitu sebesar 96,7%.

```
In [19]: # Prediksi pada data test
pred_test = model.predict(X_test)

In [20]: cm = confusion_matrix(y_test, pred_test)
accuracy = accuracy_score(y_test,pred_test)
precision =precision_score(y_test, pred_test,average='micro')
recall = recall_score(y_test, pred_test,average='micro')
f1 = f1_score(y_test,pred_test,average='micro')
print('Confusion matrix Naive Bayes\n',cm)
print('')
print('Akurasi pada data test: %.3f' %accuracy)
print('precision: %.3f' %precision)
print('recall: %.3f' %recall)
print('f1-score: %.3f' %f1)

Confusion matrix Naive Bayes
[[11  0  0]
 [ 0 13  0]
 [ 0  1  5]]

Akurasi pada data test: 0.967
precision: 0.967
recall: 0.967
f1-score: 0.967
```

8.5 Tugas

1. Buatlah eksplorasi data dari dataset Breast-Cancer.csv!
2. Cari 4 fitur yang memiliki pengaruh paling besar terhadap proses klasifikasi!
3. Gunakan 4 fitur yang sudah dipilih untuk melakukan klasifikasi!
4. Buatlah model untuk algoritma Naive Bayes dan hitunglah nilai akurasi, presisi, recall dan F1-score!

Modul 9

Algoritma Decision Tree (Pohon Keputusan)

9.1 Tujuan

1. Mahasiswa mampu menggunakan dan membuat model klasifikasi dengan teorema pohon keputusan.
2. Mahasiswa mampu menerapkan algoritma pohon keputusan terhadap studi kasus tertentu.

9.2 Landasan Teori

Decision tree merupakan suatu struktur yang digunakan untuk membantu proses pengambilan keputusan. Disebut sebagai “tree” karena struktur ini menyerupai sebuah pohon lengkap dengan akar, batang, dan percabangannya. Dalam data science, struktur decision tree dapat membantu mengambil keputusan efektif dan tetap memperhatikan kemungkinan hasil serta konsekuensinya. Konsep dari pohon keputusan ini adalah mengubah data menjadi decision tree dan aturan-aturan keputusan. Manfaat utama dari penggunaan decision tree adalah kemampuannya untuk mem-break down proses pengambilan keputusan yang kompleks menjadi lebih simple, sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan.

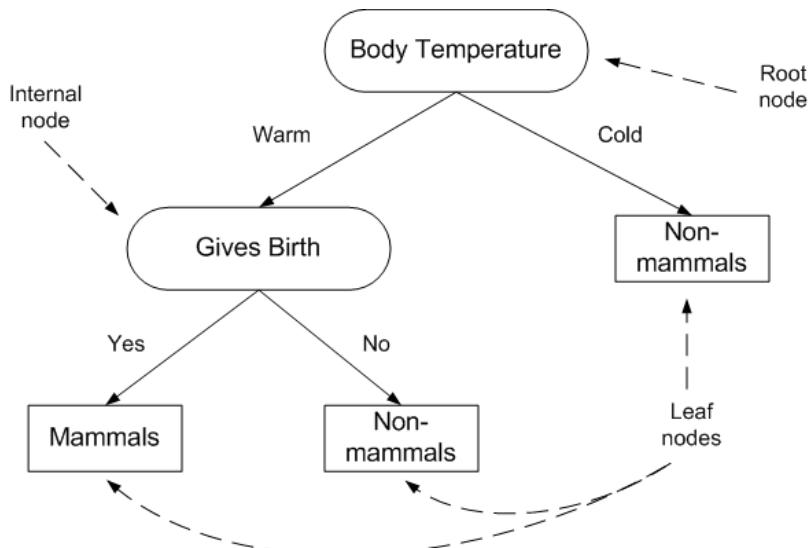
Metode ini merupakan metode yang berusaha menemukan fungsi-fungsi pendekatan yang bernilai diskrit dan tahan terhadap data-data yang memiliki kesalahan (*noisy data*) serta mampu mempelajari ekspresi-

ekspresi disjunctive seperti ekspresi OR. *Iterative Dychotomizer version 3* (ID3) adalah salah satu jenis *decision tree* yang umumnya digunakan untuk menemukan aturan yang diharapkan bisa berlaku umum untuk data-data yang tidak lengkap atau yang belum pernah kita ketahui. Salah satu varian lainnya adalah C4.5.

Pohon (Tree) adalah sebuah struktur data yang terdiri dari simpul (node) dan rusuk (edge). Simpul pada sebuah pohon terdiri dari 3:

1. Simpul akar (*root node*)
2. Simpul percabangan/internal (*branch/internal node*)
3. Simpul daun (*leaf node*)

Pohon keputusan merupakan representasi sederhana dari teknik klasifikasi untuk sejumlah kelas berhingga. Simpul internal maupun simpul akar ditandai dengan nama atribut, rusuk diberi label nilai atribut yang mungkin, sedangkan simpul daun ditandai dengan kelas-kelas yang berbeda. Objek (*record*) diklasifikasikan dengan mengikuti suatu jalur (*path*) yang dimulai dari simpul akar sesuai dengan nilai atribut dalam *record* tersebut. Gambar 9.1 menunjukkan contoh klasifikasi pohon keputusan mamalia.



Gambar 9.1 Klasifikasi Pohon Keputusan Binatang Mamalia

9.3 Alat dan Bahan

1. Jupyter Notebook.
2. Dataset iris.csv
3. Modul Praktikum Data Warehousing dan Data Mining.
4. Bahasa pemrograman python dan beberapa library python.

9.4 Langkah-langkah Praktikum

Pada kegiatan praktikum ini, kita menggunakan pohon keputusan untuk membuat klasifikasi pada data iris. Dengan metode ini, suatu data uji dapat diklasifikasikan berdasarkan kelas datanya.

Langkah-langkah menggunakan algoritma decision tree dengan bahasa pemrograman python adalah sebagai berikut:

1. Bukalah jupyter notebook yang sudah terinstal pada komputer yang digunakan.

- Buatlah file baru pada jupyter notebook dengan mengklik menu New-Python 3, secara otomatis akan keluar window baru untuk melakukan coding pada jupyter nootebook. Pastikan file yang dibuat dengan dataset yang digunakan berapa dalam satu folder.



- Pada baris pertama tuliskan beberapa library yang akan digunakan dalam praktikum. Ada beberapa library seperti numpy, pandas, matplotlib, seaborn dan lain sebagainya.

PRAKTIKUM DWDM DECISION TREE

```
In [1]: #Import Library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score
```

- Setelah itu lakukan read dataset menggunakan library pandas, disini kita menggunakan dataset iris dengan ekstensi .csv. Kita akan melakukan beberapa hal untuk mengekplorasi data yang digunakan. Pertama kita akan melihat isi dataset yang dipakai menggunakan fungsi head, fungsi tersebut akan menampilkan isi data beserta atribut yang ada. Terdapat 6 atribut yang digunakan pada dataset tersebut yaitu Id, SepalLengthCm, SepalWidthCm, PetalLengthCM,

PetalWidthCm dan Species yang nanti digunakan sebagai label klasifikasi.

```
In [2]: iris=pd.read_csv('iris.csv')
```

```
In [3]: iris.head(5)
```

```
Out[3]:
```

		Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1		5.1	3.5	1.4	0.2	Iris-setosa
1	2		4.9	3.0	1.4	0.2	Iris-setosa
2	3		4.7	3.2	1.3	0.2	Iris-setosa
3	4		4.6	3.1	1.5	0.2	Iris-setosa
4	5		5.0	3.6	1.4	0.2	Iris-setosa

5. Kita juga bisa mengecek bentuk dari data yang digunakan menggunakan fungsi shape, bisa juga melihat value apa saja yang ada pada atribut Species.

```
In [4]: # Cek jumlah baris and kolom  
iris.shape
```

```
Out[4]: (150, 6)
```

```
In [5]: iris['Species'].unique()
```

```
Out[5]: array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

6. Hal lain yang harus kita lakukan adalah memeriksa apakah dataset memiliki nilai kosong (null) atau tidak, hal ini bisa dilakukan dengan menggunakan fungsi info(). Terlihat tidak ada data yang kosong pada semua atribut, ini menunjukkan dataset yang digunakan sangat baik.

```
In [6]: iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  --  
 0   Id          150 non-null    int64  
 1   SepalLengthCm 150 non-null    float64 
 2   SepalWidthCm  150 non-null    float64 
 3   PetalLengthCm 150 non-null    float64 
 4   PetalWidthCm  150 non-null    float64 
 5   Species      150 non-null    object  
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

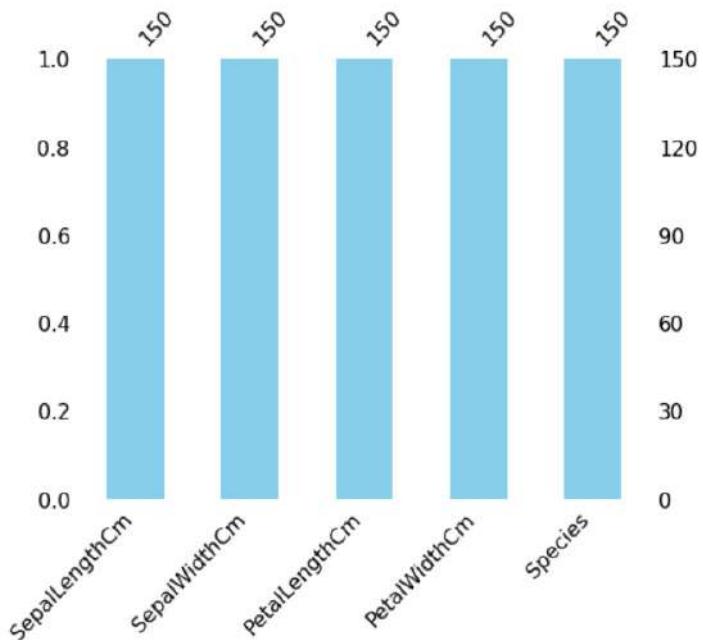
7. Dari semua atribut yang ada kita akan memakai lima fitur saja yaitu SepalLengthCm, SepalWidthCm, PetalLengthCM, PetalWidthCm dan Species. Sehingga atribut Id akan dihapus karena fitur tersebut tidak memiliki pengaruh dalam proses klasifikasi, penghapusan atribut bisa menggunakan fungsi drop.

```
In [7]: iris.drop(columns="Id",inplace=True)
iris.isnull().sum()
```

```
Out[7]: SepalLengthCm      0
SepalWidthCm       0
PetalLengthCm     0
PetalWidthCm      0
Species           0
dtype: int64
```

8. Kita juga bisa mengecek data yang kosong menggunakan grafik.

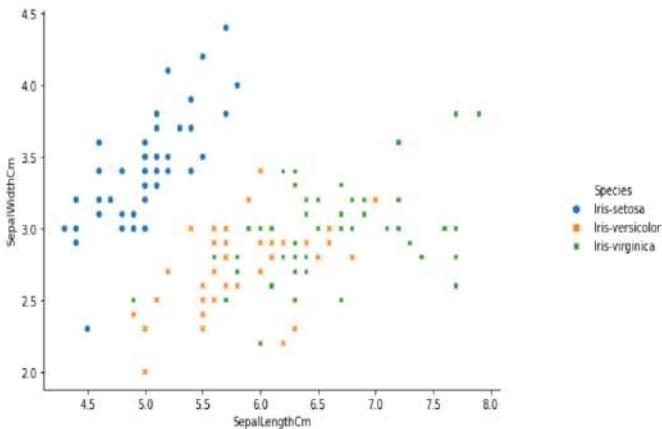
```
In [8]: import missingno as msno  
msno.bar(iris,figsize=(8,6),color='skyblue')  
plt.show()
```



9. Proses selanjutnya adalah melakukan visualisasi data, proses ini sangat penting dilakukan untuk mengetahui sebaran data yang ada dalam dataset.

Data Visualization

```
In [9]: # Scatterplot
g=sns.relplot(x='SepalLengthCm',y='SepalWidthCm',data=iris,hue='Species',style='Species')
g.fig.set_size_inches(10,5)
plt.show()
```



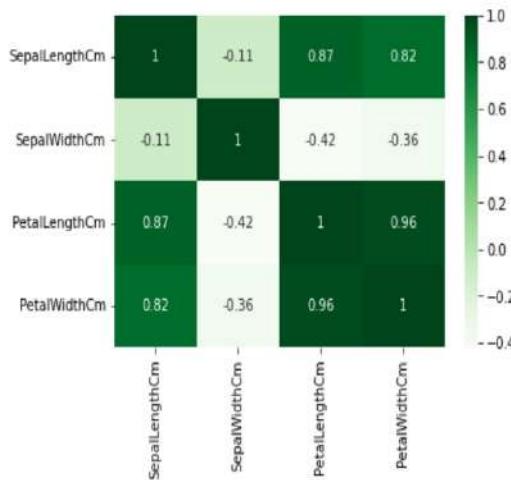
10. Kemudian kita bisa mencari korelasi pada setiap fitur yang digunakan untuk melakukan klasifikasi.

```
In [10]: # Cek korelasi antar fitur  
iris.corr()
```

Out[10]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000

```
In [11]: # Draw heatmap  
sns.heatmap(iris.corr(), annot=True, cmap='Greens')  
plt.show()
```



11. Hal penting lainnya yang harus dilakukan adalah memisahkan antara fitur dan label, atribut Species akan digunakan sebagai label dan atribut lainnya digunakan sebagai fitur pada proses klasifikasi.

Data Preprocessing

```
In [12]: # Pisahkan antara fitur dan Label  
X = iris.drop('Species', axis = 1)  
y = iris['Species']
```

12. Kita bisa mengecek atribut yang sudah terpisah dengan menampilkan variabel X yang merupakan fitur dan variabel y sebagai label.

```
In [13]: X
```

```
out[13]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

```
In [14]: y
```

```
out[14]: 0      Iris-setosa  
1      Iris-setosa  
2      Iris-setosa  
3      Iris-setosa  
4      Iris-setosa  
      ...  
145    Iris-virginica  
146    Iris-virginica  
147    Iris-virginica  
148    Iris-virginica  
149    Iris-virginica  
Name: Species, Length: 150, dtype: object
```

13. Pada dataset iris, label yang digunakan bukan merupakan data numerical, padahal mesin hanya bisa memahami angka, sehingga kita harus merubah isi dari label menjadi angka, proses ini dinamakan dengan encoding.

```
In [15]: # Label encoding
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
y

Out[15]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
```

14. Sebelum memasuki proses klasifikasi data perlu dibagi menjadi dua bagian yaitu data training dan data testing. Data training akan digunakan untuk pembuatan model, sedangkan data testing digunakan untuk melakukan evaluasi pada model yang sudah dibuat. proses pembagian dataset ini menggunakan library sklearn. Terlihat pada gambar dataset terbagi menjadi 120 untuk data training dan 30 untuk data testing.

```
In [16]: # Splitting data menjadi train data dan test data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size= 0.2, random_state=0)

print('The shape of X_train is: {}'.format(X_train.shape))
print('The shape of X_test is: {}'.format(X_test.shape))
print('The shape of y_train is: {}'.format(y_train.shape))
print('The shape of y_test is: {}'.format(y_test.shape))

The shape of X_train is: (120, 4)
The shape of X_test is: (30, 4)
The shape of y_train is: (120,)
The shape of y_test is: (30,)
```

15. Tahap selanjutnya adalah pembuatan model menggunakan algoritma Naive Bayes. Kita bisa memanggilnya dari library sklearn dan menaruhnya pada variabel model.

Membuat Model

```
# Model Training
model = DecisionTreeClassifier()
model.fit(X_train,y_train)

* DecisionTreeClassifier
DecisionTreeClassifier()
```

16. Kita lakukan training model menggunakan algoritma naive bayes pada data training. Disini kita mendapatkan akurasi pada saat training memperoleh skor 100%.

```

# Prediksi pada data train
pred_train = model.predict(X_train)

cm = confusion_matrix(y_train, pred_train)

# confusion matrix
print('Confusion matrix Decision Tree\n',cm)
print('')

# Akurasi
print('Akurasi pada saat training: {}' .format(accuracy_score(y_train,pred_train)))# confusion matrix

Confusion matrix Decision Tree
[[39  0  0]
 [ 0 37  0]
 [ 0  0 44]]

Akurasi pada saat training: 1.0

```

17. Langkah terakhir adalah menguji model yang dibuat dengan data testing yang sudah disiapkan. Pada saat evaluasi model kita mendapatkan nilai yang sama pada akurasi, presisi, recall dan f1-score, yaitu sebesar 100%.

```

In [19]: # Prediksi pada data test
pred_test = model.predict(X_test)

In [20]: cm = confusion_matrix(y_test, pred_test)
accuracy = accuracy_score(y_test,pred_test)
precision =precision_score(y_test, pred_test,average='micro')
recall = recall_score(y_test, pred_test,average='micro')
f1 = f1_score(y_test,pred_test,average='micro')
print('Confusion matrix for DecisionTree\n',cm)
print('')
print('Akurasi pada data test: %.3f' %accuracy)
print('precision: %.3f' %precision)
print('recall: %.3f' %recall)
print('f1-score: %.3f' %f1)

Confusion matrix for DecisionTree
[[11  0  0]
 [ 0 13  0]
 [ 0  0  6]]

Akurasi pada data test: 1.000
precision: 1.000
recall: 1.000
f1-score: 1.000

```

9.5 Tugas

1. Buatlah eksplorasi data dari dataset heart_failure.csv!
2. Cari 5 fitur yang memiliki pengaruh paling besar terhadap proses klasifikasi!
3. Gunakan 5 fitur yang sudah dipilih untuk melakukan klasifikasi!
4. Buatlah model untuk algoritma decision tree dan hitunglah nilai akurasi, presisi, recall dan F1-score!

Modul 10

Regresi Linier Sederhana

10.1 Tujuan

1. Mahasiswa mampu menggunakan metode regresi linier.
2. Mahasiswa mampu melakukan analisis regresi linier.
3. Mahasiswa mampu menerapkan metode regresi linier dalam kasus nyata.

10.2 Landasan Teori

Ketika kita ingin memahami model machine learning, salah satu hal pertama yang biasanya kita temui adalah Regresi Linier Sederhana. Regresi linier adalah metode statistika yang digunakan untuk membentuk model hubungan antara variabel terikat (dependent; Y) dengan satu atau lebih variabel bebas (independent; X). Apabila banyaknya variabel bebas hanya ada satu, disebut sebagai regresi linier sederhana, sedangkan apabila terdapat lebih dari 1 variabel bebas, disebut sebagai regresi linier berganda.

Regrisi linear sederhana ataupun regresi linier berganda pada intinya memiliki beberapa tujuan, yaitu :

1. Menghitung nilai estimasi rata-rata dan nilai variabel terikat berdasarkan pada nilai variabel bebas.
2. Menguji hipotesis karakteristik dependensi
3. Meramalkan nilai rata-rata variabel bebas dengan didasarkan pada nilai variabel bebas diluar jangkauan sampel.

Di dalam suatu model regresi akan ditemukan koefisien-koefisien. Koefisien pada model regresi sebenarnya adalah nilai duga parameter di dalam model regresi untuk kondisi yang sebenarnya (*true condition*), sama halnya dengan statistik *mean* (rata-rata) pada konsep statistika dasar. Hanya saja, koefisien-koefisien untuk model regresi merupakan suatu nilai rata-rata yang berpeluang terjadi pada variabel Y (variabel terikat) bila suatu nilai X (variabel bebas) diberikan.

$$Y = \beta_0 + \beta_1 \times X$$

Secara matematika, permasalahan regresi linear ini dapat dimodelkan seperti persamaan di atas. Di sini, terdapat dua konstanta (β) yang merupakan Intercept dan Slope. Kita mungkin mengenali ekspresi ini di mata kuliah aljabar, di mana ekspresi umum untuk garis lurus adalah:

$$y = c + mx$$

dimana c adalah intercept dan m adalah kemiringan. Itulah yang akan kita lakukan ketika kita menggunakan metode regresi linear. Pada metode ini, kita mencoba menyesuaikan garis lurus untuk mengamati hubungan antara variabel input dan output dan kemudian menggunakannya lebih lanjut untuk memprediksi output dari input yang tidak terlihat.

Pada analisis regresi sederhana, ada beberapa asumsi dan persyaratan yang perlu diperiksa dan diuji, beberapa diantaranya adalah :

1. Variabel bebas tidak berkorelasi dengan *disturbance term (Error)*. Nilai disturbance term sebesar 0 atau dengan simbol sebagai berikut: $(E(U/X)) = 0$,
2. Jika variabel bebas lebih dari satu, maka antara variabel bebas (*explanatory*) tidak ada hubungan linier yang nyata,
3. Model regresi dikatakan layak jika angka signifikansi ANOVA sebesar < 0.05 ,

4. Variabel independen (*predictor*) yang digunakan sebagai variabel bebas harus layak. Kelayakan ini diketahui jika angka *Standard Error of Estimate < Standard Deviation*,
5. Koefisien regresi harus signifikan. Pengujian dilakukan dengan Uji T. Koefisien regresi signifikan jika $T \text{ hitung} > T \text{ table}$ (nilai kritis).
6. Model regresi dapat diterangkan dengan menggunakan nilai koefisien determinasi ($KD = r^2 \times 100\%$) semakin besar nilai tersebut maka model semakin baik. Jika nilai mendekati 1 maka model regresi semakin baik,
7. Data harus berdistribusi normal,
8. Data berskala interval atau rasio,
9. Kedua variabel bersifat dependen, artinya satu variabel merupakan variabel bebas sedang variabel lainnya variabel terikat.

Sederhananya, pada metode regresi linear ini kita perlu membuat garis lurus pada titik data yang kita miliki, yang mana nantinya garis tersebut yang akan kita gunakan untuk melakukan prediksi terhadap titik data yang baru.

10.3 Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi Jupyter Notebook.
3. Modul Praktikum Data Warehousing dan Data Mining.
4. Dataset yang bisa diunduh di tautan berikut ini: https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM

10.4 Langkah-langkah Praktikum

Contoh Kasus:

Perusahaan asuransi kesehatan hanya dapat menghasilkan uang jika pemasukan yang didapatkan lebih banyak dibanding dengan biaya yang dikeluarkan untuk pembiayaan kesehatan pasien penerima

manfaat. Namun, memprediksi biaya kesehatan sangatlah sulit karena setiap individu memiliki karakteristik yang berbeda-beda. Tujuan dari eksperimen ini adalah untuk memprediksi biaya asuransi secara akurat berdasarkan data orang, termasuk usia, indeks masa tubuh, merokok atau tidak, dll. Selain itu, kita juga akan menentukan variabel terpenting yang mempengaruhi biaya asuransi. Prediksi ini akan bermanfaat bagi perusahaan asuransi untuk menentukan besaran premi yang harus dibayarkan per bulannya. Dari deskripsi di atas dapat kita simpulkan bahwa permasalahan kasus ini dapat diselesaikan dengan teknik regresi.

Dataset yang akan kita gunakan pada eksperimen ini tersedia di tautan yang tertera di atas dengan nama file *insurance.csv*. Dataset ini memiliki 1338 baris dan 7 kolom. Berikut deskripsi dari masing-masing kolom:

1. Usia: usia penerima manfaat asuransi
2. Jenis kelamin: jenis kelamin penerima manfaat asuransi
3. BMI : Indeks Massa Tubuh, memberikan pemahaman tentang berat badan yang relatif tinggi atau rendah relatif terhadap tinggi badan, indeks objektif berat badan (kg/m^2) menggunakan rasio tinggi terhadap berat, idealnya 18,5 hingga 24,9
4. Anak: jumlah anak yang ditanggung oleh asuransi kesehatan
5. Perokok: merokok atau tidak
6. Wilayah: daerah perumahan penerima di Amerika Serikat yang meliputi timur laut, tenggara, barat daya, barat laut.
7. Biaya: biaya pengobatan individu yang ditagih oleh asuransi kesehatan

Karena kita akan memprediksi biaya asuransi, maka yang akan menjadi target fitur (fitur yang akan diprediksi) adalah kolom biaya.

Memprediksi biaya asuransi berdasarkan fitur-fitur yang tersedia.

1. Pada Langkah pertama kita perlu memasukan beberapa library dan kelas yang akan kita gunakan pada eksperimen ini. Berikut beberapa library yang akan kita gunakan.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Selanjutnya, kita akan membaca dataset yang sudah kita unduh dengan menggunakan library pandas. Untuk membaca dataset dan melihat 5 data teratas dapat digunakan potongan kode di bawah ini.

```
df = pd.read_csv('insurance.csv')
df.head()
```

Sehingga akan ditampilkan lima data teratas dari dataset di atas seperti gambar berikut ini.

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

3. Selanjutnya kita juga bisa melihat gambaran dan ringkasan secara umum dari dataset yang kita gunakan dengan potongan kode berikut.

```
df.describe()
```

Sehingga akan ditampilkan seperti table berikut ini.

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

4. Sebelum kita bisa menggunakan dataset, kita perlu memastikan dulu bahwa tidak ada data yang null (kosong). Sehingga dapat dipastikan bahwa data konsisten dan tidak menyebabkan error. Untuk mengecek, kita bisa gunakan potongan kode berikut ini.

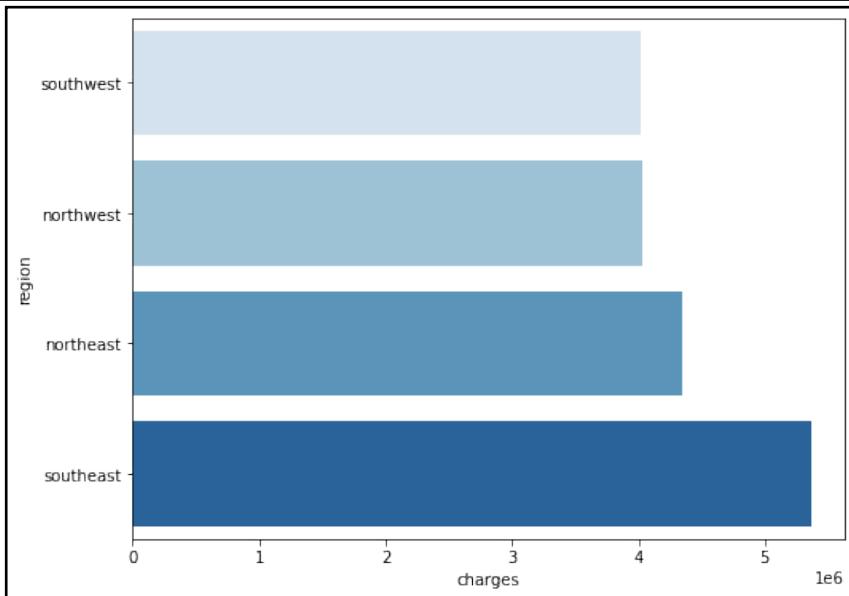
```
df.isnull().sum()
```

Sehingga kita akan dapatkan hasil seperti pada gambar berikut ini.

age	0
sex	0
bmi	0
children	0
smoker	0
region	0
charges	0
dtype:	int64

- Dapat kita lihat bahwa untuk masing-masing fitur tidak ada fitur yang nilainya 0. Sehingga dapat kita pastikan bahwa semua fitur ada nilainya.
5. Selanjutnya kita bisa melakukan beberapa visualisasi terhadap data yang ada untuk meningkatkan pemahaman kita tentang dataset yang akan digunakan dalam eksperimen. Pertama kita bisa lihat distribusi biaya kesehatan untuk masing-masing daerah dengan kode dibawah ini.

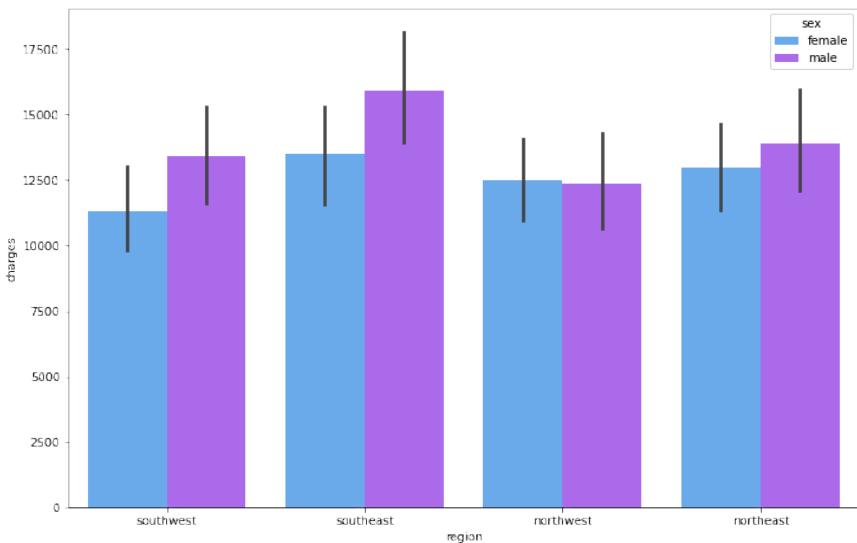
```
charges = df['charges'].groupby(df.region).sum().sort_values(ascending = True)
f, ax = plt.subplots(1, 1, figsize=(8, 6))
ax = sns.barplot(charges.head(), charges.head().index, palette='Blues')
```



- Jadi secara keseluruhan biaya pengobatan tertinggi ada di wilayah southeast dan terendah di southwest.
6. Selanjutnya kita juga bisa melakukan analisa terhadap data-data yang lain meliputi jenis kelamin, merokok atau tidak dan jumlah anak (khusus data yang dalam format kategorikal). Dalam hal ini kita akan menganalisa data tersebut berdasarkan wilayah yang berbeda. Sekarang kita mulai dengan jenis kelamin terlebih dahulu.

```
f, ax = plt.subplots(1, 1, figsize=(12, 8))
ax = sns.barplot(x='region', y='charges', hue='sex',
data=df, palette='cool')
```

Sehingga akan dihasilkan ilustrasi sebagai berikut ini.

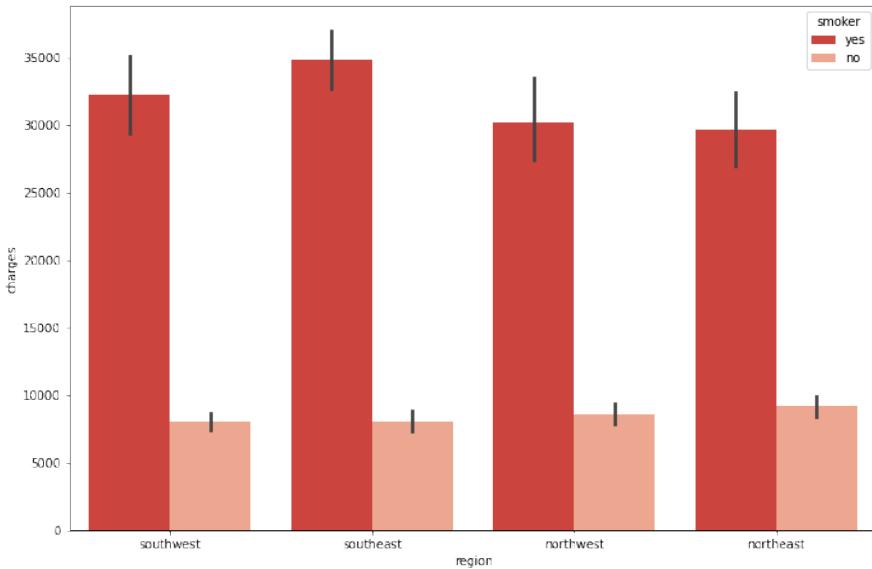


Dari gambar di atas dapat dilihat bahwa pasien laki-laki memiliki biaya lebih banyak di 3 dari 4 wilayah.

7. Selanjutnya kita akan melihat pengaruh variabel merokok atau tidak terhadap biaya kesehatan. Berikut potongan kode yang digunakan untuk memvisualisasikan pengaruh variabel merokok.

```
f, ax = plt.subplots(1, 1, figsize=(12, 8))
ax = sns.barplot(x = 'region', y = 'charges',
hue='smoker', data=df, palette='Reds_r')
```

Setelah menjalankan kode di atas akan ditampilkan diagram batang seperti berikut ini.

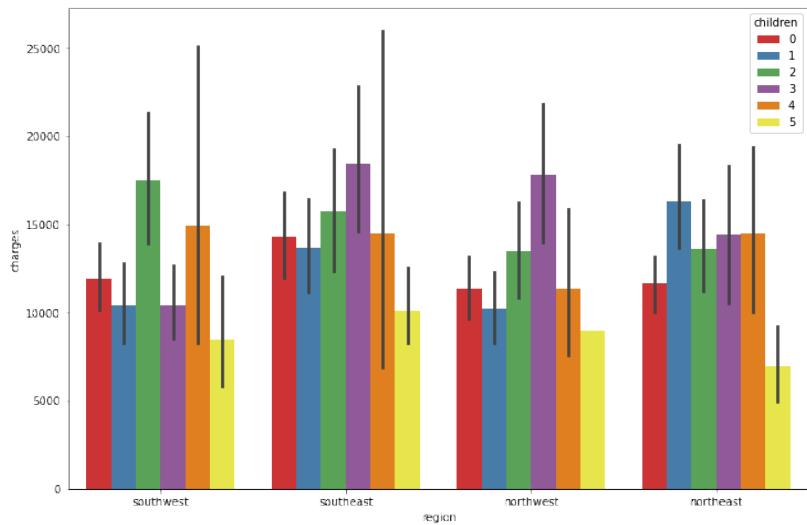


Dari gambar di atas dapat dilihat bahwa pasien merokok jauh menggunakan biaya kesehatan yang lebih tinggi dibandingkan yang tidak merokok di semua wilayah.

8. Terakhir kita akan melihat pengaruh jumlah anak yang dimiliki oleh pasien terhadap biaya kesehatan yang diklaim. Untuk melihat analisa tersebut bisa dengan menggunakan kode di bawah ini.

```
f, ax = plt.subplots(1, 1, figsize=(12, 8))
ax = sns.barplot(x='region', y='charges',
                  hue='children', data=df,
                  palette='Set1')
```

Dari menjalankan di atas akan dihasilkan diagram batang seperti gambar berikut ini.



Dari gambar di atas dapat disimpulkan juga bahwa orang dengan anak-anak cenderung memiliki biaya medis yang lebih tinggi secara keseluruhan.

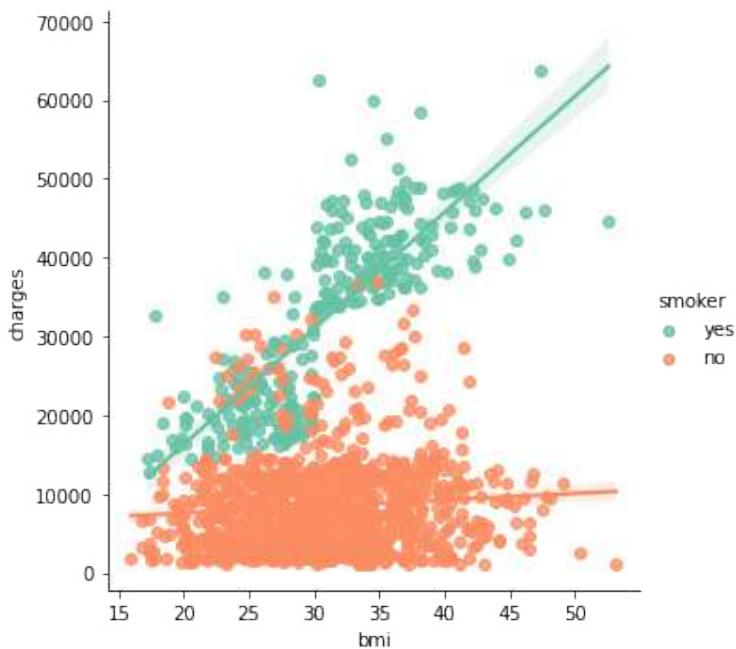
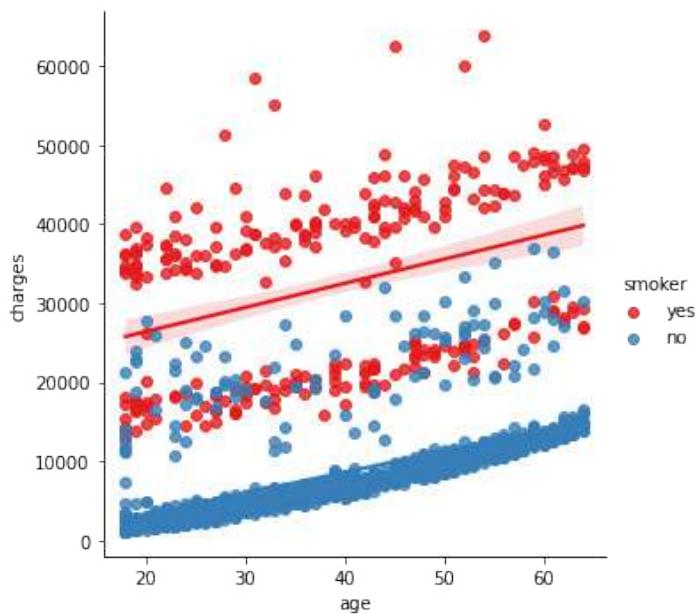
9. Dari sini kita sudah paham bahwa faktor merokok atau tidak merupakan variabel yang sangat penting dan berpengaruh terhadap biaya medis pasien. Untuk menganalisa dua variabel selanjutnya yaitu umur dan BMI, kita akan sekaligus menyertakan variabel merokok atau tidak. Untuk melakukan itu, silahkan gunakan kode dibawah ini.

```

ax = sns.lmplot(x = 'age', y = 'charges', data=df,
                 hue='smoker', palette='Set1')
ax = sns.lmplot(x = 'bmi', y = 'charges', data=df,
                 hue='smoker', palette='Set2')

```

Sehingga akan dihasilkan plot seperti pada gambar berikut ini.



Dari kedua gambar di atas kita bisa simpulkan bahwa merokok memiliki dampak tertinggi pada biaya medis, meskipun biayanya meningkat seiring bertambahnya usia dan BMI.

10. Setelah kita selesai untuk menganalisa data dengan melakukan visualisasi dan interaksi antar data, selanjutnya kita akan melakukan prediksi biaya medis suatu pasien dengan menggunakan algoritma regresi linear. Langkah pertama yang harus kita lakukan adalah mengubah data yang sifatnya label atau teks menjadi data kategorikal. Dari data di atas yang memiliki data label ada 3 yaitu jenis kelamin, merokok atau tidak, dan wilayah. Untuk mengubah dari teks menjadi kategorikal dapat dilakukan dengan kode di bawah ini.

```
df[['sex', 'smoker', 'region']] = df[['sex', 'smoker',  
'region']].astype('category')
```

Kode di atas akan mengubah data jenis kelamin, merokok atau tidak, dan wilayah menjadi data kategorikal.

11. Selanjutnya kita perlu mengubah data yang sudah diubah menjadi kategorikal pada langkah sebelumnya ke data numerikal. Untuk melakukan hal tersebut, kita bisa gunakan library LabelEncoder seperti potongan kode di bawah ini.

```
from sklearn.preprocessing import LabelEncoder  
label = LabelEncoder()  
label.fit(df.sex.drop_duplicates())  
df.sex = label.transform(df.sex)  
label.fit(df.smoker.drop_duplicates())  
df.smoker = label.transform(df.smoker)  
label.fit(df.region.drop_duplicates())  
df.region = label.transform(df.region)
```

12. Untuk data yang lain tidak perlu dilakukan perubahan karena sudah dalam format numerikal. Setelah semua data telah siap digunakan,

kita dapat menggunakannya untuk eksperimen prediksi biaya kesehatan. Seperti yang sudah disampaikan sebelumnya, bahwa kita akan menggunakan algoritma linear regression. Pertama kita perlu melakukan pemanggilan terhadap library yang akan digunakan untuk mengeksekusi algoritma linear regression dengan kode seperti berikut.

```
from sklearn.model_selection import train_test_split as  
holdout  
from sklearn.linear_model import LinearRegression  
from sklearn import metrics
```

Baris pertama kode di atas digunakan untuk membagi data yang kita miliki menjadi dua porsi yaitu data training dan data testing. Baris kedua digunakan untuk memanggil algoritma linear regression. Baris ketiga digunakan untuk mengevaluasi performa dari algoritma linear regression dalam melakukan prediksi.

13. Langkah selanjutnya, kita perlu mendefinisikan data training dan data testing yang akan kita gunakan dalam eksperimen. Untuk melakukan itu, gunakan kode berikut ini.

```
x = df.drop(['charges'], axis = 1)  
y = df['charges']  
x_train, x_test, y_train, y_test = holdout(x, y,  
test_size=0.2, random_state=0)
```

Variabel ‘x’ digunakan untuk menampung atribut yang digunakan dalam eksperimen, di mana semua data digunakan kecuali data ‘charges’. Kemudian variabel ‘y’ digunakan untuk menampung atribut target, yaitu atribut yang akan diprediksi, di mana yang digunakan sebagai atribut target adalah data ‘charges’. Kode baris ketiga digunakan untuk membagi data menjadi data training dan data testing, dimana 80% data akan digunakan sebagai data training dan 20% data digunakan sebagai data testing.

14. Selanjutnya kita bisa lakukan proses training (fitting) model linear regression kita dengan menggunakan data training yaitu x_train dan y_train. Untuk melakukan training bisa digunakan kode di bawah ini.

```
linear_reg = LinearRegression()  
linear_reg.fit(x_train, y_train)
```

Baris kode pertama digunakan untuk menginisialisasi model linear regression. Selanjutnya baris kedua digunakan untuk melakukan training dengan menggunakan data training, dimana x_train (berisi fitur yang akan dipelajari berisi data-data karakteristik pasien) akan di fitting dengan y_train (berisi target fitur berupa biaya kesehatan).

15. Setelah proses training, kita bisa melakukan testing dengan menggunakan data testing yang telah kita persiapkan. Pada proses testing ini, kita akan memprediksi data testing dengan menggunakan model linear regression yang telah kita train. Untuk melakukan testing dapat digunakan kode di bawah ini.

```
y_pred = linear_reg.predict(x_test)
```

Hasil prediksi akan ditampung di variabel 'y_pred'.

16. Selanjutnya sudah tentu kita ingin melakukan evaluasi terhadap performa prediksi model linear regression yang telah kita kembangkan. Untuk melakukan evaluasi regresi, metrik yang paling umum digunakan adalah R2 dan RMSE (root mean squared error). Gunakan kode berikut untuk mengevaluasi hasil prediksi.

```
R2 = metrics.r2_score(y_test, y_pred)  
rmse = (np.sqrt(metrics.mean_squared_error(y_test,  
y_pred)))  
print('R2 : {:.3f}'.format(R2))  
print('RMSE : {:.3f}'.format(rmse))
```

Setelah menjalankan baris kode di atas akan dihasilkan output sebagai berikut.

```
R2 : 0.800
RMSE : 5643.220
```

Jangan khawatir jika kalian mendapatkan hasil yang tidak sama persis, karena mungkin ketika pembagian data training dan data testing terdapat random variabel yang memungkinkan bagian data yang digunakan berbeda. Namun, seharusnya selisih hasil tidak berbeda jauh.

17. Selanjutnya kita juga masih bisa mencoba meningkatkan performa model linear regression dalam melakukan prediksi dengan mengeliminasi fitur yang tidak terlalu berpengaruh. Sebelum mengeliminasi fitur yang kurang penting, kita perlu mengetahui peranan masing-masing fitur dalam eksperimen ini. Untuk melihat peranan masing-masing fitur, gunakan kode di bawah ini.

```
importance = linear_reg.coef_
variables = ['age', 'sex', 'bmi', 'children', 'smoker',
'region']
for i,v in zip(variables,importance):
    print('Feature: %s, Score: %.5f' % (i,v))
```

Sehingga akan dihasilkan keluaran sebagai berikut.

```
Feature: age, Score: 253.99185
Feature: sex, Score: -24.32455
Feature: bmi, Score: 328.40262
Feature: children, Score: 443.72930
Feature: smoker, Score: 23568.87948
Feature: region, Score: -288.50857
```

18. Dari hasil analisa peranan masing-masing fitur dapat disimpulkan bahwa jenis kelamin dan wilayah memiliki peranan yang tidak signifikan dalam menentukan biaya medis pasien asuransi. Selanjutnya kita juga bisa menyimpulkan bahwa faktor merokok atau tidaknya pasien merupakan fitur paling penting untuk menentukan biaya medis pasien. Penemuan ini sesuai dengan analisa yang telah kita lakukan pada bagian awal eksperimen ini.

10.5 Tugas

Dikerjakan saat ini, jika tidak selesai bisa dilanjutkan di rumah.

Kasus:

Pada tugas ini, anda akan diberikan sebuah dataset tentang karakteristik beberapa rumah beserta dengan harganya. Tugas anda adalah memprediksi harga rumah dengan menggunakan algoritma regresi linear. Dataset dapat diunduh pada tautan yang tercantum bagian alat dan bahan dengan nama file Real estate.csv. Pada dataset ini terdapat beberapa kolom diantaranya:

1. transaction date
2. house age
3. distance to the nearest MRT station
4. number of convenience stores
5. latitude
6. longitude
7. house price of unit area

Dari data di atas data nomor 7 yang akan menjadi kelas target. Silahkan ikuti langkah-langkah praktikum di atas untuk mengerjakan tugas ini. Beberapa hal yang perlu kalian perhatikan adalah :

1. Silahkan kerjakan tugas ini dengan mengikuti semua langkah-langkah di atas. Untuk visualisasi dan analisa data (langkah 5 sampai langkah 9) bersifat opsional. Namun yang mengerjakan langkah 5 sampai 9 akan mendapatkan nilai yang maksimal.

2. Jika menurut anda ada fitur atau data yang tidak berpengaruh terhadap harga rumah, anda bisa mengeliminasi fitur tersebut. Kemudian bandingkan hasilnya dengan ketika menggunakan semua fitur.

Modul 11

Clustering: Algoritma K-Means

11.1 Tujuan

1. Mahasiswa mampu menggunakan algoritma k-means.
2. Mahasiswa mampu menerapkan algoritma k-means dalam kasus nyata.

11.2 Landasan Teori

Clustering adalah salah satu teknik analisis dan eksplorasi data yang paling umum digunakan untuk mendapatkan intuisi tentang sekumpulan data. Clustering juga dapat didefinisikan sebagai tugas mengidentifikasi subkelompok dalam data sedemikian rupa sehingga titik data dalam subkelompok yang sama (cluster) sangat mirip sedangkan titik data dalam kelompok yang berbeda sangat berbeda. Dengan kata lain, kita mencoba untuk menemukan subkelompok yang homogen dalam data sedemikian rupa sehingga titik data di setiap klaster semirip mungkin menurut ukuran kesamaan seperti euclidean distance atau correlation distance.

Analisis pengelompokan dapat dilakukan berdasarkan fitur atau karakteristik yang dimiliki oleh masing-masing titik data. Clustering dapat diaplikasikan ke beberapa permasalahan nyata di kehidupan sehari-hari termasuk membantu proses segmentasi pasar, dimana kita dapat mencari pelanggan yang mirip satu sama lain baik dari segi perilaku atau atribut. Kemudian juga bisa digunakan untuk segmentasi/kompresi citra (gambar), mengelompokkan wilayah yang serupa, mengelompokkan dokumen berdasarkan topik, dll. Tidak seperti supervised learning,

pengelompokan dianggap sebagai metode unsupervised learning karena dalam permasalahan ini kita tidak memiliki ground truth untuk dibandingkan dengan hasil output dari proses pengelompokan.

K-means clustering juga merupakan salah satu metode unsupervised yang paling sederhana dan populer. Biasanya, algoritma unsupervised membuat kesimpulan dari kumpulan data hanya menggunakan vektor input tanpa mengacu pada hasil yang diketahui, atau diberi label. Algoritma K-Means diperkenalkan oleh J.B. MacQueen pada tahun 1976, salah satu algoritma clustering sangat umum yang mengelompokkan data sesuai dengan karakteristik atau ciri-ciri bersama yang serupa. Grup data ini dinamakan sebagai *cluster*. Data di dalam suatu cluster mempunyai ciri-ciri (atau fitur, karakteristik, atribut, properti) serupa dan tidak serupa dengan data pada *cluster* lain.

K-means merupakan salah satu metode clustering non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*. Metode ini mempartisi data ke dalam *cluster* sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu *cluster* yang sama dan data yang mempunyai karakteristik yang berbeda di kelompokan ke dalam *cluster* yang lain. Secara umum algoritma dasar dari K-Means *Clustering* adalah sebagai berikut :

1. Tentukan jumlah *cluster*
2. Alokasikan data ke dalam *cluster* secara random
3. Hitung *centroid* / rata-rata dari data yang ada di masing-masing *cluster*
4. Alokasikan masing-masing data ke *centroid*/rata-rata terdekat
5. Kembali ke langkah 3, apabila masih ada data yang berpindah *cluster* atau apabila perubahan nilai *centroid*, ada yang di atas nilai *threshold* yang ditentukan atau apabila perubahan nilai pada *objective function* yang digunakan di atas nilai *threshold* yang ditentukan.

Distance space digunakan untuk menghitung jarak antara data dan *centroid*. Adapun persamaan yang dapat digunakan salah satunya yaitu *Euclidean Distance Space*. *Euclidean distance space* sering digunakan dalam

perhitungan jarak, hal ini dikarenakan hasil yang diperoleh merupakan jarak terpendek antara dua titik yang diperhitungkan. Adapun persamaannya adalah sebagai berikut :

$$d_{ij} = \sqrt{\sum_{k=1}^p \{x_{ik} - x_{jk}\}^2}$$

Keterangan:

d_{ij} = Jarak objek antara objek i dan j

p = Dimensi data

x_{ik} = Koordinat dari obyek i pada dimensi k

x_{jk} = Koordinat dari obyek j pada dimensi k

11.3 Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi Jupyter Notebook.
3. Modul Praktikum Data Warehousing dan Data Mining.
4. Dataset yang bisa diunduh di tautan berikut ini : https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM

11.4 Langkah-langkah Praktikum

Contoh Kasus:

Sekarang kita akan melihat contoh nyata dari pengelompokan k-means di mana kita akan membuat segmen pelanggan berdasarkan pendapatan tahunan mereka dan metrik yang akan kita sebut skor pengeluaran (dari 1 hingga 100). Dalam kumpulan data, pelanggan yang menghabiskan lebih banyak diberi skor lebih tinggi dibandingkan dengan pelanggan yang berbelanja lebih sedikit. Tujuan dari pengelompokan ini adalah mengidentifikasi pelanggan dengan pendapatan tinggi dan skor pengeluaran tinggi. Ini adalah pelanggan yang dapat ditargetkan lebih banyak dalam promosi dan kampanye pemasaran sehingga dapat lebih

memaksimalkan keuntungan. Pada eksperimen ini kita menggunakan dataset *Mall_Customers.csv* yang terdapat di tautan sesuai tertulis di bagian alat dan bahan.

1. Buka Jupyter Notebook yang tersedia di masing-masing komputer kalian.
2. Pada langkah pertama, kita perlu memanggil beberapa library yang akan kita gunakan untuk melakukan eksperimen ini seperti pada potongan kode berikut ini:

```
from sklearn.cluster import KMeans  
  
import numpy as np  
import pandas as pd  
  
import seaborn as sns  
import matplotlib.pyplot as plt  
sns.set_style("darkgrid")
```

Penjelasan :

- a. Kita perlu memanggil method KMeans karena kita akan menggunakan metode ini untuk melakukan clustering.
 - b. Kemudian pandas akan kita gunakan untuk membaca data dari dataset.
 - c. Terakhir, seaborn dan pyplot akan digunakan untuk memvisualisasikan data.
3. Langkah selanjutnya kita dapat melakukan pembacaan terhadap dataset yang akan kita gunakan. Untuk membaca/mengimpor dataset gunakan potongan kode berikut ini:

```
X = pd.read_csv('Mall_Customers.csv')
```

Untuk membaca file CSV kita dapat menggunakan method yang ada pada pandas.

4. Selanjutnya kita bisa lihat 5 data teratas untuk mendapat gambaran dari data yang akan kita cluster. Untuk melihat 5 data teratas bisa menggunakan potongan kode berikut:

```
x.head()
```

Hasilnya adalah sebagai berikut:

	CustomerID	Genre	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

5. Selanjutnya kita perlu untuk melakukan filter terhadap data yang ingin digunakan sebagai dasar pengelompokan. Sesuai yang sudah disampaikan sebelumnya, bahwa pengelompokan untuk data pemasaran ini berdasarkan annual income (pendapatan tahunan) dan spending score (skor pengeluaran). Sehingga untuk memilih kolom data yang digunakan dapat dengan menjalankan potongan kode berikut.

```
x = x.filter(["Annual Income (k$)", "Spending Score (1-100)"], axis = 1)
```

6. Langkah awal untuk melakukan pengelompokan dengan algoritma K-means adalah menentukan jumlah cluster. Sekarang kita akan

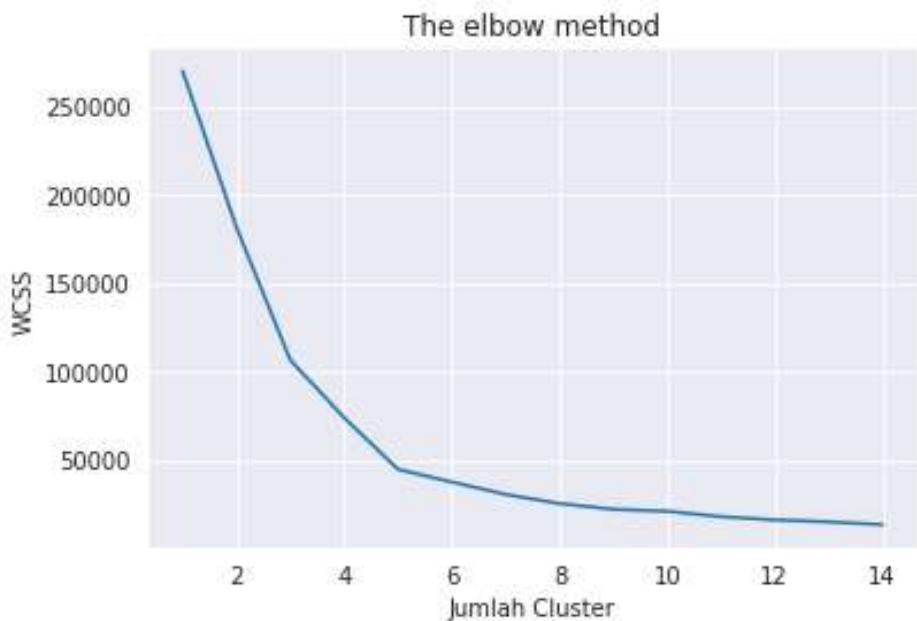
menerapkan Elbow Method pada dataset di atas. Elbow method ini memungkinkan kita untuk memilih jumlah cluster yang optimal untuk pengelompokan data. Untuk menerapkan Elbow method, silahkan gunakan potongan kode berikut ini.

```
wcss = []

for i in range(1, 15):
    kmeans = KMeans(n_clusters = i, init = 'k-means++',
max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

#Melakukan plot untuk hasil sehingga bisa melakukan
observasi terhadap elbow
plt.plot(range(1, 15), wcss)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') #WCSS = within cluster sum of squares
plt.show()
```

Hasil dari kode di atas adalah diagram berikut ini.

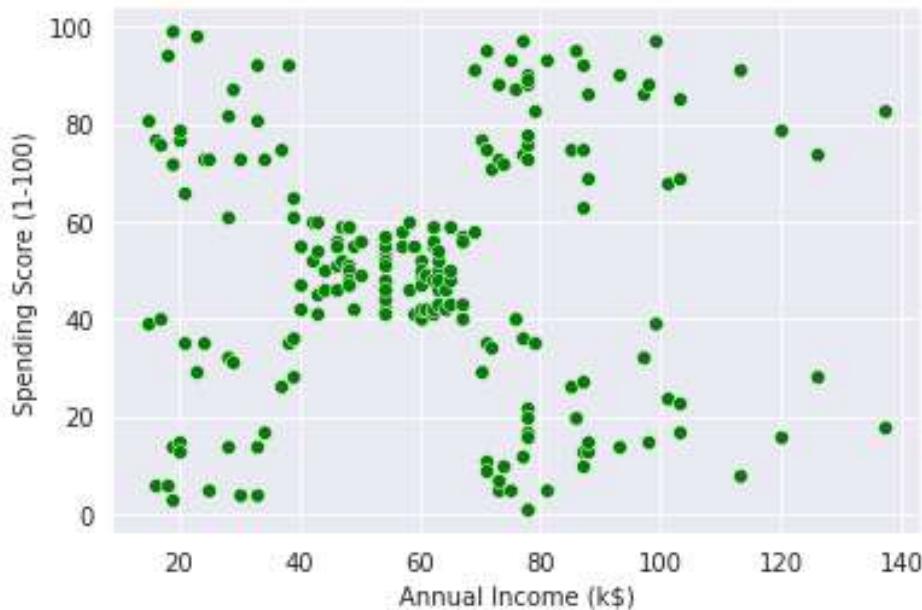


Kalian dapat melihat dengan jelas mengapa disebut elbow method dari grafik di atas. Jumlah cluster optimal adalah tempat dimana siku terjadi. Hal ini terjadi ketika WCSS (within cluster sum of squares) tidak berkurang secara signifikan pada iterasi berikutnya. Pada grafik di atas berarti jumlah cluster yang paling optimum adalah 5. Sekarang kita telah memiliki jumlah cluster yang optimal, selanjutnya kita bisa lakukan pengelompokan dengan algoritma K-means pada dataset penjualan di atas.

7. Selain dengan elbow method, kita juga bisa menganalisa kemungkinan jumlah cluster dengan melakukan visualisasi data dengan melakukan plot untuk masing-masing data point. Untuk melakukan plot kita bisa memanfaatkan library seaborn seperti potongan kode berikut ini.

```
sns.scatterplot(data = X, x="Annual Income (k$)", y="Spending Score (1-100)", c = ["green"])
```

Setelah menjalankan kode di atas akan didapatkan hasil plot seperti pada gambar di bawah ini.



Dari gambar visualisasi data di atas, kalian dapat melihat bahwa data pelanggan secara kasar dibagi menjadi 5 cluster, satu cluster di masing-masing dari empat sudut dan satu cluster di tengah. Sehingga hasil ini memverifikasi hasil dari elbow method sebelumnya.

8. Setelah kita mendapatkan jumlah cluster yang paling optimal, selanjutnya kita dapat melakukan pengelompokan data pelanggan dengan menggunakan algoritma K-means dengan menjalankan kode di bawah ini.

```
model = KMeans(n_clusters= 5)
model.fit(X)
```

Ketika proses clustering selesai dan berhasil, nantinya akan ditampilkan output seperti berikut.

```
KMeans(n_clusters=5)
```

- Setelah mengeksekusi kode di atas, data secara otomatis akan dikelompokkan menjadi lima kelompok. Kita bisa mengecek centroid untuk masing-masing kelompok data dengan kode di bawah ini.

```
print(model.cluster_centers_)
```

Dari eksekusi kode di atas akan ditampilkan titik data yang merupakan centroid untuk masing-masing kelompok yang dihasilkan dari algoritma K-means.

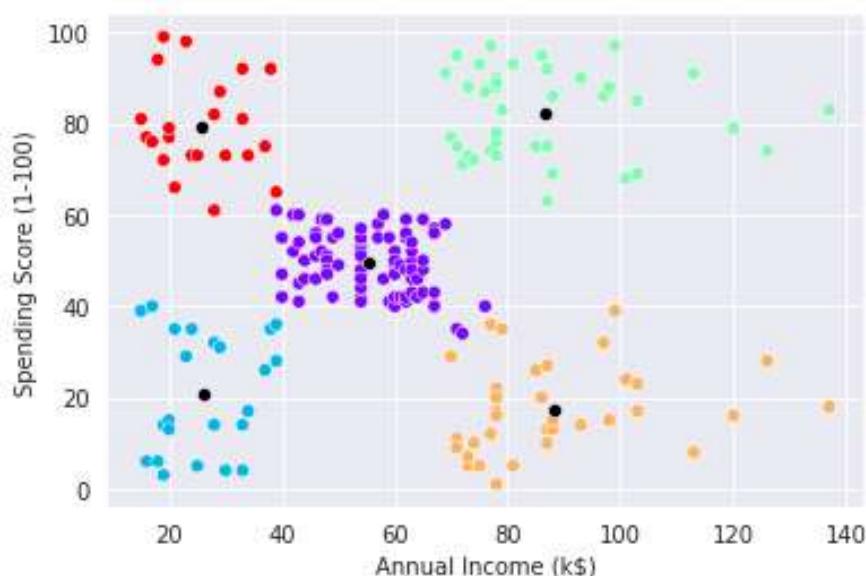
```
[[26.30434783 20.91304348]
 [55.2962963 49.51851852]
 [86.53846154 82.12820513]
 [25.72727273 79.36363636]
 [88.2 17.11428571]]
```

- Output di atas menunjukkan bahwa nilai centroid untuk kedua koordinat (skor pendapatan dan pengeluaran tahunan) paling tinggi untuk cluster ke-4 (pada indeks 3). Mungkin kelompok ini adalah segmen pelanggan yang akan ditargetkan dengan kampanye pemasaran karena mereka memiliki pendapatan tahunan tertinggi dan kemungkinan pengeluaran uang tertinggi. Selanjutnya kita juga bisa memvisualisasikan hasil pengelompokan data dengan warna yang berbeda untuk masing-masing cluster dengan menggunakan kode di bawah ini.

```
sns.scatterplot(data = X, x="Annual Income (k$)", y="Spending Score (1-100)", c= model.labels_, cmap='rainbow')

sns.scatterplot(x=model.cluster_centers_[:, 0], y=model.cluster_centers_[:, 1], c=['black'])
```

Kemudian hasil cluster yang di dapat adalah sebagai berikut ini.



11.5 Tugas

Dikerjakan saat ini, jika tidak selesai bisa dilanjutkan di rumah.

1. Spotify adalah layanan streaming musik, podcast, dan video digital yang memberi Anda akses ke jutaan lagu dan konten lain dari artis di seluruh dunia. Discover Weekly, Daily Mix, dan Spotify Wrapped tahunan adalah beberapa fitur yang juga memanfaatkan teknologi

sistem rekomendasi. Pernahkah anda mencari lagu dan akhirnya menemukan banyak lagu serupa yang Anda sukai dan simpan secara instan? Pada tugas kali ini kita akan melakukan eksperimen dengan Spotify dataset. Dataset yang digunakan adalah file *spotify_data.csv* yang dapat diperoleh di tautan tertera pada bagian alat dan bahan.

2. Setelah mengunduh dataset, lakukan pengelompokan data lagu dari spotify untuk mendapatkan playlist berdasarkan mood dari musik. Kalian bisa melakukannya dengan mengikuti langkah-langkah seperti yang dilakukan pada sesi praktikum sebelumnya. Dataset ini terdiri dari 17 kolom, namun yang akan digunakan digunakan sebagai fitur dalam pengelompokan hanya 14 kolom dari kolom 3 sampai kolom 16. Beberapa hal yang perlu diperhatikan :
 - a. Pada sesi praktikum, sebagai dasar penentuan cluster hanya digunakan dua fitur. Sedangkan pada akan digunakan 14 fitur yang berbeda.
 - b. Karena fitur yang digunakan lebih dari dua, untuk memvisualisasikan hasil pengelompokan, kalian perlu melakukan plot dalam ruang 3 dimensi (opsional untuk nilai yang lebih maksimal).

Modul 12

Induksi dan Aturan Asosiasi

12.1 Tujuan

1. Mahasiswa mampu menggunakan induksi aturan, dan aturan asosiasi.
2. Mahasiswa mampu menerapkan aturan induksi aturan, dan aturan asosiasi dalam kasus nyata.

12.2 Landasan Teori

Rule induction adalah salah satu teknik dalam data mining yang paling sering digunakan untuk menemukan pengetahuan dalam sistem *unsupervised learning*. *Rule* (aturan) adalah bentuk sederhana dari “jika ini maka ini dan ini dan kemudian ini”. Sebagai contoh: jika seseorang membeli roti maka orang tersebut juga cenderung untuk membeli selai.

Agar aturan-aturan tersebut bermanfaat maka harus ditambahkan dua informasi tambahan sesuai dengan keadaaan sebenarnya yaitu:

1. Keakuratan (*accuracy / confidence*) yang menunjukkan seberapa sering aturan tersebut benar.
2. Penerapan (*coverage / support*) yaitu angka yang menunjukkan seberapa sering aturan tersebut dipakai.

Association Rule merupakan suatu proses untuk menemukan semua aturan assosiatif yang memenuhi syarat minimum untuk *support* (*minsup*) dan syarat minimum untuk *confidence* (*minconf*) pada sebuah database.

Dalam menentukan suatu *Association Rule* umumnya terdapat dua ukuran kepercayaan (*interestingness measure*), yaitu *support* dan *confidence*. Kedua ukuran ini akan digunakan untuk *interesting association rules* dibandingkan dengan batasan yang telah ditentukan. Batasan inilah yang terdiri dari *minsup* dan *minconf*. *Association Rule Mining* adalah suatu prosedur untuk mencari hubungan antar item dalam suatu dataset. Dimulai dengan mencari frequent itemset, yaitu kombinasi yang paling sering terjadi dalam suatu itemset dan harus memenuhi minimum support.

Dalam tahap ini akan dicari kombinasi item yang memenuhi syarat minimum dari nilai support dalam database. Untuk mendapatkan nilai support untuk sebuah item A dapat diperoleh dari rumus berikut:

$$Support(A) = \frac{\text{Jumlah transaksi yang mengandung item } A}{\text{Total transaksi}}$$

Sementara itu, untuk mencari nilai support dari 2-item dapat diperoleh dari rumus berikut:

$$P(A \cap B) = \frac{\text{Jumlah transaksi yang mengandung } A \text{ dan } B}{\text{Total transaksi}}$$

Setelah semua *frequent item* dan *large itemset* ditemukan, dapat dicari semua *Association Rules* yang memenuhi syarat minimum untuk *confidence* (*minconf*) dengan menggunakan rumus berikut ini:

$$P(B|A) = \frac{\text{Jumlah transaksi yang mengandung } A \text{ dan } B}{\text{Jumlah transaksi yang mengandung item } A}$$

Kemudian, kita hitung nilai lift ratio dari aturan asosiasi yang telah dihasilkan dengan menggunakan rumus berikut ini:

$$Lift (A \rightarrow B) = \frac{P(B|A)}{P(B)}$$

12.3 Alat dan Bahan

1. Komputer dengan Python Environment.
2. Program aplikasi Jupyter Notebook.
3. Modul Praktikum Data Warehousing dan Data Mining.

12.4 Langkah-langkah Praktikum

12.4.1 Mengimport Library

Meng-import library yang diperlukan, yaitu library pandas, numpy, dan apriori. Untuk meng-import library yang akan digunakan, kita meng-import library pandas, numpy, dan apriori.

```
import numpy as np
import pandas as pd
from apyori import apriori
```

12.4.2 Membaca Dataset

Membaca dataset dari sebuah directory. Disini, kita mengambil dataset store_data.csv yang diambil dari platform gitea. Dataset tersebut akan tersimpan di variable store_data sebagai sebuah dataframe. Kemudian, kita akan mencetak tujuh data teratas pada dataset. Untuk link nya adalah sebagai berikut:

https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM/src/branch/master/Data/Bab12/store_data.csv

```
store_data = pd.read_csv("/Users/mbp/Documents/store_data.csv")
store_data.head(7)
```

Hasil:

	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	whole wheat pasta	french fries	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	soup	light cream	shallot	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

12.4.3 Mengkonversikan Dataframe ke dalam Array

Pada kode di bawah ini, kita akan mengkonversikan dataframe menjadi sebuah array bernama records. Disini, kita mengaplikasikan for loop dimulai dari baris pertama hingga terakhir. Kemudian di setiap iterasi tersebut, kita membuat sebuah list yang menyimpan nilai kolom pada setiap barisnya. Terakhir, kita cetak array records tersebut.

```
records = []
for i in range(0, store_data.shape[0]):
    records.append([str(store_data.values[i,j]) for j in range(0, store_data.shape[1])])

print(records)
```

Hasil:

12.4.4 Membuat Model Aturan Asosiasi

Selanjutnya, kita jalankan kode untuk aturan asosiasi menggunakan algoritma apriori dengan cara memanggil fungsi apriori. Adapun fungsi apriori memiliki beberapa parameter sebagai berikut:

1. Parameter pertama adalah array yang akan digunakan yaitu records.
 2. Parameter kedua adalah nilai minimum support ($\text{min_support}=0.0045$) yang berarti nilai minimum confidence yang diimplementasikan sebagai ambang batas confidence adalah 0.45%
 3. Parameter ketiga adalah minimum confidence ($\text{min_confidence}=0.2$) yang berarti nilai minimum confidence yang diimplementasikan sebagai ambang batas confidence adalah 20%
 4. Parameter keempat adalah minimum lift ratio ($\text{min_lift} = 3$) yang berarti nilai minimum lift ratio yang diimplementasikan sebagai ambang batas lift ratio adalah 3.
 5. Parameter kelima adalah $\text{min_length} = 2$ adalah jumlah minimum item yang kita inginkan dalam aturan asosiasi. Disini kita menggunakan $\text{min_length} = 2$ berarti kita menginginkan setidaknya dua produk dalam aturan asosiasi yang dihasilkan.

Hasil dari fungsi apriori disimpan ke dalam variabel `assocoation_rules`, kemudian `association_results` akan menyimpan `association_`

rules yang telah diubah menjadi sebuah list. Terakhir, kita cetak nilai association_results dengan menggunakan iterasi.

```
association_rules      = apriori(records,    min_support=0.0045,
                                    min_confidence=0.2, min_lift=3,
                                    min_length=2)
```

Hasil:

```
RelationRecord(items=frozenset({'chicken', 'light cream'}), support=0.00453333333333334, ordered_statistics=[OrderedStatistic(items_base=frozenset({'light cream'}), items_add=frozenset({'chicken'}), confidence=0.2905982905982906, lift=4.843304843304844)])  
RelationRecord(items=frozenset({'escalope', 'mushroom cream sauce'}), support=0.00573333333333333, ordered_statistics=[OrderedStatistic(items_base=frozenset({'mushroom cream sauce'}), items_add=frozenset({'escalope'}), confidence=0.30069930069930073, lift=3.7903273197390845)])  
RelationRecord(items=frozenset({'pasta', 'escalope'}), support=0.005866666666666667, ordered_statistics=[OrderedStatistic(items_base=frozenset({'pasta'}), items_add=frozenset({'escalope'}), confidence=0.37288135593220345, lift=4.700185158809287)])  
RelationRecord(items=frozenset({'ground beef', 'herb & pepper'}), support=0.016, ordered_statistics=[OrderedStatistic(items_base=frozenset({'herb & pepper'}), items_add=frozenset({'ground beef'}), confidence=0.3234501347708895, lift=3.2915549671393096)])  
RelationRecord(items=frozenset({'ground beef', 'tomato sauce'}), support=0.00533333333333333, ordered_statistics=[OrderedStatistic(items_base=frozenset({'tomato sauce'}), items_add=frozenset({'ground beef'}), confidence=0.37735849056603776, lift=3.840147461662528)])
```

12.4.5 Mencetak Rules, Support, Confidence, dan Lift Ratio

Kode berikut menampilkan aturan asosiasi, support, confidence, dan lift ratio untuk setiap aturan asosiasi:

```
for item in association_results:  
    pair = item[0]  
    items = [x for x in pair]  
  
    print("Rule: " + items[0] + " -> " + items[1])  
    print("Support: " + str(item[1]))  
    print("Confidence: " + str(item[2][0][2]))  
    print("Lift: " + str(item[2][0][3]))  
    print("====")
```

Hasil:

```
Rule: chicken -> light cream
Support: 0.004533333333333334
Confidence: 0.2905982905982906
Lift: 4.843304843304844
=====
Rule: escalope -> mushroom cream sauce
Support: 0.00573333333333333
Confidence: 0.30069930069930073
Lift: 3.7903273197390845
=====
Rule: pasta -> escalope
Support: 0.005866666666666667
Confidence: 0.37288135593220345
Lift: 4.700185158809287
=====
Rule: ground beef -> herb & pepper
Support: 0.016
Confidence: 0.3234501347708895
Lift: 3.2915549671393096
```

Nilai support untuk aturan pertama adalah 0.0045. Jumlah ini dihitung dengan membagi jumlah transaksi yang mengandung light cream dibagi dengan jumlah total transaksi. Tingkat confidence untuk aturan tersebut adalah 0.2905 yang menunjukkan bahwa dari semua transaksi yang mengandung light cream, 29.05% transaksi juga mengandung chicken. Terakhir, lift 4.84 berarti bahwa pembelian chicken 4.84 kali lebih mungkin dibeli oleh pelanggan yang membeli light cream dibandingkan dengan kemungkinan penjualan ayam secara umum.

12.5 Tugas

Terdapat dataset pada Grocery Store Dataset yang dapat diunduh pada link berikut: https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM/src/branch/master/Data/Bab12/GroceryStoreDataSet.csv

Pada dataset tersebut terdapat 19 data transaksi dengan daftar item sebagai berikut:

	products
0	MILK,BREAD,BISCUIT
1	BREAD,MILK,BISCUIT,CORNFLAKES
2	BREAD,TEA,BOURNVITA
3	JAM,MAGGI,BREAD,MILK
4	MAGGI,TEA,BISCUIT
5	BREAD,TEA,BOURNVITA
6	MAGGI,TEA,CORNFLAKES
7	MAGGI,BREAD,TEA,BISCUIT
8	JAM,MAGGI,BREAD,TEA
9	BREAD,MILK
10	COFFEE,COCK,BISCUIT,CORNFLAKES
11	COFFEE,COCK,BISCUIT,CORNFLAKES
12	COFFEE,SUGER,BOURNVITA
13	BREAD,COFFEE,COCK
14	BREAD,SUGER,BISCUIT
15	COFFEE,SUGER,CORNFLAKES
16	BREAD,SUGER,BOURNVITA
17	BREAD,COFFEE,SUGER
18	BREAD,COFFEE,SUGER
19	TEA,MILK,COFFEE,CORNFLAKES

Kemudian kerjakanlah soal-soal berikut ini:

1. Dengan ketentuan jumlah minimum support = 0.3% dan minumum confidence = 20%. Tuliskan hasil aturan asosiasi yang dihasilkan?
2. Dengan ketentuan jumlah minimum support = 0.7% dan minumum confidence = 60%. Tuliskan hasil aturan asosiasi yang dihasilkan?
3. Jelaskan bagaimana nilai minimum support dan minumum confidence memengaruhi aturan asosiasi dan nilai lift ratio yang dihasilkan.

Modul 13

Principal Component Analysis

13.1 Tujuan

1. Mahasiswa mampu menggunakan algoritma principal component analysis.
2. Mahasiswa mampu menerapkan algoritma principal component analysis dalam kasus nyata.

13.2 Landasan Teori

Pada modul ini kita akan memahami teknik yang disebut Principal Component Analysis (PCA). Teknik ini digunakan untuk mengurangi dimensi ketika kita memiliki terlalu banyak fitur input pada sebuah dataset. Dimensi ini diartikan sebagai sebuah fitur. Disini, kita akan memahami apa itu PCA dan cara kerjanya dengan contoh implementasi kode pada Python.

Mengapa Pengurangan Dimensi / Fitur perlu dilakukan? Hal ini berkaitan dengan istilah *Curse of Dimensionality*. Saat kita memiliki dataset dengan fitur input atau dimensi yang tinggi, program akan menghasilkan sebuah model yang kompeks. Untuk membuat model tersebut, perlu adanya *resource* yang besar dan waktu pemrosesan yang lama. Oleh karena itu, dibutuhkan sebuah teknik untuk mengurangi jumlah dimensi pada suatu dataset.

Misalkan, kita mencoba untuk menghapus fitur yang tidak relevan atau berlebihan karena tidak berkontribusi pada keakuratan masalah prediksi. Ketika kita membuang fitur tersebut, kita dapat kehilangan

informasi yang tersimpan di dalamnya. Maka dari itu, cara lain yang dapat kita implementasikan Adalah kita dapat membuat fitur independen baru dari fitur input yang ada. Dengan cara ini, kita tidak kehilangan informasi dalam fitur awal.

Sebagai gambarannya, jika kita ingin memprediksi penjualan sebuah toko retail untuk *item* tertentu. Fitur input yang digunakan untuk prediksi adalah angka penjualan, perubahan retail item, pergerakan inventaris, detail toko, retail pesaing, demografi pelanggan, dan informasi pelanggan seperti alamat, kode pos, dan lain sebagainya. Dari beberapa fitur input diatas, kita dapat mengeliminasi fitur tertentu seperti informasi pelanggan karena fitur tersebut tidak berkontribusi dalam memprediksi penjualan untuk toko ritel. Akan tetapi, ketika kita menghapus fitur ini, maka kita akan kehilangan informasi yang tersedia di fitur tersebut. Oleh karena itu, teknik reduksi dimensi yang cocok untuk kasus di atas adalah PCA, karena PCA dapat menyimpan informasi penting tanpa menghilangkan fitur.

Teknik PCA menyimpan informasi penting dalam kumpulan data tanpa menghilangkan fitur. Mekanisme PCA adalah membuat fitur independen baru dari fitur input yang ada. Pada PCA, fitur independen baru disebut dengan *principal components*. Sebagai contoh, saat kita memiliki dataset besar dari fitur input dan kita ingin mengurangi jumlah fitur input dengan tetap mempertahankan informasi penting. PCA mereduksi dimensi data menggunakan ekstraksi fitur. PCA merupakan salah satu teknik unsupervised learning karena hanya memproses fitur input tetapi tidak memproses fitur target.

Tujuan PCA adalah untuk mengurangi dimensi fitur input dari dataset dari n-dimensi ke dalam p-dimensi dimana $p < m$ dengan tetap mempertahankan semua informasi penting yang ada dalam data, namun dengan jumlah dimensi yang dikurangi.

Bagaimana PCA bekerja?

1. Langkah pertama dalam PCA adalah menentukan fitur input. Kita menggunakan dataset yang terdiri dari $n+1$ dimensi dan abaikan labelnya sehingga dataset baru kita menjadi n -dimensi.

2. Hitung rata-rata setiap dimensi dari keseluruhan dataset.
3. Hitung matriks kovarians dari seluruh dataset (kadang-kadang juga disebut sebagai matriks *variance-covariance*)

$$cov(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

4. Hitung nilai eigenvalue dan eigenvector

$$\det(A - \lambda I) = 0$$

5. Urutkan eigenvector dengan menurunkan eigenvalue dan pilih sebanyak k eigenvector dengan eigenvalue terbesar untuk membentuk matriks W berdimensi $d \times k$.
6. Ambil beberapa principal components sampel pertama, dengan nilai cumulative explained variance ratio minimal 90%.

13.3 Alat dan Bahan

1. Komputer dengan Python Environment.
2. Program aplikasi Jupyter Notebook.
3. Modul Praktikum Data Warehousing dan Data Mining.

13.4 Langkah-langkah Praktikum

Contoh Kasus:

Terdapat sebuah dataset tentang review dari sebuah platform travel TripAdvisor, yang berjumlah 980 reviews. Kemudian dataset tersebut terdapat 11 features sebagai berikut:

Atribut 1: Unique user id

- Merupakan atribut yang berisikan id user

Atribut 2: Average user feedback on art galleries

- Merupakan atribut yang berisikan nilai rata-rata feedback dari art gallery

Atribut 3: Average user feedback on dance clubs

- Merupakan atribut yang berisikan nilai rata-rata feedback dari dance club

Atribut 4: Average user feedback on juice bars

- Merupakan atribut yang berisikan nilai rata-rata feedback dari juice bars

Atribut 5: Average user feedback on restaurants

- Merupakan atribut yang berisikan nilai rata-rata feedback dari restoran

Atribut 6: Average user feedback on museums

- Merupakan atribut yang berisikan nilai rata-rata feedback dari museum

Atribut 7: Average user feedback on resorts

- Merupakan atribut yang berisikan nilai rata-rata feedback dari resorts

Atribut 8: Average user feedback on parks/picnic spots

- Merupakan atribut yang berisikan nilai rata-rata feedback dari taman dan tempat wisata

Atribut 9: Average user feedback on beaches

- Merupakan atribut yang berisikan nilai rata-rata feedback dari pantai

Atribut 10: Average user feedback on theaters

- Merupakan atribut yang berisikan nilai rata-rata feedback dari teater

Atribut 11: Average user feedback on religious institutions

- Merupakan atribut yang berisikan nilai rata-rata feedback dari lembaga keagamaan

Hipotesis:

Bagaimana mendapatkan dataset dengan dimensi yang lebih rendah dengan menggunakan algoritma PCA berdasarkan features yang telah diketahui pada dataset?

13.4.1 Mengimport Library

Meng-import library yang diperlukan, yaitu library pandas, numpy, dan PCA. Untuk meng-import library yang akan digunakan, kita meng-import library pandas, numpy, dan PCA.

```
import pandas as pd  
import numpy as np  
from sklearn.decomposition import PCA
```

13.4.2 Membaca Dataset

Membaca dataset dari sebuah directory. Disini, kita mengambil dataset tripadvisor.csv yang diambil dari platform gitea. Dataset tersebut akan tersimpan di variable data sebagai sebuah dataframe. Untuk link nya adalah sebagai berikut: https://gitea.ums.ac.id/yusufsn/Praktikum_DWDM/src/branch/master/Data/Bab13/tripadvisor.csv

```
data = pd.read_csv("/Users/mbp/Documents/tripadvisor.csv")  
print(data)
```

Hasil:

```
User ID  Category 1  Category 2  Category 3  Category 4  Category 5 \
0    User 1      0.93      1.80      2.29      0.62      0.80
1    User 2      1.02      2.20      2.66      0.64      1.42
2    User 3      1.22      0.80      0.54      0.53      0.24
3    User 4      0.45      1.80      0.29      0.57      0.46
4    User 5      0.51      1.20      1.18      0.57      1.54
...
975   User 976     0.74      1.12      0.30      0.53      0.88
976   User 977     1.25      0.92      1.12      0.38      0.78
977   User 978     0.61      1.32      0.67      0.43      1.30
978   User 979     0.93      0.20      0.13      0.43      0.30
979   User 980     0.93      0.56      1.13      0.51      1.34

Category 6  Category 7  Category 8  Category 9  Category 10
0          2.42      3.19      2.79      1.82      2.42
1          3.18      3.21      2.63      1.86      2.32
2          1.54      3.18      2.80      1.31      2.50
3          1.52      3.18      2.96      1.57      2.86
4          2.02      3.18      2.78      1.18      2.54
...
975       1.38      3.17      2.78      0.99      3.20
976       1.68      3.18      2.79      1.34      2.80
977       1.78      3.17      2.81      1.34      3.02
978       0.40      3.18      2.98      1.12      2.46
979       2.36      3.18      2.87      1.34      2.40

[980 rows x 11 columns]
```

13.4.3 Menghapus Kolom yang Tidak Diperlukan

Disini kita akan mengeliminasi feature yang tidak diperlukan, dalam hal ini adalah User ID, karena User ID tidak akan berpengaruh pada hasil clustering. Pada kode di bawah ini, kita mengaplikasikan fungsi drop pada variable data yang menyimpan data frame. Fungsi drop disini bertujuan untuk menghapus kolom User ID. Setelah User ID dihapus, dataframe disimpan ke dalam variable X. Kemudian variable X dicetak.

```
X = data.drop(columns=['User ID'])
print(X)
```

Hasil:

```
Category 1  Category 2  Category 3  Category 4  Category 5  Category 6  \
0          0.93       1.80      2.29       0.62       0.80      2.42
1          1.02       2.20      2.66       0.64      1.42      3.18
2          1.22       0.80      0.54       0.53       0.24      1.54
3          0.45       1.80      0.29       0.57       0.46      1.52
4          0.51       1.20      1.18       0.57      1.54      2.02
..         ...
975        0.74       1.12      0.30       0.53       0.88      1.38
976        1.25       0.92      1.12       0.38       0.78      1.68
977        0.61       1.32      0.67       0.43      1.30      1.78
978        0.93       0.20      0.13       0.43      0.30      0.40
979        0.93       0.56      1.13       0.51      1.34      2.36

Category 7  Category 8  Category 9  Category 10
0          3.19       2.79      1.82      2.42
1          3.21       2.63      1.86      2.32
2          3.18       2.80      1.31      2.50
3          3.18       2.96      1.57      2.86
4          3.18       2.78      1.18      2.54
..         ...
975        3.17       2.78      0.99      3.20
976        3.18       2.79      1.34      2.80
977        3.17       2.81      1.34      3.02
978        3.18       2.98      1.12      2.46
979        3.18       2.87      1.34      2.40
```

[980 rows x 10 columns]

13.4.4 Menampilkan Jumlah Fitur

Kode di bawah ini akan menampilkan jumlah fitur yang ada pada dataset setalah User ID dieliminasi. Fungsi shape menampilkan bentuk atau ordo dari sebuah dataset. Untuk mengetahui jumlah baris pada dataset X, kita menggunakan X.shape[0] sedangkan apabila kita ingin mengetahui jumlah kolom pada dataset X, kita menggunakan X.shape[1] pada kode Python tersebut. Kemudian jumlah kolom kita simpan pada variable num_features, yang akan kita cetak. Dalam kasus ini, kita memiliki 10 kolom/fitur.

```
num_features = X.shape[1]
print("Jumlah fitur input:", num_features, "fitur")
```

Hasil:

```
Jumlah fitur input: 10 fitur
```

13.4.5 Implementasi PCA dengan Jumlah Fitur Awal

Selanjutnya, kita jalankan kode untuk PCA dengan cara memanggil fungsi PCA. Adapun fungsi PCA memiliki parameter wajib yaitu: n_components yang berarti jumlah fitur yang ada pada dataset. Disini, dikarenakan jumlah fitur pada dataset setelah User ID dieliminasi adalah sejumlah 10 fitur, maka kita menggunakan n_components = 10 yang berarti kita akan membuat 10 *principal components* menggunakan teknik PCA.

```
pca = PCA(n_components=num_features)
pca.fit(X)
```

Hasil:

```
PCA(n_components=10)
```

13.4.6 Menampilkan Hasil Variance pada Tiap *Principal Components*

Kemudian, kita menampilkan hasil variance pada tiap principal components. Pada kode di bawah ini kita menerapkan fungsi enumerate untuk mendapatkan indeks dan data dari setiap elemen di dalam sebuah list. Elemen pada list pca.explained_variance_ratio_ indeksnya akan tersimpan dalam variable i sedangkan data akan tersimpan dalam variable j. Setelah itu, pada setiap iterasi, program akan mencetak nilai variance

untuk setiap *principal components*. Hasilnya, semakin rendah urutan *principal components*, nilai variance semakin tinggi.

```
for i,j in enumerate(pca.explained_variance_ratio_):
    print("Fitur independen ke-", (i+1), "menghasilkan variance
          ratio sebesar", round(j,7))
```

Hasil:

```
Fitur independen ke- 1 menghasilkan variance ratio sebesar 0.4252009
Fitur independen ke- 2 menghasilkan variance ratio sebesar 0.1772314
Fitur independen ke- 3 menghasilkan variance ratio sebesar 0.1245329
Fitur independen ke- 4 menghasilkan variance ratio sebesar 0.0731861
Fitur independen ke- 5 menghasilkan variance ratio sebesar 0.0693468
Fitur independen ke- 6 menghasilkan variance ratio sebesar 0.0538007
Fitur independen ke- 7 menghasilkan variance ratio sebesar 0.0412973
Fitur independen ke- 8 menghasilkan variance ratio sebesar 0.0258732
Fitur independen ke- 9 menghasilkan variance ratio sebesar 0.0095227
Fitur independen ke- 10 menghasilkan variance ratio sebesar 8e-06
```

13.4.7 Menampilkan Beberapa *Principal Component* Pertama Dengan Cumulative Explained Ratio Minimal 90%

Pada kode di bawah ini kita menambahkan kondisi dimana kita akan mengambil nilai variance kumulatif dari *principal components* jika sudah mencapai angka minimal 0.9. Disini, kita menambahkan variable cumulative_variance untuk menyimpan nilai variance kumulatif untuk *principal components*, juga variable num_pc untuk menyimpan jumlah *principal components* saat iterasi dijalankan.

Pada setiap iterasi, nilai num_pc bertambah satu dan nilai cumulative_variance bertambah sesuai dengan nilai variance pada setiap *principal components*; program akan mencetak nilai variance untuk setiap *principal components* untuk cumulative variance maksimal 0.9.

```

cummulative_variance = 0
num_pc = 0
for i,j in enumerate(pca.explained_variance_ratio_):
    if cummulative_variance < 0.9:
        num_pc += 1
        cummulative_variance += j
    print("Fitur independen ke-", (i+1), "menghasilkan
          variance ratio sebesar", round(j,7))

```

Hasil:

Fitur independen ke- 1 menghasilkan variance ratio sebesar 0.4252009
 Fitur independen ke- 2 menghasilkan variance ratio sebesar 0.1772314
 Fitur independen ke- 3 menghasilkan variance ratio sebesar 0.1245329
 Fitur independen ke- 4 menghasilkan variance ratio sebesar 0.0731861
 Fitur independen ke- 5 menghasilkan variance ratio sebesar 0.0693468
 Fitur independen ke- 6 menghasilkan variance ratio sebesar 0.0538007

13.4.8 Implementasi PCA dengan Jumlah Fitur yang Dikurangi

Selanjutnya, kita jalankan kode untuk PCA untuk mengurangi dimensi pada dataset. Disini kita akan menggunakan fungsi PCA dengan n_components sejumlah 6 dikarenakan jumlah yang dapat mewakili 90% dari dataset adalah sebanyak 6 fitur, maka kita menggunakan n_components = 6 yang berarti kita akan membuat 6 *principal components* menggunakan teknik PCA.

```

pca_reduced = PCA(n_components=num_pc)
pca_reduced.fit(X)

```

Hasil:

`PCA(n_components=6)`

13.5 Tugas

Terdapat dataset pada GPS Trajectories yang dapat diunduh pada halaman berikut: <http://archive.ics.uci.edu/ml/datasets/GPS+Trajectories>

Setelah itu, bukalah dataset **go_track_tracks.csv**.

Pada dataset tersebut terdapat 163 data dengan atribut sebagai berikut:

1. id: id dari objek
2. id_android: perangkat yang digunakan untuk membaca objek
3. speed: kecepatan rata-rata (km/h)
4. time: waktu tempuh perjalanan (h)
5. distance: jarak total (km)
6. rating: rating lalu lintas perjalanan. (3- baik, 2- normal, 1-buruk).
7. rating_bus: rating bus (1 - Penumpang bus sedikit, 2 - Penumpang Bus cukup banyak, 3- Penumpang Bus banyak.
8. rating_weather: rating cuaca (1- hujan, 2- cerah,).
9. mobil_atau_bus: (1 - mobil, 2 bus)
10. linha: informasi tentang bus yang melakukan jalur tersebut

Kemudian kerjakanlah soal-soal berikut ini:

1. Tentukan berapa jumlah fitur input yang digunakan untuk PCA. Bagaimana cara anda mendapatkan nilai tersebut?
2. Tuliskan algoritma PCA dengan n_components sebesar jumlah fitur input.
3. Setelah itu, tampilkan nilai variance ratio untuk setiap *principal components*. Kemudian, tentukan ada berapa fitur independen yang dapat memenuhi 90% cumulative variance ratio.
4. Cetaklah Data pada beberapa *principal components* pertama dengan Cumulative Explained Ratio Minimal 90%.

