

**MODEL PEMBELAJARAN DAN LAPORAN AKHIR
PROJECT-BASED LEARNING
MATA KULIAH DATA MINING II
KELAS C**



**"PENGEMBANGAN DATA WAREHOUSE UNTUK PEMANTAUAN CUACA DI KOTA
SURABAYA MENGGUNAKAN PENTAHO"**

DISUSUN OLEH KELOMPOK "IV":

- | | |
|---------------------------|-------------------------|
| 1. MUHAMAD HARIS HARTANTO | (21083010045) - KETUA |
| 2. ELLEXIA LEONIE GUNAWAN | (21083010027) - ANGGOTA |
| 3. ANGELA LISANTHONI | (21083010032) - ANGGOTA |
| 4. ALYA MIRZA SAFIRA | (21083010039) - ANGGOTA |
| 5. AMANDA AULIA | (21083010048) - ANGGOTA |

DOSEN PENGAMPU:

TRESNA MAULANA FAHRUDIN, S.ST., M.T. (NIP. 199305012022031007)
ANDRI FAUZAN ADZIIMA, S.Si., M.Si. (NIP. 199502122024061001)

PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
JAWA TIMUR
2024

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR	iii
DAFTAR TABEL	xi
BAB I PENGANTAR MYSQL, PENTAHO DATA INTEGRATION, DAN TABLEAU.....	1
1.1. Pentaho Data Integration	1
1.2. Tableau.....	4
1.3. MySQL.....	8
BAB II USULAN <i>DATA WAREHOUSE</i> MENGGUNAKAN MYSQL, PENTAHO DATA INTEGRATION, DAN TABLEAU.....	11
2.1. Variabel dan Sumber Data	11
2.2. Basis Data <i>Online Analytical Processing</i> (OLAP).....	12
2.3. Tujuan Penggunaan MySQL, Pentaho Data Integration, dan Tableau (Tableau Desktop)	15
2.3.1. MySQL 8.4.3 (Community).....	15
2.3.2. Pentaho Data Integration 9.4.0.0-343 (Developer/Community Edition).....	16
2.3.3. Tableau Desktop 2024.3.....	19
BAB III TAHAPAN PENGGUNAAN MYSQL, PENTAHO DATA INTEGRATION, DAN TABLEAU.....	21
3.1. Pengunduhan <i>Software Data Warehouse</i>	21
3.1.1. Pengunduhan Java Development Kit (JDK) 11	21
3.1.2. Pengunduhan MySQL.....	22
3.1.3. Pengunduhan Pentaho Data Integration 9.4.0.0-343 (Developer/Community Edition).....	25
3.1.4. Pengunduhan Tableau Desktop 2024.3	26
3.2. Instalasi <i>Software Data Warehouse</i>	28
3.2.1. Instalasi dan Konfigurasi Java Development Kit (JDK) 11.....	28
3.2.2. Instalasi MySQL.....	30
3.2.3. Instalasi Pentaho Data Integration (PDI) 9.4.0.0-343 (Developer/Community Edition)	42
3.2.4. Instalasi Tableau Desktop 2024.3	46
3.3. Proses <i>Extract, Transform, and Load</i> (ETL) pada Pentaho Data Integration (PDI)	49
3.4. Visualisasi Dasbor pada Tableau Desktop	53
BAB IV IMPLEMENTASI STUDI KASUS MENGGUNAKAN MYSQL, PENTAHO DATA INTEGRATION, DAN TABLEAU.....	54
4.1. Pengonfigurasian Koneksi Basis Data MySQL.....	54
4.1.1. DataGrip	54
4.1.2. Pentaho Data Integration	54
4.1.3. Tableau Desktop.....	56
4.2. Penyiapan Basis Data.....	57
4.2.1. Pembuatan Basis Data <i>Online Transaction Processing</i> (OLTP).....	57

4.2.2. Pembuatan Basis Data <i>Online Analytical Processing</i> (OLAP)	64
4.3. Ekstraksi Data dan Praproses Data	68
4.4. Transformasi Data	108
4.5. Visualisasi Dasbor	131
BAB V KESIMPULAN	143
DAFTAR PUSTAKA.....	144
LAMPIRAN	145
Lampiran 1. Video Dokumentasi Proyek	145
Lampiran 2. Berkas (<i>File</i>) Proyek (Data, Pentaho <i>Transformation</i> , dan Tableau <i>Workbook</i>)	145

DAFTAR GAMBAR

Gambar 1.1.	Proses ETL pada Pentaho Data Integration	1
Gambar 1.2.	Komponen Utama Pentaho Data Integration. (a) <i>Transformation</i> ; (b) <i>Job</i>	2
Gambar 1.3.	Proses Integrasi Data pada Pentaho Data Integration.....	3
Gambar 1.4.	Produk-produk Tableau.....	4
Gambar 2.1.	Diagram Skema/Basis Data <i>Online Analytical Processing</i> (OLAP).....	13
Gambar 3.1.	Halaman Web “Java SE 11 Archive Downloads” pada Situs Web Resmi Oracle	21
Gambar 3.2.	Opsi Berkas Instalasi Java SE Development Kit 11.0.24 pada Situs Web Resmi Oracle	22
Gambar 3.3.	Halaman web “MySQL Community Downloads” pada Situs Web Resmi MySQL (Developer Zone)	22
Gambar 3.4.	Opsi Pengunduhan Berkas Instalasi MySQL Community Server	23
Gambar 3.5.	Halaman Web “MySQL Product Archieves – MySQL Connector/J (Archived Versions)” pada Situs Web Resmi MySQL (Developer Zone)	23
Gambar 3.6.	Halaman Web “MySQL Product Archieves – MySQL Connector/ODBC (Archived Versions)” pada Situs Web Resmi MySQL (Developer Zone) ..	24
Gambar 3.7.	Halaman Web “Download Data Grip” pada Situs Web Resmi JetBrains... <td>25</td>	25
Gambar 3.8.	Halaman Web “Pentaho Developer Edition” pada Situs Web Resmi Pentaho	25
Gambar 3.9.	Daftar Lisensi Produk Pentaho Edisi Pengembang (<i>Developer</i>) atau Komunitas (<i>Community</i>).....	26
Gambar 3.10.	Daftar Produk Pentaho Edisi Pengembang (<i>Developer</i>) atau Komunitas (<i>Community</i>) 9.4	26
Gambar 3.11.	Halaman Web “PRODUCT DOWNLOADS – Tableau Desktop” pada Situs Web Resmi Tableau	27
Gambar 3.12.	Halaman Web “PRODUCT RELEASE AND DOWNLOAD - Tableau Desktop 2024.3” pada Situs Web Resmi Tableau	27
Gambar 3.13.	Tahapan Pertama dan Kedua Instalasi JDK 11	28
Gambar 3.14.	Tahapan Terakhir Instalasi JDK 11	28
Gambar 3.15.	Tahapan Pertama dan Kedua Pengonfigurasian JDK 11	29
Gambar 3.16.	Tahapan Ketiga Pengonfigurasian JDK 11	29
Gambar 3.17.	Tahapan Keempat dan Terakhir Pengonfigurasian JDK 11	30
Gambar 3.18.	Tahapan Pertama Instalasi MySQL Server 8.4.3 (<i>Community</i>)	31
Gambar 3.19.	Tahapan Kedua dan Ketiga Instalasi MySQL Server 8.4.3 (<i>Community</i>)..	31
Gambar 3.20.	Tahapan Keempat dan Kelima Instalasi MySQL Server 8.4.3 (<i>Community</i>).....	32
Gambar 3.21.	Tahapan Terakhir Instalasi MySQL Server 8.4.3 (<i>Community</i>).....	32
Gambar 3.22.	Tahapan Pertama Instalasi DataGrip 2024.3.....	33
Gambar 3.23.	Tahapan Kedua dan Ketiga Instalasi DataGrip 2024.3.....	33
Gambar 3.24.	Tahapan Keempat dan Kelima Instalasi DataGrip 2024.3	34
Gambar 3.25.	Tahapan Terakhir Instalasi DataGrip 2024.3	34

Gambar 3.26.	Opsi Aktivasi Lisensi DataGrip 2024.3.....	35
Gambar 3.27.	Informasi DataGrip 2024.3 dengan Lisensi yang Telah Aktif.....	35
Gambar 3.28.	Tahapan Pertama Instalasi <i>Driver ODBC MySQL 8.4.0</i>	36
Gambar 3.29.	Tahapan Kedua dan Ketiga Instalasi <i>Driver ODBC MySQL 8.4.0</i>	36
Gambar 3.30.	Tahapan Keempat dan Kelima Instalasi <i>Driver ODBC MySQL 8.4.0</i>	37
Gambar 3.31.	Tahapan Terakhir Instalasi <i>Driver ODBC MySQL 8.4.0</i>	37
Gambar 3.32.	Tahapan Pertama Pengonfigurasian MySQL Server 8.4.3 (<i>Community</i>)..	38
Gambar 3.33.	Tahapan Kedua dan Ketiga Pengonfigurasian MySQL Server 8.4.3 (<i>Community</i>).....	38
Gambar 3.34.	Tahapan Keempat dan Kelima Pengonfigurasian MySQL Server 8.4.3 (<i>Community</i>).....	39
Gambar 3.35.	Tahapan Keenam dan Ketujuh Pengonfigurasian MySQL Server 8.4.3 (<i>Community</i>).....	39
Gambar 3.36.	Tahapan Kedelapan dan Kesembilan Pengonfigurasian MySQL Server 8.4.3 (<i>Community</i>).....	40
Gambar 3.37.	Tahapan Terakhir Pengonfigurasian MySQL Server 8.4.3 (<i>Community</i>)..	41
Gambar 3.38.	Tahapan Pertama Pengonfigurasian <i>Environment Variable</i> untuk MySQL Server 8.4.3	41
Gambar 3.39.	Tahapan Kedua dan Terakhir Pengonfigurasian <i>Environment Variable</i> untuk MySQL Server 8.4.3	42
Gambar 3.40.	Tahapan Pertama Instalasi PDI 9.4.....	43
Gambar 3.41.	Tahapan Terakhir Instalasi PDI 9.4	43
Gambar 3.42.	Cara Mengakses PDI 9.4.....	44
Gambar 3.43.	Tampilan Awal PDI 9.4	44
Gambar 3.44.	Tahapan Pertama Pemasangan <i>Driver JDBC MySQL</i> pada PDI 9.4 (<i>Developer/Community Edition</i>)	45
Gambar 3.45.	Tahapan Kedua Pemasangan <i>Driver JDBC MySQL</i> pada PDI 9.4 (<i>Developer/Community Edition</i>)	45
Gambar 3.46.	Tahapan Terakhir Pemasangan <i>Driver JDBC MySQL</i> pada PDI 9.4 (<i>Developer/Community Edition</i>)	46
Gambar 3.47.	Tahapan Pertama Instalasi Tableau Desktop 2024.3	46
Gambar 3.48.	Tahapan Kedua, Ketiga, dan Terakhir Instalasi Tableau Desktop 2024.3	47
Gambar 3.49.	Opsi Aktivasi Lisensi Tableau Desktop 2024.3	48
Gambar 3.50.	Tahapan Aktivasi Lisensi Tableau Desktop 2024.3 Menggunakan Kode Aktivasi.....	48
Gambar 3.51.	Tampilan Awal Tableau Desktop 2024.3	49
Gambar 3.52.	Struktur Basis Data OLTP dalam Proyek.....	52
Gambar 3.53.	Struktur Basis Data OLTP dalam Proyek.....	53
Gambar 4.1.	Pengonfigurasian Koneksi Basis Data MySQL dalam DataGrip	54
Gambar 4.2.	Tahapan Pertama Pengonfigurasian Koneksi Basis Data MySQL dalam PDI.....	55
Gambar 4.3.	Tahapan Kedua Pengonfigurasian Koneksi Basis Data MySQL dalam PDI.....	55

Gambar 4.4.	Tahapan Ketiga Pengonfigurasian Koneksi Basis Data MySQL dalam PDI.....	56
Gambar 4.5.	Tahapan Terakhir Pengonfigurasian Koneksi Basis Data MySQL dalam PDI.....	56
Gambar 4.6.	Pengonfigurasian Koneksi Basis Data MySQL dalam Tableau Desktop ...	57
Gambar 4.7.	Tampilan Awal Tableau Desktop Setelah Terkoneksi dengan Basis Data MySQL.....	57
Gambar 4.8.	Kode SQL Inisialisasi Basis Data OLTP	63
Gambar 4.9.	Kode SQL Inisialisasi Basis Data OLAP	68
Gambar 4.10.	Rangkaian Step PDI untuk Ekstraksi Nama dan Tautan Wikidata Kecamatan	68
Gambar 4.11.	Konfigurasi Step “Generate Rows” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan.....	68
Gambar 4.12.	Konfigurasi Step “HTTP Client” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan.....	69
Gambar 4.13.	Konfigurasi Step “Split field to rows” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan.....	69
Gambar 4.14.	Konfigurasi Step “Filter rows 3” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan.....	70
Gambar 4.15.	Konfigurasi Step “Split field to rows 2” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan.....	70
Gambar 4.16.	Konfigurasi Step “Regex evaluation” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan	71
Gambar 4.17.	Konfigurasi Step “Filter rows” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan	72
Gambar 4.18.	Konfigurasi Step “Select Values” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan	72
Gambar 4.19.	Konfigurasi Step “Filter rows 4” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan	73
Gambar 4.20.	Konfigurasi Step “Calculator” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan	73
Gambar 4.21.	Konfigurasi Step “Select Values 4” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan	74
Gambar 4.22.	Potongan Data Hasil Ekstraksi Nama dan Tautan Wikidata Kecamatan .	74
Gambar 4.23.	Rangkaian Step PDI untuk Ekstraksi Tautan GNS Kecamatan	74
Gambar 4.24.	Konfigurasi Step “HTTP Client 2” dalam Ekstraksi Tautan GNS Kecamatan	75
Gambar 4.25.	Konfigurasi Step “Split field to rows 3” dalam Ekstraksi Tautan GNS Kecamatan	75
Gambar 4.26.	Konfigurasi Step “Filter rows 5” dalam Ekstraksi Tautan GNS Kecamatan	76
Gambar 4.27.	Konfigurasi Step “Split field to rows 4” dalam Ekstraksi Tautan GNS Kecamatan	76

Gambar 4.28.	Konfigurasi Step “Regex evaluation 3” dalam Ekstraksi Tautan GNS Kecamatan	76
Gambar 4.29.	Konfigurasi Step “Filter rows 6” dalam Ekstraksi Tautan GNS Kecamatan	77
Gambar 4.30.	Konfigurasi Step “Select value 5” dalam Ekstraksi Tautan GNS Kecamatan	77
Gambar 4.31.	Konfigurasi Step “Replace in string” dalam Ekstraksi Tautan GNS Kecamatan	77
Gambar 4.32.	Konfigurasi Step “Select Values 6” dalam Ekstraksi Tautan GNS Kecamatan	78
Gambar 4.33.	Potongan Data Hasil Ekstraksi Tautan GNS Kecamatan	78
Gambar 4.34.	Rangkaian Step PDI untuk Ekstraksi Koordinat <i>Latitude</i> dan <i>Longitude</i> Kecamatan	78
Gambar 4.35.	Konfigurasi Step “HTTP Client 3” dalam Ekstraksi Koordinat <i>Latitude</i> dan <i>Longitude</i> Kecamatan	79
Gambar 4.36.	Konfigurasi Step “Split field to rows 5” dalam Ekstraksi Koordinat <i>Latitude</i> dan <i>Longitude</i> Kecamatan	79
Gambar 4.37.	Konfigurasi Step “Filter rows 7” dalam Ekstraksi Koordinat <i>Latitude</i> dan <i>Longitude</i> Kecamatan	80
Gambar 4.38.	Konfigurasi Step “Split field to rows 6” dalam Ekstraksi Koordinat <i>Latitude</i> dan <i>Longitude</i> Kecamatan	80
Gambar 4.39.	Konfigurasi Step “Regex evaluation 4” dalam Ekstraksi Koordinat <i>Latitude</i> dan <i>Longitude</i> Kecamatan	81
Gambar 4.40.	Konfigurasi Step “Filter rows 2” dalam Ekstraksi Koordinat <i>Latitude</i> dan <i>Longitude</i> Kecamatan	81
Gambar 4.41.	Konfigurasi Step “Group by” dalam Ekstraksi Koordinat <i>Latitude</i> dan <i>Longitude</i> Kecamatan	82
Gambar 4.42.	Potongan Data Hasil Ekstraksi Koordinat <i>Latitude</i> dan <i>Longitude</i> Kecamatan	82
Gambar 4.43.	Rangkaian Step PDI untuk Ekstraksi Elevasi dan Kode Wilayah Kecamatan	83
Gambar 4.44.	Konfigurasi Step “Formula” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	83
Gambar 4.45.	Konfigurasi Step “HTTP Client 4” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	84
Gambar 4.46.	Konfigurasi Step “JSON Input” pada tab “File” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	84
Gambar 4.47.	Konfigurasi Step “JSON Input” pada tab “Content” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	85
Gambar 4.48.	Konfigurasi Step “JSON Input” pada tab “Fields” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	85
Gambar 4.49.	Konfigurasi Step “Select Values 2” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	85

Gambar 4.50.	Konfigurasi Step “CSV file input” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan.....	86
Gambar 4.51.	Bentuk Data File CSV Kode Wilayah di Indonesia (base.csv)	86
Gambar 4.52.	Konfigurasi Step “Filter rows 8” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	87
Gambar 4.53.	Konfigurasi Step “Calculator 2” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	87
Gambar 4.54.	Konfigurasi Step “Filter rows 9” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	87
Gambar 4.55.	Konfigurasi Step “Fuzzy match” pada tab “general” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	88
Gambar 4.56.	Konfigurasi Step “Fuzzy match” pada tab “general” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan	89
Gambar 4.57.	Konfigurasi Step “Regex evaluation 2” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan.....	89
Gambar 4.58.	Konfigurasi Step “Replace in string 2” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan.....	90
Gambar 4.59.	Konfigurasi Step “Select values 3” pada tab “Select & Alter” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan.....	90
Gambar 4.60.	Konfigurasi Step “Select values 3” pada tab “Select & Alter” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan.....	91
Gambar 4.61.	Konfigurasi Step “Insert/update” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan.....	91
Gambar 4.62.	Keseluruhan Rangkaian Step PDI untuk Ekstraksi Data Kecamatan	92
Gambar 4.63.	Potongan Data Hasil Ekstraksi Data Kecamatan	92
Gambar 4.64.	Hasil Ekstraksi Data Kecamatan dalam Basis Data OLTP	93
Gambar 4.65.	Rangkaian Step PDI untuk Ekstraksi Data Historis Cuaca.....	93
Gambar 4.66.	Konfigurasi Step “Table Input” dalam Ekstraksi Data Historis Cuaca.....	93
Gambar 4.67.	Konfigurasi Step “Metadata Structure of stream” dalam Ekstraksi Data Historis Cuaca.....	94
Gambar 4.68.	<i>Output</i> Step "Metadata Structure of stream" dalam Ekstraksi Data Historis Cuaca	94
Gambar 4.69.	Konfigurasi Step "Add Constants" dalam Ekstraksi Data Historis Cuaca	94
Gambar 4.70.	Konfigurasi Step "Formula" dalam Ekstraksi Data Historis Cuaca.....	95
Gambar 4.71.	Konfigurasi Step "HTTP client" dalam Ekstraksi Data Historis Cuaca.....	95
Gambar 4.72.	Konfigurasi Step "JSON input" pada tab “File” dalam Ekstraksi Data Historis Cuaca	96
Gambar 4.73.	Konfigurasi Step "JSON input" pada tab “Content” dalam Ekstraksi Data Historis Cuaca	96
Gambar 4.74.	Konfigurasi Step "JSON input" pada tab “Fields” dalam Ekstraksi Data Historis Cuaca	96
Gambar 4.75.	Konfigurasi Step "HTTP client" dalam Ekstraksi Data Historis Cuaca.....	97

Gambar 4.76.	Konfigurasi Step "JSON input" pada tab "File" dalam Ekstraksi Data Historis Cuaca.....	97
Gambar 4.77.	Konfigurasi Step "JSON input" pada tab "Content" dalam Ekstraksi Data Historis Cuaca.....	98
Gambar 4.78.	Konfigurasi Step "JSON input" pada tab "Fields" dalam Ekstraksi Data Historis Cuaca.....	98
Gambar 4.79.	Konfigurasi Step "Stream Lookup" dalam Ekstraksi Data Historis Cuaca.....	99
Gambar 4.80.	Konfigurasi Step "Select Values" dalam Ekstraksi Data Historis Cuaca ...	99
Gambar 4.81.	Konfigurasi Step "Switch/case" dalam Ekstraksi Data Historis Cuaca ..	100
Gambar 4.82.	Konfigurasi Step "Table Output" dalam Ekstraksi Data Historis Cuaca	100
Gambar 4.83.	Konfigurasi Step "Abort" dalam Ekstraksi Data Historis Cuaca.....	101
Gambar 4.84.	Potongan Hasil Ekstraksi Data Historis Cuaca dalam Basis Data OLTP	101
Gambar 4.85.	Rangkaian Step PDI untuk Ekstraksi Data Kode Cuaca	102
Gambar 4.86.	Konfigurasi Step "Generate rows" dalam Ekstraksi Data Kode Cuaca...	102
Gambar 4.87.	Konfigurasi Step "HTTP client" dalam Ekstraksi Data Kode Cuaca	102
Gambar 4.88.	Konfigurasi Step "Split field to rows" dalam Ekstraksi Data Kode Cuaca.....	103
Gambar 4.89.	Konfigurasi Step "Filter rows" dalam Ekstraksi Data Kode Cuaca	103
Gambar 4.90.	Konfigurasi Step "Split field to rows 2" dalam Ekstraksi Data Kode Cuaca	104
Gambar 4.91.	Konfigurasi Step "Modified JavaScript value" dalam Ekstraksi Data Kode Cuaca	104
Gambar 4.92.	Konfigurasi Step "Regex evaluation" dalam Ekstraksi Data Kode Cuaca.....	105
Gambar 4.93.	Konfigurasi Step "Filter rows 2" dalam Ekstraksi Data Kode Cuaca	106
Gambar 4.94.	Konfigurasi Step "Group by 2" dalam Ekstraksi Data Kode Cuaca	106
Gambar 4.95.	Konfigurasi Step "Insert/update" dalam Ekstraksi Data Kode Cuaca....	107
Gambar 4.96.	Potongan Hasil Ekstraksi Data Kode Cuaca dalam Basis Data OLTP	107
Gambar 4.97.	Hasil Ekstraksi Data Kota dan Provinsi dalam Basis Data OLTP	108
Gambar 4.98.	Rangkaian Step PDI untuk Mentransformasikan Data Tabel "dim_location".....	108
Gambar 4.99.	Konfigurasi Step "Table input" dalam Transformasi Data Tabel "dim_location".....	109
Gambar 4.100.	Konfigurasi Step "Table input 2" dalam Transformasi Data Tabel "dim_location".....	109
Gambar 4.101.	Konfigurasi Step "Get system info" dalam Transformasi Data Tabel "dim_location".....	110
Gambar 4.102.	Konfigurasi Step "Insert/update" dalam Transformasi Data Tabel "dim_location".....	111
Gambar 4.103.	Hasil Transformasi Data Tabel "dim_location" dalam Basis Data OLAP	112
Gambar 4.104.	Rangkaian Step PDI untuk Mentransformasikan Data Tabel "dim_rainfall"	112

Gambar 4.105. Konfigurasi Step "Data grid" pada tab "Meta" dalam Transformasi Data Tabel "dim_rainfall"	112
Gambar 4.106. Konfigurasi Step "Data grid" pada tab "Data" dalam Transformasi Data Tabel "dim_rainfall"	113
Gambar 4.107. Konfigurasi Step "Add sequence" dalam Transformasi Data Tabel "dim_rainfall"	113
Gambar 4.108. Konfigurasi Step "Get system info" dalam Transformasi Data Tabel "dim_rainfall"	114
Gambar 4.109. Konfigurasi Step "Insert/update" dalam Transformasi Data Tabel "dim_rainfall"	114
Gambar 4.110. Hasil Transformasi Data Tabel "dim_rainfall" dalam Basis Data OLAP .	115
Gambar 4.111. Rangkaian Step PDI untuk Mentransformasikan Data Tabel "dim_time".....	115
Gambar 4.112. Konfigurasi Step "Generate rows" dalam Transformasi Data Tabel "dim_time".....	115
Gambar 4.113. Konfigurasi Step "Add sequences" dalam Transformasi Data Tabel "dim_time".....	116
Gambar 4.114. Konfigurasi Step "Calculator" dalam Transformasi Data Tabel "dim_time"	116
Gambar 4.115. Konfigurasi Step "Modified JavaScript value" dalam Transformasi Data Tabel "dim_time"	117
Gambar 4.116. Konfigurasi Step "Select values" dalam Transformasi Data Tabel "dim_time".....	117
Gambar 4.117. Konfigurasi Step "Table output" pada tab "Main options" dalam Transformasi Data Tabel "dim_time"	118
Gambar 4.118. Konfigurasi Step "Table output" pada tab "Database fields" dalam Transformasi Data Tabel "dim_time"	119
Gambar 4.119. Potongan Hasil Transformasi Data Tabel "dim_time" dalam Basis Data OLAP.....	119
Gambar 4.120. Rangkaian Step PDI untuk Mentransformasikan Data Tabel "dim_weather_condition"	120
Gambar 4.121. Konfigurasi Step "Table input" dalam Transformasi Data Tabel "dim_weather_condition"	120
Gambar 4.122. Konfigurasi Step "Table input 2" dalam Transformasi Data Tabel "dim_weather_condition"	121
Gambar 4.123. Konfigurasi Step "Get system info" dalam Transformasi Data Tabel "dim_weather_condition"	121
Gambar 4.124. Konfigurasi Step "Insert/update" dalam Transformasi Data Tabel "dim_weather_condition"	122
Gambar 4.125. Potongan Hasil Transformasi Data Tabel "dim_weather_condition" dalam Basis Data OLAP	122
Gambar 4.126. Rangkaian Step PDI untuk Mentransformasikan Data Tabel "fact_weather"	123

Gambar 4.127. Konfigurasi Step "Table input" dalam Transformasi Data Tabel "fact_weather"	123
Gambar 4.128. Konfigurasi Step "Table input 2" dalam Transformasi Data Tabel "fact_weather"	124
Gambar 4.129. Konfigurasi Step "Select values" dalam Transformasi Data Tabel "fact_weather"	124
Gambar 4.130. Konfigurasi Step "Select values 2" dalam Transformasi Data Tabel "fact_weather"	125
Gambar 4.131. Konfigurasi Step "Database lookup" dalam Transformasi Data Tabel "fact_weather"	126
Gambar 4.132. Konfigurasi Step "Database lookup 2" dalam Transformasi Data Tabel "fact_weather"	127
Gambar 4.133. Konfigurasi Step "Database lookup 3" dalam Transformasi Data Tabel "fact_weather"	128
Gambar 4.134. Konfigurasi Step "Modified JavaScript value" dalam Transformasi Data Tabel "fact_weather"	128
Gambar 4.135. Konfigurasi Step "Get System info" dalam Transformasi Data Tabel "fact_weather"	129
Gambar 4.136. Konfigurasi Step "Select Values 3" dalam Transformasi Data Tabel "fact_weather"	129
Gambar 4.137. Konfigurasi Step "Insert/update" dalam Transformasi Data Tabel "fact_weather"	130
Gambar 4.138. Potongan Hasil Transformasi Data Tabel "fact_weather" dalam Basis Data OLAP.....	131
Gambar 4.139. Pembuatan Data Model dalam Tableau Desktop.....	131
Gambar 4.140.Bentuk Hubungan Antar Tabel dalam Basis Data OLAP pada Tableau Desktop	132
Gambar 4.141. <i>Worksheet</i> Statistik Deskriptif pada Tableau Desktop	132
Gambar 4.142. <i>Worksheet</i> Komparasi Statistik Deskriptif pada Tableau Desktop	133
Gambar 4.143. <i>Worksheet</i> Rata-rata Suhu pada Tableau Desktop	134
Gambar 4.144. <i>Worksheet</i> Rata-rata Kelembaban pada Tableau Desktop	135
Gambar 4.145. <i>Worksheet</i> Rata-rata Curah Hujan pada Tableau Desktop	136
Gambar 4.146. <i>Worksheet</i> Rata-rata Kecepatan Angin pada Tableau Desktop	137
Gambar 4.147. <i>Worksheet</i> Rata-rata Durasi Siang Hari pada Tableau Desktop	138
Gambar 4.148. <i>Worksheet</i> Proporsi Intensitas Hujan pada Tableau Desktop	139
Gambar 4.149. Dasbor Analisis Data Historis Cuaca Harian pada Tableau Desktop	140
Gambar 4.150. Opsi "Publish Workbook" pada Tableau Dekstop	142
Gambar 4.151. Jendela Pemilihan Server dalam opsi "Publish Workbook" pada Tableau Dekstop	142
Gambar 4.152. Jendela Inisialisasi Judul Publikasi <i>Workbook</i> dalam Tableau Public	142

DAFTAR TABEL

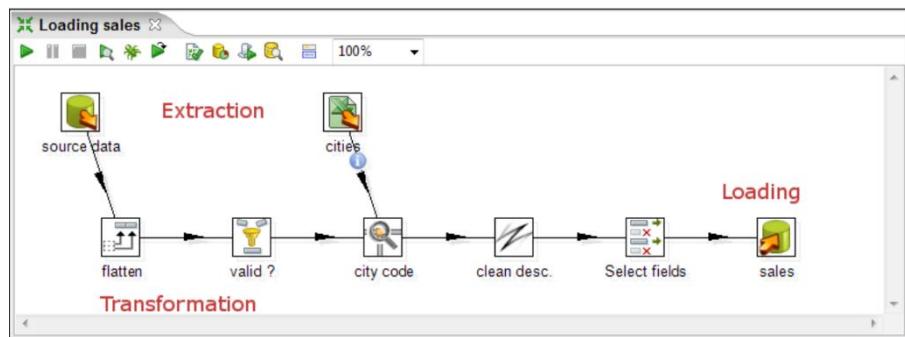
Tabel 2.1. Variabel Data	11
Tabel 2.2. Fitur MySQL yang Digunakan dalam Proyek.....	15
Tabel 2.3. Fitur Pentaho Data Integration (PDI) yang Digunakan dalam Proyek	17
Tabel 2.4. Fitur Tableau Desktop yang Digunakan dalam Proyek	19

BAB I

PENGANTAR MYSQL, PENTAHO DATA INTEGRATION, DAN TABLEAU

1.1. Pentaho Data Integration

Pentaho Data Integration (PDI) atau yang sering disebut dengan Kettle (*Extraction, Transformation, Transportation, and Loading Environment*) merupakan perangkat lunak (*software*) dari Pentaho yang dirancang untuk mendukung proses *Extraction, Transformation, and Loading* (ETL) (Pratama & Widhiasih, 2020). Pada April 2006, proyek Kettle diakuisisi oleh Pentaho Corporation dan *Matt Casters* sebagai pendirinya bergabung sebagai Arsitek Integrasi Data. Sejak akuisisi ini, PDI terus berkembang pesat dengan pembaruan rutin. Pada Juni 2006, PDI 2.3 rilis yang menambah dukungan multibahasa dan memperbaiki performa untuk *environment* besar. Kemudian pada Februari 2007, PDI 2.4 memperkenalkan dukungan eksekusi jarak jauh dan pengelompokan, serta penggabungan desain untuk *job* dan *transformation*. Kemudian, PDI 3.0 yang diluncurkan pada November 2007, menghadirkan desain ulang total dan peningkatan performa. Pada Juni 2010, PDI 4.0 dirilis dan memberikan perbaikan terutama pada fitur untuk *environment* perusahaan, seperti kontrol versi. Selanjutnya, PDI 4.4 pada November 2012 difokuskan pada *big data* yang memungkinkan pengelolaan *dataset* besar dan kompleks (Roldán, 2013). Hingga saat ini (per Agustus 2024), PDI telah mencapai versi 10.2 yang mencakup peningkatan fungsionalitas dan kinerja di beberapa komponen, seperti dukungan Java 17, peningkatan *license manager*, dukungan Cipher Block Chaining (CBC) untuk Advanced Encryption Standard (AES), peningkatan penjadwalan, parameter baru, dan sebagainya (Hitachi Vantara, 2024a).



Gambar 1.1. Proses ETL pada Pentaho Data Integration

PDI memiliki kemampuan dalam melakukan proses ETL yang diilustrasikan pada gambar 1.1, antara lain (Bouman & Dongen, 2009):

- *Extraction*

Proses yang mencakup pengambilan data dari satu atau lebih sistem sumber. Pada tahap ekstraksi, PDI menawarkan fitur *Change Data Capture* (CDC) yang memungkinkan hanya data yang berubah sejak ekstraksi terakhir yang diekstrak. Fitur ini berfungsi untuk memastikan bahwa data yang diperoleh selalu terbaru, dapat mengurangi beban dan meminimalkan waktu pemrosesan. Selain itu, PDI

menyediakan fitur *data staging* yang memungkinkan penyimpanan sementara data yang diekstrak dalam *staging area* sebelum diproses lebih lanjut.

- *Transformation*

Proses ini melibatkan perubahan bentuk dan format data agar sesuai dengan struktur target *data warehouse* yang diinginkan. PDI menyediakan berbagai fitur yang mendukung transformasi data, mulai dari *data validation* untuk memastikan keakuratan dan konsistensi data yang diterima dari sumber serta, *data cleansing* untuk memperbaiki data yang tidak valid atau tidak lengkap. Fitur *lookup* di PDI memungkinkan penggantian kode atau nilai numerik dengan nama yang lebih deskriptif. Selain itu, PDI juga menyediakan fitur *aggregation* yang memungkinkan pengguna untuk menghitung nilai agregat dari data mentah, serta *key generation* untuk menghasilkan kunci pengganti yang digunakan dalam pembuatan dimensi baru.

- *Loading*

Proses penyimpanan data ke dalam *data warehouse* yang dituju. Pada tahap *loading*, PDI mendukung pemuatan tabel fakta dan dimensi dengan efisien. Fitur seperti *Bulk Loading* memungkinkan pemuatan data dalam jumlah besar dengan cepat untuk mengurangi waktu untuk menyimpan data ke dalam *data warehouse*. PDI juga menyediakan kemampuan untuk memuat dan memperbarui data dengan menggunakan *Update* dan *Insert* steps, sehingga memungkinkan pembaruan data yang lebih mudah dan otomatis, serta memastikan bahwa metrik dan data yang disimpan sesuai dengan sumber data.



Gambar 1.2. Komponen Utama Pentaho Data Integration. (a) *Transformation*; (b) *Job*

Selanjutnya, gambar 1.2 menunjukkan komponen utama dalam PDI terdiri dari *transformation*, *step*, dan *job* yang memiliki kegunaan sebagai berikut (Casters dkk., 2010):

- *Transformation*

Merupakan Komponen utama dalam proses ETL. Fungsi utamanya adalah memanipulasi data atau baris data dalam konteks ekstraksi, transformasi, dan pemuatan. Transformasi terdiri dari satu atau lebih step yang melakukan pekerjaan inti ETL, seperti membaca data dari berkas (*file*), menyaring baris, membersihkan data, atau memuat data ke dalam basis data (*database*). *Step* dalam transformasi ini dihubungkan oleh *hop* yang mendefinisikan jalur satu arah untuk aliran data antar *step*.

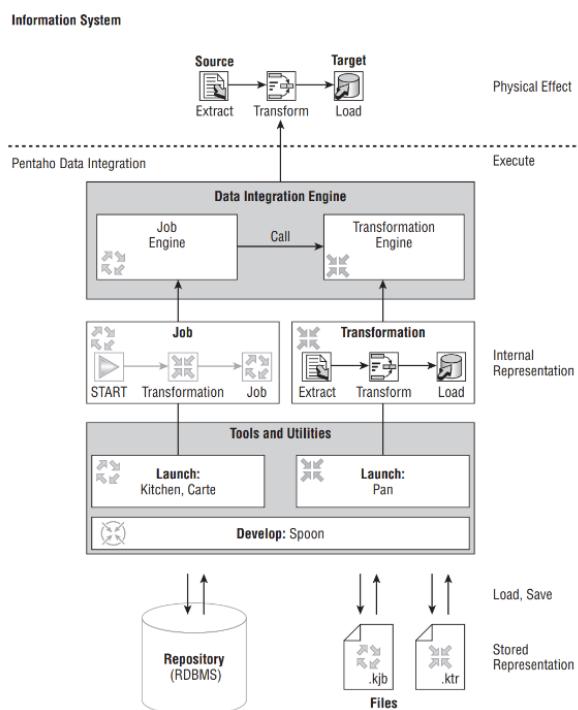
- *Step*

Merupakan unit dasar yang digunakan untuk memproses data dalam *transformation*. Setiap langkah digambarkan secara grafis melalui ikon, seperti “Table input” untuk

membaca data dari basis data dan “Text file output” untuk menulis data ke berkas teks. Setiap *step* membutuhkan nama yang unik dalam satu transformasi dan dapat melakukan operasi pembacaan (*read*) serta penulisan (*write*) data. *Step* mengirimkan data ke *hop out* yang menghubungkannya dengan *step* berikutnya. Proses ini dijalankan secara paralel dan data terus mengalir di antara *step* yang saling terhubung.

- *Job*

Merupakan komponen yang digunakan untuk mengelola urutan eksekusi tugas-tugas pemeliharaan dalam proyek ETL. Tugas ini mencakup proses seperti pemverifikasian keberadaan tabel basis data atau menentukan urutan transfer berkas. Komponen ini diperlukan untuk mengelola dan mengatur urutan eksekusi tugas-tugas *transformation* yang dijalankan secara paralel. *Job* terdiri dari satu atau lebih *job entry* yang dieksekusi secara berurutan dengan urutan yang ditentukan oleh *job hop* antar *job entry* serta hasil eksekusi dari masing-masing *job entry* tersebut.



Gambar 1.3. Proses Integrasi Data pada Pentaho Data Integration

Selain itu, PDI juga memiliki berbagai alat dan utilitas yang mendukung proses integrasi data yang diilustrasikan pada gambar 1.3, antara lain (Bouman & Dongen, 2009):

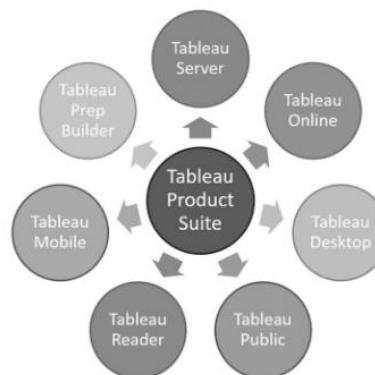
- *Spoon*: *Integrated Development Environment* (IDE) yang dirancang untuk menyederhanakan pembuatan *transformation* dan *job*.
- *Kitchen*: Alat (*tool*) baris perintah yang berfungsi untuk menjalankan *job*.
- *Pan*: Utilitas lain yang digunakan untuk menjalankan *transformation* melalui antarmuka baris perintah seperti untuk membaca, memproses, dan menyimpan data.
- *Carte*: Server ringan yang memungkinkan pelaksanaan *job* dan *transformation* pada *host* jarak jauh yang memberikan fleksibilitas dalam pemrosesan data terdistribusi.

Terakhir, PDI menyediakan dua jenis edisi layanan, yaitu edisi pengembang atau komunitas (*developer/community edition*) yang dapat diunduh secara gratis. Edisi ini dirancang khusus untuk eksplorasi dan pengembangan dalam lingkungan non-produksi. Sebaliknya, terdapat *enterprise edition* yang merupakan versi berbayar dengan opsi paket layanan mulai dari “starter”, “pro”, hingga “pro suite” dengan fitur-fitur yang lebih komprehensif dan dirancang untuk lingkungan produksi. Informasi lebih lanjut mengenai PDI, dapat dilihat pada dokumentasi penggunaannya yang dapat dapat diakses pada situs web Dokumentasi Hitachi Vantara melalui tautan <https://docs.hitachivantara.com/>.

1.2. Tableau

Tableau adalah *tools* atau platform yang digunakan untuk membuat visualisasi data yang lebih interaktif, mudah dibaca, dan mudah dilakukan proses analisis. Visualisasi yang dilakukan, yaitu dengan mengubah data dalam format tabel yang cenderung statis menjadi visualisasi yang dinamis dalam bentuk grafik, diagram, pemetaan geografis (*geomapping*), dan berbagai jenis visualisasi lainnya. Transformasi tersebut memungkinkan perbedaan dan tren dalam data menjadi lebih jelas sehingga memudahkan dalam interpretasi dan pengambilan keputusan. Hasil analisis data statistik yang sebelumnya bersifat konvensional dapat disajikan dalam bentuk visual yang estetis dengan memanfaatkan dasbor (*dashboard*) interaktif yang telah disediakan oleh Tableau (Saepuloh, 2020).

Sementara itu, Tableau Software adalah penyedia perangkat lunak visualisasi data interaktif yang berbasis di Amerika Serikat dan berspesialisasi dalam pengelolaan informasi korporat. Saat ini, kantor pusatnya terletak di Seattle, Washington. Tableau awalnya didirikan oleh Pat Hanrahan, Christian Chabot, dan Chris Stolte di Mountain View, California pada tahun 2003 dan diakuisisi oleh *Salesforce* pada tahun 2019. Tableau dikenal sebagai salah satu alat visualisasi data yang paling kuat dan berkembang pesat dalam ruang *business intelligence* (Mittal & Raheja, 2024).



Gambar 1.4. Produk-produk Tableau

Tableau memiliki berbagai macam produk yang dikelompokkan berdasarkan fungsionalitasnya. Kategori tersebut diantaranya adalah *data preparation tools*, *viewer tools*, *server tools*, dan *developer tools*. Berikut ini merupakan penjelasan dari masing-masing kategori (Mittal & Raheja, 2024):

a. *Tableau Server Tools*

- **Tableau Server**

Merupakan aplikasi server berbayar yang harus diinstal pada server berbasis Windows atau Linux. Produk ini banyak digunakan dalam berbagai sektor bisnis. Pengelolaan server seperti manajemen memori, pengguna (*user*), sumber data, direktori, dan tugas lainnya dilakukan oleh individu atau tim khusus yang bertugas sebagai administrator.

- **Tableau Online**

Tableau Online tidak memerlukan instalasi pada server Windows seperti Tableau Server karena telah *di-hosting* oleh Tableau Software sebagai program server berbayar berbasis *cloud*. Produk ini lebih disukai karena akses ke dasbor Tableau dapat dilakukan dari lokasi manapun. Sebagian besar fungsi Tableau Server juga tersedia dalam Tableau Online dengan perbedaan utama yang terletak pada keberadaannya yang berbasis daring (*online*). Selain itu, Tableau ini dan juga Tableau Server mendukung *data streaming* dari aplikasi web seperti Google Analytic dan Salesforce.com.

b. *Tableau Developer Tools*

- **Tableau Desktop**

Tableau Desktop atau yang juga dikenal sebagai Tableau Desktop Professional Edition adalah program desktop premium dan canggih untuk analisis data. Analis data dapat membuat berbagai jenis grafik, perhitungan menggunakan rumus, dasbor menarik, dan fitur canggih lainnya. Meskipun program ini berbayar terdapat periode uji coba yang tersedia selama 14 hari. Dasbor yang dibuat di program ini dapat dipublikasikan ke server Tableau yang telah disebutkan sebelumnya dan dapat diakses dengan mudah menggunakan Tableau Reader.

- **Tableau Public**

Tableau Public merupakan versi Tableau Desktop yang dikembangkan dengan mempertimbangkan efisiensi biaya. *Workbook* yang dikerjakan dalam program ini akan ditandai sebagai “Public” dan tidak dapat dilakukan penyimpanan secara lokal. Sebaliknya, penyimpanan *workbook* harus dilakukan secara daring di *cloud* yang disediakan oleh Tableau dan dapat diakses secara publik. Tableau versi ini tidak memiliki fitur untuk mengelola pengguna atau folder, serta hanya mendukung pembacaan data dengan sumber data berupa berkas yang disimpan secara lokal. Oleh karena itu, Tableau Public ideal bagi pengguna yang ingin mempelajari analisis data menggunakan Tableau dengan berbagai data publik.

c. *Tableau Viewer Tools*

- **Tableau Reader**

Tableau Reader adalah program gratis berbasis desktop yang memungkinkan pengguna membaca dasbor Tableau yang dibuat menggunakan Tableau Desktop. Tableau Reader hanya mendukung pembacaan data dari berkas yang tersimpan secara lokal. Melalui pembelian satu lisensi Tableau Desktop dan menginstal Tableau Reader sebagai program gratis pada komputer, pengguna dapat dengan mudah merasakan pengalaman menggunakan Tableau. Tableau Reader

memungkinkan akses ke dasbor dan mendukung operasi seperti *filter*, *tooltip*, dan parameter, tetapi tidak dapat mengubah grafik atau formula yang terdapat dalam dasbor.

- **Tableau Mobile**

Tableau Mobile adalah aplikasi untuk perangkat seluler yang tersedia di iOS dan Android. Aplikasi ini berfungsi seperti Tableau Reader, tetapi juga dapat terhubung ke Tableau Online atau Tableau Server. Dasbor interaktif yang dipublikasikan ke Tableau Server atau Tableau Online dapat diakses dan dilihat pada perangkat seluler melalui aplikasi ini.

d. *Data Preparation Tools*

- **Tableau Prep Builder**

Tableau Prep Builder adalah rilisan terbaru dari Tableau yang dirancang untuk membantu pengembang dalam menggabungkan, membersihkan, dan mengelola data mentah. Program ini merupakan perangkat *Extraction, Transformation, and Load* (ETL) berskala kecil yang mampu terhubung ke berbagai basis data. Tableau Prep Builder dapat menggabungkan data, mengubah data, dan menghasilkan berkas *output*. Selain itu, proses tersebut dapat diotomatisasi sehingga membantu pengembang (*developer*) mengurangi proses pengolahan data manual yang tidak perlu.

Saat ini (per Desember 2024), rilisan Tableau terbaru (versi 2024.3) menghadirkan sejumlah fitur inovatif yang dirancang untuk meningkatkan kemampuan analisis data dan mendukung kebutuhan pengguna secara lebih komprehensif. Berikut adalah fitur-fitur terbaru yang tersedia (Salesforce, 2024b):

1. **Tableau Cloud Manager**

Fitur ini memungkinkan administrator Tableau untuk dengan mudah mengelola dan memperluas analisis data di lingkungan *cloud*. Tableau Cloud Manager menyederhanakan proses ekspansi dan pengelolaan ekosistem analitik *cloud* sambil memastikan kepatuhan terhadap persyaratan residensi data serta mempertahankan tata kelola (*governance*) yang baik.

2. **Table Viz Extension**

Fitur ini memberikan kemampuan untuk beralih dari pelaporan tradisional berbasis teks ke analitik visual yang lebih modern dengan tetap memenuhi kebutuhan pengguna yang memerlukan pelaporan konvensional. Tabel tetap menjadi visualisasi penting bagi analis terutama untuk memeriksa data dalam format teks dan menyampaikan detail. Melalui Table Viz Extension, pengguna dapat menambahkan tabel detail dan tampilan *grid* ke dasbor yang dirancang untuk kenyamanan pengguna dengan format laporan tradisional.

3. **Spatial Parameters**

Fitur ini memperluas dukungan Tableau untuk data geospasial yang memberikan pengalaman Tableau yang lebih lengkap dalam analisis berbasis lokasi. Spatial Parameters memungkinkan pengguna untuk membuat dasbor interaktif dan dapat menjawab berbagai pertanyaan terkait data geospasial. Fitur ini memungkinkan pengguna untuk memilih titik-titik dalam data spasial yang kemudian digunakan

sebagai *input* dalam kalkulasi untuk menciptakan perilaku dinamis berdasarkan pilihan tersebut.

4. Tableau Cloud App untuk Microsoft Teams

Fitur ini berguna untuk integrasi Tableau dengan Microsoft Teams melalui Tableau Cloud App yang memungkinkan pengguna dengan mudah memasukkan wawasan berbasis data ke dalam alur kerja harian. Fitur ini mendukung berbagi data secara *real-time* yang memungkinkan pengguna untuk membagikan visualisasi Tableau, dasbor, dan metrik Tableau Pulse secara langsung melalui obrolan dan pertemuan di Microsoft Teams.

Tableau menawarkan dua jenis layanan utama yang dirancang untuk memenuhi kebutuhan pengguna dengan skala dan kompleksitas yang berbeda, yaitu *Self-Service Analytic for Teams* dan *Enterprise Edition*. Kedua layanan ini bersifat berbayar dengan fitur yang disesuaikan berdasarkan tingkat kebutuhan organisasi atau tim. Berikut adalah penjelasan lebih rinci mengenai masing-masing layanan (Salesforce, 2024a).

1. *Self-Service Analytics for Teams*

Layanan ini dirancang untuk mendukung analisis mandiri dalam tim yang memungkinkan pengguna untuk mengeksplorasi, memvisualisasikan, dan membagikan data secara kolaboratif. Tableau menawarkan tiga paket utama dalam kategori ini:

- Tableau Creator

Biaya langganan sebesar \$75 per bulan per pengguna yang mencakup Tableau Desktop, Tableau Prep Builder, Tableau Pulse, dan satu *creator lisence* dari Tableau Cloud atau Server.

- Tableau Explorer

Paket ini dikenakan biaya \$42 per bulan per pengguna yang mencakup Tableau Pulse dan satu *explorer license* dari Tableau Cloud atau Server.

- Tableau Viewer

Paket ini tersedia dengan biaya \$15 per bulan per pengguna yang mencakup Tableau Pulse dan satu *viewer license* dari Tableau Cloud atau Server.

2. *Enterprise Edition*

Layanan ini dirancang untuk organisasi besar dengan kebutuhan pengelolaan data yang lebih kompleks dan mendalam. Paket yang ditawarkan dalam kategori ini meliputi:

- Enterprise Creator

Biaya langganan sebesar \$115 per bulan per pengguna dengan fitur mencakup Tableau Desktop, Tableau Prep Builder, Tableau Pulse, satu *creator license* pada Tableau Cloud atau Server dengan *advanced management* dan *data management*, serta akses *eLearning* untuk *creators* selama setahun.

- Enterprise Explorer

Paket ini dikenakan biaya \$70 per bulan per pengguna dan mencakup Tableau Pulse, satu *explorer license* dari Tableau Cloud atau Server dengan *advanced management* dan *data management*, serta akses *eLearning* untuk *explorers* selama setahun.

- Enterprise Viewer

Paket ini tersedia dengan biaya \$35 per bulan per pengguna, mencakup Tableau Pulse, dan satu *viewer license* dari Tableau Cloud atau Server dengan *advanced management* dan *data management*.

Informasi lebih lanjut mengenai Tableau, dapat dilihat pada dokumentasi resminya yang terdapat pada halaman “Tableau Help” dalam situs resmi Tableau yang dapat diakses melalui tautan <https://www.tableau.com/support/help>.

1.3. MySQL

MySQL adalah sistem manajemen basis data relasional atau yang dikenal dengan istilah *Relational Database Management System* (RDBMS) yang bersifat *open-source* dan mendukung *multithreaded* yang diciptakan oleh Michael “Monty” Widenius pada tahun 1995. Pada tahun 2000, MySQL dirilis di bawah model lisensi ganda yang memungkinkan publik untuk menggunakannya secara gratis di bawah GNU General Public License (GPL). Kombinasi fitur-fitur canggih, stabilitas, dan lisensi yang terbuka ini menjadikan popularitas MySQL meningkat pesat (Dyer, 2015).

MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu *Structured Query Language* (SQL). SQL adalah sebuah konsep pengoperasian basis data terutama untuk pemilihan (seleksi) dan pemasukan data yang memungkinkan pengolahan data dilakukan dengan mudah dan otomatis. Sebagai server basis data, MySQL dapat dikatakan lebih unggul dibandingkan dengan server basis data lainnya dalam hal pemrosesan kueri data. Hal ini terbukti bahwa pada kueri yang dilakukan oleh pengguna, kecepatan kueri MySQL bisa sepuluh kali lipat lebih cepat dibandingkan dengan PostgreSQL dan lima kali lebih cepat dibandingkan dengan Interbase (Maharani, 2017).

MySQL menawarkan berbagai fitur unggulan seperti dukungan penyimpanan data terstruktur dengan menggunakan SQL standar yang memungkinkan pengguna untuk melakukan kueri (*query*), agregasi, dan manipulasi data secara efisien. MySQL menerapkan arsitektur *client-server*, di mana server basis data (*database*) terinstal pada mesin *server* dan dapat diakses oleh *client* yang terhubung melalui jaringan lokal atau internet. Salah satu kemampuan penting yang dimiliki MySQL adalah dukungan untuk *sub-select* yang memungkinkan penggunaan kueri seleksi di dalam kueri seleksi lainnya. Selain itu, MySQL juga mendukung fitur-fitur seperti *views*, *stored procedures* (SP), dan *triggers* untuk pemrosesan data. MySQL juga menyediakan fitur *replication* yang memungkinkan data untuk disalin secara otomatis dari satu server ke server lain dan adanya *foreign key* yang memungkinkan pengguna untuk menjaga integritas referensial antar tabel, sementara transaksi membantu memastikan konsistensi data. MySQL juga dikenal dengan fungsi GIS (*Geographic Information System*) yang memungkinkan pengelolaan data spasial. Selain itu, MySQL dapat diunduh secara gratis dan terus berkembang pesat dengan pembaruan dan peningkatan fungsionalitas (Hadi, 2017; Solichin, 2010).

MySQL menawarkan dua jenis opsi pengunduhan, yaitu MySQL *Community Edition* yang gratis dan MySQL *Enterprise Edition* yang berbayar. Berikut ini adalah penjelasan mengenai kedu jenis MySQL tersebut.

- MySQL *Community Edition*

Merupakan versi MySQL yang dapat diunduh secara gratis di bawah lisensi GPL. Edisi ini menawarkan berbagai fitur penting, di antaranya dukungan untuk SQL dan NoSQL dalam pengembangan aplikasi relasional dan NoSQL. Fitur lainnya termasuk MySQL *Document Store* yang mencakup X *Protocol*, XDev API, dan MySQL Shell, serta *Transactional Data Dictionary* dengan pernyataan DDL atomik. MySQL juga menyediakan arsitektur *Pluggable Storage Engine* dengan opsi seperti InnoDB, NDB, dan MyISAM, serta kemampuan MySQL *Replication* dan MySQL *Group Replication* untuk meningkatkan skalabilitas dan ketersediaan data.

Selain itu, MySQL InnoDB *Cluster* menawarkan solusi ketersediaan tinggi, sementara MySQL *Router* memungkinkan *routing* transparan antar *server* MySQL. Fitur lain yang tersedia mencakup MySQL *Partitioning*, *Stored Procedures*, *Triggers*, *Views*, serta *Performance Schema* dan *Information Schema* untuk pemantauan dan akses metadata. MySQL *Community Edition* juga mendukung berbagai MySQL *Connectors* (*ODBC*, *JDBC*, *.NET*) dan MySQL *Workbench* (hingga MySQL versi 8.0.40) untuk pengembangan visual, SQL, dan administrasi. Semua fitur ini tersedia di lebih dari 20 platform dan sistem operasi, termasuk Linux, Unix, Mac, dan Windows.

- MySQL *Enterprise Edition*

MySQL Enterprise Edition merupakan versi yang berbayar dengan harga mulai dari Rp84.802.850 hingga Rp508.817.100 tergantung pada set yang dipilih melalui langganan di Oracle. Edisi ini menawarkan fitur, alat manajemen, dan dukungan teknis yang lebih lengkap untuk mencapai tingkat skalabilitas, keamanan, keandalan, dan waktu operasional MySQL yang lebih tinggi. Beberapa fitur utama dalam MySQL *Enterprise Edition* termasuk MySQL *Enterprise Backup* yang memungkinkan pencadangan “Hot” *online* untuk mengurangi risiko kehilangan data. MySQL *Enterprise High Availability* menyediakan solusi ketersediaan tinggi terintegrasi dengan MySQL *InnoDB Cluster* yang menggabungkan *Group Replication*, MySQL *Router*, dan MySQL *Shell*. Untuk skalabilitas, MySQL *Enterprise Scalability* memastikan kinerja berkelanjutan dengan MySQL *Thread Pool* untuk menangani koneksi dan eksekusi pernyataan secara lebih efisien.

MySQL Enterprise Edition juga memperkenalkan fitur baru berupa MySQL *Enterprise Stored Programs* yang memungkinkan pengembang menulis program dan fungsi berbasis *JavaScript* di dalam MySQL. Dalam hal keamanan, MySQL menawarkan MySQL *Enterprise Authentication* yang memudahkan integrasi dengan infrastruktur keamanan yang ada seperti PAM dan Active Directory, serta fitur enkripsi. Fitur lainnya termasuk MySQL *Enterprise Encryption* untuk perlindungan data dengan enkripsi dan tanda tangan digital, MySQL *Enterprise Masking* and *De-identification* untuk melindungi data sensitif, dan MySQL *Enterprise Firewall* untuk perlindungan terhadap ancaman keamanan. MySQL *Enterprise Audit* menyediakan kemampuan audit berbasis kebijakan, sementara MySQL *Enterprise Telemetry* memungkinkan pemantauan kinerja menggunakan dan MySQL *Enterprise Monitor* dapat memantau secara *real-time*. Serta, MySQL *Query Analyzer* tersedia untuk menganalisis kueri (Oracle, 2024b).

Informasi lebih lanjut mengenai MySQL, dapat dilihat pada dokumentasi resminya yang terdapat situs resmi MySQL yang dapat diakses melalui tautan <https://dev.mysql.com/doc/>.

BAB II

USULAN DATA WAREHOUSE MENGGUNAKAN MYSQL, PENTAHO DATA INTEGRATION, DAN TABLEAU

2.1. Variabel dan Sumber Data

Data yang digunakan berasal dari platform Open-Meteo, sebuah penyedia data cuaca berbasis API yang menawarkan data historis, prakiraan, dan analisis cuaca secara detail. Sumber data ini dipilih karena keandalan dan kelengkapannya dalam menyediakan data cuaca dari berbagai lokasi secara konsisten dan akurat. Data yang diambil untuk proyek ini merupakan data historis cuaca (*historical weather*) dengan periode harian yang mencakup periode waktu dari 1 Januari 2000 hingga 30 November 2024.

Proyek ini secara khusus fokus pada data cuaca dari 31 kecamatan yang ada di Kota Surabaya. Pendekatan berbasis wilayah ini bertujuan untuk memastikan bahwa analisis yang dilakukan dapat memberikan gambaran yang representatif dan mendalam terhadap kondisi cuaca di setiap kecamatan. Data dikelompokkan urut berdasarkan kecamatan untuk menjaga akurasi informasi pada tingkat mikro, sehingga dapat mendukung kebijakan dan tindakan spesifik sesuai dengan kebutuhan masing-masing wilayah.

Data historis cuaca yang digunakan mencakup variabel-variabel penting yang relevan untuk analisis kondisi lingkungan di Kota Surabaya. Variabel-variabel ini dipilih karena data ini berfungsi sebagai dasar dalam membangun sistem pemantauan cuaca yang terintegrasi. Berikut adalah rincian variabel yang digunakan dalam proyek ini:

Tabel 2.1. Variabel Data

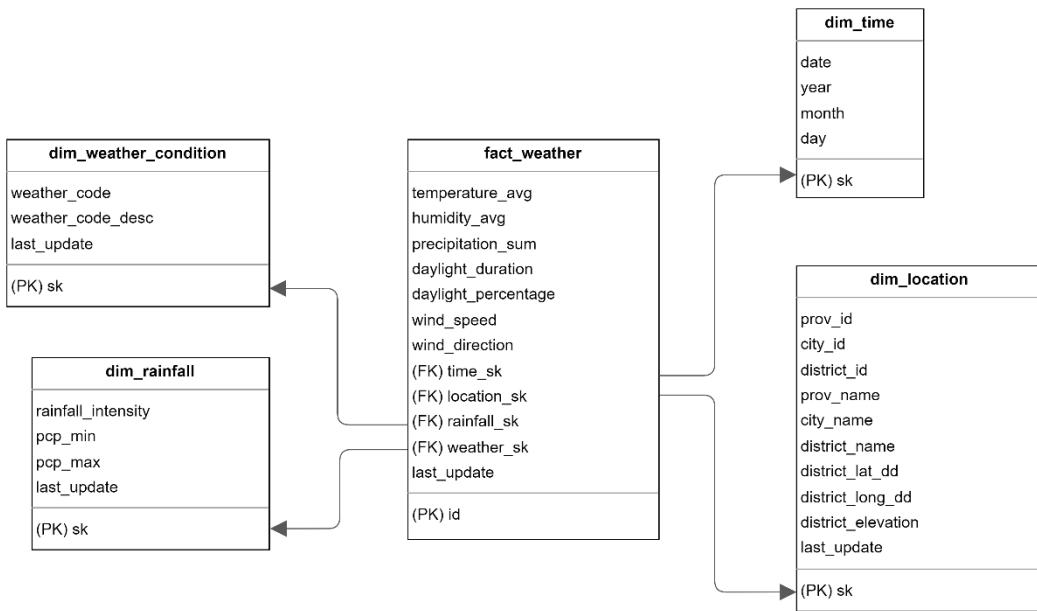
Variabel	Deskripsi
<i>location_id</i>	Variabel berisi nilai unik yang digunakan untuk mengidentifikasi lokasi pengamatan, dalam hal ini 31 kecamatan di Kota Surabaya. Variabel ini membantu memetakan data cuaca ke lokasi spesifik.
<i>latitude</i>	Koordinat lintang geografis suatu lokasi, digunakan untuk menentukan posisi lokasi pengamatan secara horizontal di permukaan bumi.
<i>longitude</i>	Koordinat bujur geografis suatu lokasi, digunakan untuk menentukan posisi lokasi pengamatan secara vertikal di permukaan bumi.
<i>elevation</i>	Ketinggian lokasi pengamatan dari permukaan laut, biasanya dalam satuan meter. Variabel ini penting karena ketinggian dapat memengaruhi pola cuaca di suatu area.
<i>id_waktu</i>	Variabel berisi nilai unik yang menghubungkan data lokasi dengan data temporal (tahun, bulan, dan tanggal). Variabel ini digunakan untuk menyelaraskan data waktu dengan lokasi tertentu.

Tahun	Informasi tahun pengamatan data cuaca, dalam format numerik. Pada proyek ini, mencakup periode Januari hingga Oktober 2024
Bulan	Informasi bulan pengamatan, dalam format numerik (1 untuk Januari hingga 12 untuk Desember)
Tanggal	Informasi tanggal spesifik dalam bulan tertentu yang mengindikasikan data cuaca harian
<i>weather_code</i>	Kode unik yang merepresentasikan kondisi cuaca pada suatu waktu tertentu, seperti hujan, cerah, atau mendung. Variabel ini berguna untuk klasifikasi pola cuaca
<i>avg_temperature</i>	Suhu rata-rata harian pada lokasi tertentu, biasanya dalam satuan derajat Celsius. Variabel ini merupakan indikator penting untuk memantau perubahan suhu
<i>sunlight_percentage</i>	Persentase durasi siang hari dalam satu hari. Variabel ini mengukur tingkat penerimaan energi matahari di suatu wilayah
<i>precipitation_intensity</i>	Intensitas curah hujan yang diukur dalam milimeter (mm). Variabel ini menunjukkan seberapa banyak hujan turun di lokasi tertentu dalam sehari
<i>wind_speed_10m_max</i>	Kecepatan angin maksimum pada ketinggian 10 meter dari permukaan tanah, biasanya diukur dalam satuan meter per detik (m/s). Variabel ini relevan untuk menganalisis pola angin
<i>wind_direction_10m_dominant</i>	Arah dominan angin pada ketinggian 10 meter dari permukaan tanah, dinyatakan dalam derajat (0-360). Variabel ini membantu memahami pola pergerakan angin

Berdasarkan variabel-variabel yang tercantum pada tabel 2.1, analisis pengembangan data *warehouse* untuk pemantauan cuaca di Kota Surabaya bertujuan untuk mengidentifikasi hubungan antara kondisi cuaca dengan empat indikator utama cuaca, yaitu suhu, durasi siang hari, intensitas curah hujan, dan kecepatan angin. Analisis ini diharapkan dapat mengungkap pola atau kecenderungan dari keempat indikator cuaca tersebut, sehingga memberikan gambaran yang lebih menyeluruh mengenai kondisi cuaca yang terjadi di Kota Surabaya.

2.2. Basis Data *Online Analytical Processing* (OLAP)

Untuk merealisasikan analisis ini, diperlukan struktur data *warehouse* yang mampu mengelola berbagai variabel terkait cuaca secara terorganisasi. Gambar 2.1 menunjukkan skema data *warehouse* yang dirancang.



Gambar 2.1. Diagram Skema/Basis Data *Online Analytical Processing* (OLAP)

Berdasarkan Gambar 2.1 yang disajikan, model basis data ini dirancang untuk menyimpan dan menganalisis informasi terkait cuaca melalui skema *star schema*. Tabel fakta utama, “fact_weather”, menjadi pusat penyimpanan data cuaca yang terhubung dengan empat tabel dimensi, yaitu “dim_time”, “dim_location”, “dim_rainfall”, dan “dim_weather_condition”. Tabel-tabel dimensi tersebut menyediakan informasi tambahan yang relevan untuk analisis yang lebih detail. Berikut adalah penjelasan lebih rinci mengenai setiap tabel dimensi dan atribut-atributnya.

1. Tabel Dimensional:

- “dim_time”

Tabel ini berfungsi sebagai penyedia informasi waktu yang terperinci untuk setiap data cuaca yang tercatat. Atribut-atribut tabel ini meliputi:

- “sk” (*Primary Key*): kode unik untuk setiap data waktu,
- “date”: tanggal data cuaca tercatat,
- “year”: tahun data cuaca tercatat,
- “month”: bulan data cuaca tercatat,
- “day”: tanggal hari data cuaca tercatat.

- “dim_location”

Tabel ini menyimpan informasi geografis terkait lokasi tempat pencatatan data cuaca. Atribut-atribut tabel ini meliputi:

- “sk” (*Primary Key*): kode unik untuk setiap lokasi,
- “prov_id”: kode unik untuk setiap provinsi tempat lokasi tersebut berada,
- “city_id”: kode unik untuk setiap kota tempat lokasi tersebut berada,
- “district_id”: kode unik untuk setiap kecamatan tempat lokasi tersebut berada.
- “prov_name”: nama provinsi,
- “city_name”: nama kota,

- “district_name”: nama distrik,
- “district_lat_dd”: koordinat lintang (*latitude*) kecamatan dalam derajat desimal atau decimal degree (dd),
- “district_long_dd”: koordinat bujur (*longitude*) kecamatan dalam derajat desimal atau *decimal degree* (dd),
- “district_elevation”: ketinggian kecamatan dari permukaan laut (dalam meter),
- “last_update”: waktu saat data lokasi terakhir diperbarui.

- “dim_rainfall”

Tabel ini menyimpan jenis intensitas hujan beserta dengan deskripsiya berdasarkan curah hujan dalam cuaca yang tercatat. Atribut-atribut tabel ini meliputi:

- “sk” (Primary Key): kode unik untuk setiap jenis intensitas hujan,
- “rainfall_intensity”: intensitas hujan (misalnya ringan, sedang, atau lebat),
- “pcp_min”: curah hujan minimum,
- “pcp_max”: curah hujan maksimum,
- “last_update”: waktu saat data intensitas hujan terakhir diperbarui.

- dim_weather_condition

Tabel ini menyediakan informasi tentang kondisi cuaca yang tercatat dalam cuaca yang tercatat. Atribut-atribut tabel ini meliputi:

- “sk” (Primary Key): kode unik untuk setiap kondisi cuaca,
- “weather_code”: kode unik untuk setiap kondisi cuaca sesuai dengan World Meteorological Organization (WMO) Code 4677.
- “weather_code_desc”: deskripsi kondisi cuaca (misalnya, cerah, berawan, hujan).
- “last_update”: waktu saat data kondisi cuaca terakhir diperbarui.

2. Tabel Fakta (fact_weather)

Tabel ini berisi data metrik cuaca yang terhubung ke tabel dimensi waktu, lokasi, intensitas hujan, dan kondisi cuaca. Atribut-atribut dalam tabel ini meliputi:

- “id” (Primary Key): Kode unik untuk setiap *record* data cuaca,
- “temperature_avg”: suhu rata-rata harian (dalam derajat celsius),
- “humidity_avg”: kelembaban rata-rata harian (dalam persen),
- “precipitation_sum”: jumlah curah hujan harian (dalam milimeter),
- “daylight_duration”: durasi siang hari dalam satu hari (dalam detik).
- “daylight_percent”: persentase durasi siang hari pada suatu lokasi dan waktu tertentu dalam satu hari.
- “wind_speed”: kecepatan angin rata-rata harian (dalam meter per detik),
- “wind_direction”: arah angin (dalam derajat),
- “time_sk” (Foreign Key): kode unik untuk setiap data waktu yang merujuk ke kolom “sk” dalam tabel “dim_time”,
- “location_sk” (Foreign Key): kode unik untuk setiap data lokasi yang merujuk ke kolom “sk” dalam tabel “dim_location”,
- “rainfall_sk” (Foreign Key): kode unik untuk setiap data intensitas hujan yang merujuk ke kolom “sk” dalam tabel “dim_rainfall”,

- “weather_sk” (Foreign Key): kode unik untuk setiap data kondisi cuaca yang merujuk ke kolom “sk” dalam tabel “dim_weather_condition”,
- “last_update”: waktu saat data cuaca harian terakhir diperbarui.

3. Hubungan antar tabel:

Tabel fakta “fact_weather” terhubung dengan:

- “dim_time” melalui atribut “time_sk”.
- “dim_location” melalui atribut “location_sk”.
- “dim_rainfall” melalui atribut “rainfall_sk”.
- “dim_weather_condition” melalui atribut “weather_sk”.

Skema data *warehouse* ini dirancang agar setiap tabel dimensi mendukung analisis secara terstruktur melalui tabel fakta. Melalui hubungan antara tabel dimensi dan fakta, data warehouse ini diharapkan mampu menyediakan data yang terperinci untuk menjawab berbagai kebutuhan analisis terkait cuaca harian di Kota Surabaya.

2.3. Tujuan Penggunaan MySQL, Pentaho Data Integration, dan Tableau (Tableau Desktop)

2.3.1. MySQL 8.4.3 (Community)

Dalam proyek ini, digunakan MySQL 8.4.3 sebagai basis data untuk pengelolaan data dalam proyek ini karena stabilitas, kinerja tinggi, dan fleksibilitas yang dimiliki. MySQL merupakan salah satu basis data (*database*) relasional *open-source* yang populer dan banyak digunakan sehingga didukung dengan komunitas yang luas dan dokumentasi yang komprehensif. MySQL 8.4.3 dirilis dengan fitur-fitur canggih seperti peningkatan kecepatan pemrosesan kueri (*queries*), dukungan bentuk JavaScript Object Notation (JSON) untuk data semi-terstruktur, serta keamanan yang lebih baik melalui otentikasi berbasis *plugin*. Hal ini menjadikan MySQL sangat cocok untuk kebutuhan proyek yang melibatkan pengelolaan data dalam volume besar dan beragam, seperti data cuaca yang harus disimpan secara terstruktur dan dapat diakses dengan cepat.

Tujuan utama penggunaan MySQL Server 8.4.3 pada proyek ini adalah untuk membangun data *warehouse* yang dapat menampung data dari berbagai sumber dengan struktur yang efisien dan terorganisir. MySQL digunakan untuk menyimpan data yang telah diproses oleh Pentaho Data Integration (PDI) sehingga memungkinkan akses data yang cepat dan konsisten untuk keperluan analisis dan visualisasi di Tableau. Melalui arsitektur MySQL yang mendukung skalabilitas, basis data ini mampu menangani pertumbuhan data yang terus meningkat seiring waktu. Selain itu, MySQL memungkinkan implementasi indeks dan optimasi kueri yang membantu dalam pengambilan data secara efisien sehingga mendukung keseluruhan proses monitoring cuaca di Kota Surabaya. Fitur-fitur pada MySQL yang digunakan dalam proyek ini dirangkum dalam Tabel 2.2.

Tabel 2.2. Fitur MySQL yang Digunakan dalam Proyek

Fitur	Keterangan
 Create database	Fitur “Create database” digunakan untuk membuat <i>database</i> baru di dalam server MySQL. Saat membuat <i>database</i> , pengguna bisa menentukan nama

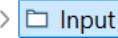
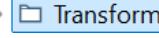
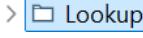
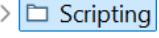
	<i>database</i> , serta memilih pengaturan tertentu seperti <i>collation</i> (penentuan set karakter) yang akan digunakan di dalam <i>database</i> .
 SQL	Fitur “SQL” memungkinkan pengguna untuk menulis dan menjalankan <i>kueri</i> SQL secara langsung pada <i>database</i> . Pengguna bisa menjalankan berbagai jenis perintah SQL, seperti SELECT, INSERT, UPDATE, DELETE, serta perintah DDL (<i>Data Definition Language</i>) untuk membuat atau mengubah struktur <i>database</i> , tabel, indeks, dan sebagainya.
 Designer	Fitur “Designer” memungkinkan pengguna untuk membuat desain visual skema database. Dengan fitur ini, pengguna dapat menggambar dan merancang struktur tabel-tabel dalam <i>database</i> dan menetapkan hubungan antar tabel (seperti relasi satu-ke-banyak atau banyak-ke-banyak) menggunakan antarmuka grafis. Fitur ini memudahkan untuk melihat dan mengelola struktur database secara keseluruhan.
 Import	Fitur “Import” memungkinkan pengguna untuk mengimpor data dari file eksternal (seperti file CSV, SQL, atau format lainnya) ke dalam database yang telah dibuat. Hal ini mempermudah proses migrasi data atau memuat data besar dari sistem lain ke dalam <i>database</i> MySQL.

2.3.2. Pentaho Data Integration 9.4.0.0-343 (Developer/Community Edition)

Pentaho Data Integration (PDI) 9.4 digunakan sebagai *tools* utama dalam proyek ini karena kemampuannya yang unggul dalam menangani proses *extract, transform, and load* (ETL). PDI memiliki *user interface* berbasis grafis yang intuitif sehingga memudahkan perancangan alur integrasi data tanpa memerlukan penulisan kode yang kompleks. Selain itu, PDI mendukung berbagai sumber data, termasuk berkas *flat*, API, basis data relasional, dan data berbasis *cloud* yang sesuai dengan kebutuhan proyek ini dalam mengintegrasikan data cuaca dari berbagai sumber. Kemampuan PDI dalam menangani volume data yang besar dan beragam format data menjadikannya pilihan yang tepat untuk membangun data *warehouse* yang handal dan efisien.

Tujuan utama penggunaan PDI pada proyek ini adalah untuk memastikan bahwa data cuaca yang melalui API dapat diekstrak, diubah, dan dimuat ke dalam *data warehouse* dengan cara yang terstruktur dan terotomatisasi. PDI memungkinkan proses transformasi data seperti pembersihan, penggabungan, dan agregasi data agar dapat digunakan secara optimal dalam analisis. Melalui *workflow* yang terorganisir di PDI, integritas dan kualitas data dapat terjaga sehingga *data warehouse* yang dihasilkan dapat memberikan informasi yang akurat dan relevan untuk mendukung visualisasi di Tableau. Pada akhirnya, integrasi ini bertujuan untuk memberikan solusi yang efisien dan dapat diandalkan dalam pemantauan cuaca harian di Kota Surabaya. Fitur-fitur pada PDI yang digunakan dalam proyek ini dirangkum dalam Tabel 2.3.

Tabel 2.3. Fitur Pentaho Data Integration (PDI) yang Digunakan dalam Proyek

Fitur	Keterangan
 Transformation	Sebuah proses atau alur kerja yang berisi langkah-langkah untuk mengekstraksi, mengubah, dan memuat (ETL) data. "Transformation" berfungsi untuk mengolah data secara detail dengan berbagai langkah transformasi. Fungsi "transformasi" adalah untuk memanipulasi data dari berbagai sumber sebelum memuatnya ke dalam <i>database target</i> atau <i>data warehouse</i> .
 Job	Digunakan untuk mengatur eksekusi alur kerja atau tugas-tugas yang lebih besar, termasuk menjalankan satu atau beberapa "transformation". Berfungsi untuk mengatur proses otomatisasi atau orkestrasi tugas-tugas yang saling berhubungan, seperti memanggil "transformation" tertentu, menangani file, atau mengirim notifikasi setelah proses selesai
>  Input	Fitur "Input" adalah kategori yang berisi langkah-langkah untuk mengambil atau membaca data dari berbagai sumber (file, <i>database</i> , API, dll.). "Input" digunakan untuk mengimpor data dari sumber eksternal yang akan diolah dalam alur "transformation".
>  Transform	Fitur "Transform" adalah kategori yang berisi langkah-langkah untuk memodifikasi, memanipulasi, atau mengubah data. "Transform" bertujuan untuk melakukan pembersihan data (<i>data cleaning</i>), transformasi, agregasi, atau normalisasi sebelum data dimuat ke tujuan akhir.
>  Lookup	Fitur "Lookup" adalah kategori yang memungkinkan pencarian atau pengambilan data terkait dari sumber lain berdasarkan kunci tertentu. "Lookup" digunakan untuk mencocokkan atau memperkaya <i>dataset</i> dengan informasi tambahan dari tabel atau sumber lain.
>  Scripting	Berisi langkah-langkah "scripting" yang memungkinkan pengguna untuk menulis skrip khusus, seperti <i>JavaScript</i> atau bahasa lain yang didukung. Tujuan dari sub <i>tools scripting</i> adalah memberikan fleksibilitas lebih dalam memproses data jika fitur bawaan Pentaho tidak mencukupi kebutuhan pengguna.
 Generate rows	Fitur ini digunakan untuk membuat baris data secara manual atau otomatis dengan nilai yang ditentukan oleh pengguna. "Generate rows" digunakan untuk keperluan <i>debugging</i> atau pembuatan data <i>dummy</i> dalam skenario pengujian alur "transformation".
 Add sequence	Langkah untuk menambahkan urutan angka ke data yang diolah. "Add sequence" digunakan untuk menambahkan kolom berisi nomor urut yang sering digunakan sebagai kunci unik (<i>primary key</i>) atau penanda urutan data.
 Calculator	Sebuah langkah untuk melakukan perhitungan matematis pada kolom data. Berfungsi untuk menambahkan kolom hasil perhitungan, seperti

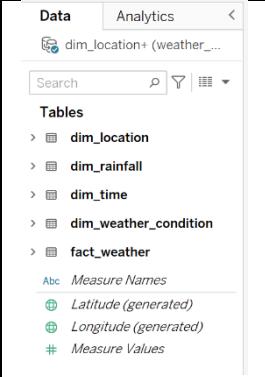
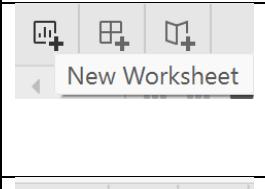
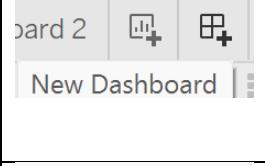
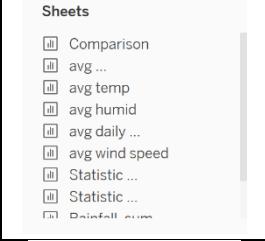
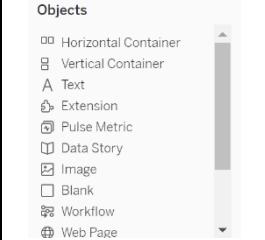
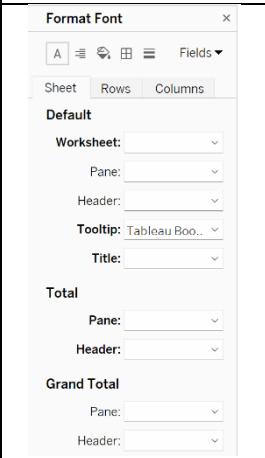
	menghitung nilai rata-rata, penjumlahan, atau operasi matematis lainnya.
 Hop	Representasi garis penghubung antara langkah-langkah dalam “transformation” atau “job”. Digunakan untuk mengarahkan aliran data dari satu langkah ke langkah berikutnya dalam alur kerja ETL.
 Data grid	Sebuah langkah untuk membuat data secara manual langsung di dalam PDI. Memungkinkan pengguna untuk membuat <i>dataset</i> kecil secara manual untuk pengujian atau simulasi alur ETL.
 Stream lookup	Langkah yang digunakan untuk mencocokkan data dari dua aliran data (<i>streams</i>) yang berbeda berdasarkan kunci tertentu. Digunakan untuk memperkaya <i>dataset</i> dengan informasi tambahan dari aliran data lain tanpa harus melakukan pencarian langsung ke <i>database</i> . Hal ini berguna dalam integrasi data <i>real-time</i> .
 Modified JavaScript Value	Fitur yang memungkinkan pengguna menulis skrip <i>JavaScript</i> untuk memproses atau memanipulasi data di dalam “transformation”. Memberikan fleksibilitas untuk melakukan perhitungan kompleks, validasi data, atau modifikasi khusus yang tidak didukung oleh langkah bawaan PDI.
 Table output	Langkah untuk menulis data hasil transformasi ke dalam tabel di <i>database</i> . Digunakan untuk memuat data yang sudah diolah (ETL) ke dalam <i>database</i> target, seperti <i>data warehouse</i> atau <i>database</i> operasional.
 Table input	Langkah untuk membaca data dari tabel di <i>database</i> . Berfungsi untuk mengambil data mentah dari tabel di <i>database</i> sebagai bahan untuk diproses lebih lanjut dalam <i>pipeline</i> ETL.
 Get system info	Langkah yang digunakan untuk mendapatkan informasi sistem, seperti waktu saat ini, nama <i>server</i> , atau <i>metadata</i> lainnya. Berguna untuk menambahkan informasi sistem ke dalam data, seperti <i>timestamp</i> untuk pelacakan atau log proses.
 Select values	Fitur yang memungkinkan pengguna memilih kolom tertentu, mengganti nama kolom, atau mengubah tipe data. Digunakan untuk menyaring, memodifikasi, atau membersihkan kolom data sebelum melanjutkan ke langkah transformasi berikutnya.
 Database lookup	Langkah untuk mencari data di <i>database</i> berdasarkan kunci tertentu. Digunakan untuk memperkaya <i>dataset</i> dengan informasi tambahan yang diperoleh dari <i>database</i> , seperti menambahkan nama pelanggan berdasarkan ID
 Start	Titik awal eksekusi sebuah “job” atau “transformation”. Menandai awal alur kerja dalam “job” atau “transformation” dan memastikan semua langkah setelahnya berjalan sesuai urutan

2.3.3. Tableau Desktop 2024.3

Tableau Desktop 2024.3 dipilih sebagai *tools* utama untuk pembuatan visualisasi dalam proyek ini karena keunggulannya dalam menghadirkan analisis data yang cepat, interaktif, dan mudah digunakan. Lebih lanjut, melalui fitur *drag-and-drop* yang intuitif, Tableau memungkinkan pembuatan dasbor yang efisien tanpa memerlukan kemampuan pemrograman yang kompleks. Selain itu, *tools* ini mampu mengolah data yang kompleks menjadi visualisasi yang menarik dan informatif, seperti grafik, peta, atau diagram interaktif. Hal ini sangat relevan untuk kebutuhan proyek ini, di mana data cuaca harian tersimpan dalam *data warehouse* perlu disajikan secara interaktif agar mudah dipahami oleh berbagai pihak, termasuk masyarakat umum dan pemangku kebijakan. Keunggulan lainnya dari *tools* ini adalah kemampuannya yang mendukung berbagai format data, termasuk koneksi langsung dengan basis data relasional seperti MySQL yang mempermudah integrasi data cuaca dari *data warehouse*. Fitur-fitur pada Tableau Desktop yang digunakan dalam proyek ini dirangkum dalam Tabel 2.4.

Tabel 2.4. Fitur Tableau Desktop yang Digunakan dalam Proyek

Fitur	Keterangan
iii Columns	“Columns” digunakan untuk menempatkan data di bagian atas <i>grid</i> yang akan menentukan dimensi horizontal dari tampilan visual.
Rows	“Rows” digunakan untuk menempatkan data di bagian samping <i>grid</i> yang menentukan dimensi vertikal dari tampilan visual.
Marks	<p>“Marks” adalah bagian untuk mengontrol bagaimana data divisualisasikan. “Marks” card dapat menyesuaikan tampilan visualisasi, seperti warna, ukuran, label, bentuk, dan detail. Ada 5 bagian utama di “Marks” card:</p> <ul style="list-style-type: none"> • “Color”: Mengontrol warna elemen visualisasi (misalnya, bar, garis, atau titik). “Color” dapat membedakan kategori atau menggambarkan intensitas nilai numerik melalui skala warna. • “Size”: Mengatur ukuran elemen visual. Ukuran dapat digunakan untuk menyoroti kepentingan atau bobot data. • “Text/Label”: Mengaktifkan dan menyesuaikan label pada elemen visualisasi. “Text/label” dapat menampilkan nilai numerik, nama kategori, atau informasi lainnya langsung di atas elemen • “Detail”: Menambahkan rincian lebih lanjut tanpa mengubah bentuk dasar visualisasi. • “Tooltip”: Menentukan informasi yang akan muncul ketika pengguna <i>hover</i> di atas elemen visual.
Filters	“Filters” digunakan untuk memfilter data yang ditampilkan dalam visualisasi, sehingga hanya data yang relevan atau sesuai dengan kriteria tertentu yang muncul.

	<p>Tab “Data” adalah bagian di sisi kiri antarmuka yang menampilkan semua sumber data yang Anda gunakan untuk membuat visualisasi.</p>
	<p>“Worksheet” digunakan sebagai ruang kerja membuat visualisasi individu seperti grafik batang, garis, peta, atau <i>scatter plot</i>. Untuk menambahkan <i>worksheet</i>, dapat menekan tombol yang berada di paling kiri.</p>
	<p>“Dashboard” adalah ruang kerja yang menggabungkan beberapa <i>worksheet</i> atau visualisasi berbeda ke dalam satu tampilan. Untuk menambahkan “dashboard”, dapat menekan tombol yang berada di paling kanan.</p>
	<p>“Size” adalah salah satu fitur dalam dashboard untuk mengatur ukuran dari dashboard. Terdapat 3 jenis ukuran yaitu <i>fixed size</i>, <i>automatic</i>, dan <i>range</i>.</p>
	<p>Tab “Sheets” dalam <i>dashboard</i> berfungsi untuk menampilkan semua elemen <i>worksheet</i> yang dibuat.</p>
	<p>“Objects” dalam “dashboard” berfungsi untuk menambahkan elemen – elemen lain selain daripada <i>worksheet</i> seperti <i>image</i>, <i>text</i>, dan lain sebagainya.</p>
	<p>“Format” adalah salah satu komponen dari visualisasi data yang dipilih untuk mengatur tampilan visual dari segi ukuran <i>font</i>, jenis <i>font</i>, warna, dan sebagainya.</p>

BAB III

TAHAPAN PENGGUNAAN

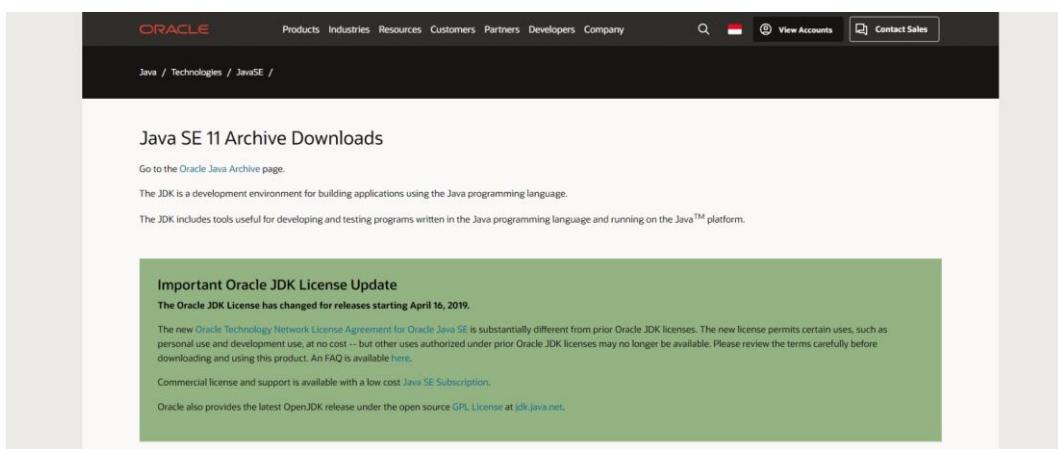
MYSQL, PENTAHO DATA INTEGRATION, DAN TABLEAU

Proyek ini dijalankan pada komputer bersistem operasi Windows 64-bit dengan memanfaatkan beberapa perangkat lunak (*software*) *data warehouse*. Perangkat lunak utama yang digunakan meliputi basis data MySQL, Pentaho Data Integration, dan Tableau Desktop. Untuk mendukung proses pengembangan, digunakan DataGrip sebagai *visual database design tools* dan *integrated development environment* (IDE) untuk mengelola basis data MySQL. Selain itu, *Java Runtime Environment* (JRE) juga digunakan untuk mendukung operasi *Pentaho Data Integration* dan konektivitasnya terhadap basis data MySQL.

3.1. Pengunduhan *Software Data Warehouse*

3.1.1. Pengunduhan Java Development Kit (JDK) 11

Pentaho Data Integration (PDI) dan komponen MySQL, seperti MySQL Connector/J memanfaatkan Java Runtime Environment (JRE) untuk operasinya. Dengan demikian, ketersediaan JRE pada sistem operasi merupakan suatu keharusan. Apabila JRE belum tersedia, instalasi Java Development Kit (JDK) dapat dilakukan karena JRE termasuk dalamnya. Berdasarkan dokumentasi resmi dari PDI 9.4.0.0-343 dan MySQL 8.4.3, kedua program ini mendukung JDK 8 dan 11 (Hitachi Vantara, 2024b; Oracle, 2024a). Oleh karena itu, dipilih JDK 11 yang merupakan versi terbaru dari Java yang didukung oleh PDI 9.4.0.0-343 dan MySQL 8.4.3. Pengunduhan JDK 11 dapat dilakukan dengan mengakses halaman web “Java SE 11 Archive Downloads” pada situs web resmi Oracle yang dapat diakses melalui tautan <https://www.oracle.com/id/java/technologies/javase/jdk11-archive-downloads.html>.



Gambar 3.1. Halaman Web “Java SE 11 Archive Downloads” pada Situs Web Resmi Oracle

Selanjutnya, dipilih versi berkas instalasi JDK 11 yang terbaru dan sesuai dengan sistem operasi yang digunakan, yakni Java SE Development Kit 11.0.24 Windows x64 Installer dengan ekstensi “.exe”. Pemilihan berkas instalasi ini didasarkan pada kompatibilitasnya dengan sistem operasi Windows yang digunakan dalam proyek ini, serta kemudahan

penggunaannya melalui antarmuka grafis yang disediakan untuk memandu proses instalasi.

Java SE Development Kit 11.0.24		
This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE		
JDK 11.0.24 checksum		
Product / File Description	File Size	Download
Linux ARM64 RPM Package	159.58 MB	*jdk-11.0.24_linux-aarch64_bin.rpm
Linux ARM64 Compressed Archive	159.69 MB	*jdk-11.0.24_linux-aarch64_bin.tar.gz
Linux x64 Debian Package	158.64 MB	*jdk-11.0.24_linux-x64_bin.deb
Linux x64 RPM Package	160.99 MB	*jdk-11.0.24_linux-x64_bin.rpm
Linux x64 Compressed Archive	161.10 MB	*jdk-11.0.24_linux-x64_bin.tar.gz
macOS ARM64 Compressed Archive	154.27 MB	*jdk-11.0.24_macos-aarch64_bin.tar.gz
macOS ARM64 DMG Installer	153.75 MB	*jdk-11.0.24_macos-aarch64_bin.dmg
macOS x64 Compressed Archive	156.55 MB	*jdk-11.0.24_macos-x64_bin.tar.gz
macOS x64 DMG Installer	156.06 MB	*jdk-11.0.24_macos-x64_bin.dmg
Solaris SPARC Compressed Archive	185.26 MB	*jdk-11.0.24_solaris-sparcv9_bin.tar.gz
Windows x64 Installer	141.70 MB	*jdk-11.0.24_windows-x64_bin.exe
Windows x64 Compressed Archive	159.46 MB	*jdk-11.0.24_windows-x64_bin.zip

Gambar 3.2. Opsi Berkas Instalasi Java SE Development Kit 11.0.24 pada Situs Web Resmi Oracle

3.1.2. Pengunduhan MySQL

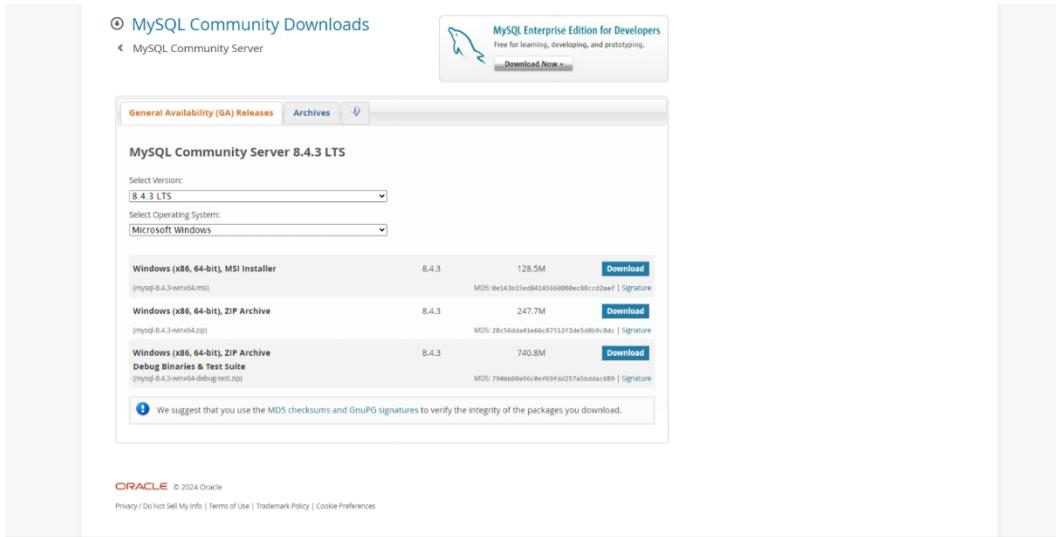
a. Pengunduhan MySQL Server 8.4.3 (Community)

Proses pengunduhan MySQL Server edisi komunitas (*community*) diawali dengan mengakses halaman web “MySQL Community Downloads” pada situs web resmi MySQL (Developer Zone) yang dapat diakses melalui tautan <https://dev.mysql.com/downloads/>. Kemudian, dipilih produk MySQL Community Server dengan mengeklik hipertaut yang terkait.



Gambar 3.3. Halaman web “MySQL Community Downloads” pada Situs Web Resmi MySQL (Developer Zone)

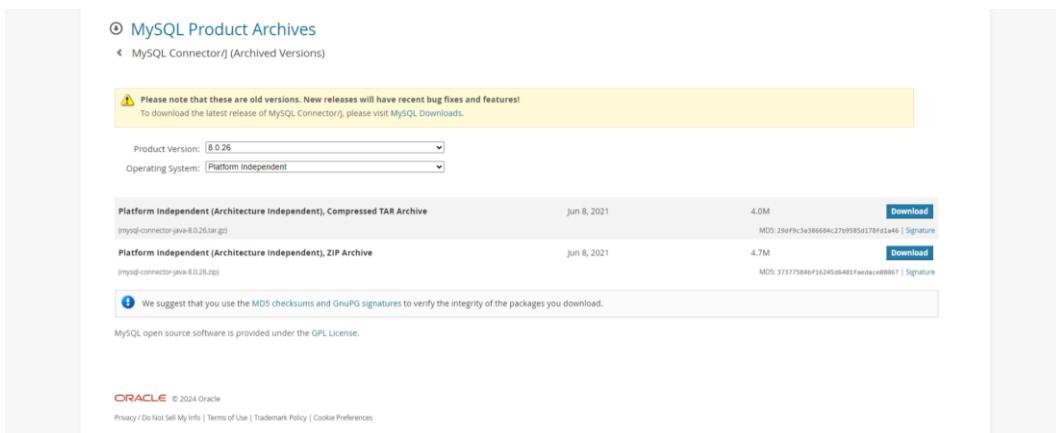
Selanjutnya, dipilih versi 8.4.3 LTS dan sistem operasi Windows (Microsoft Windows) pada dropdown yang tersedia pada menu “General Availability (GA) Releases”. Setelah itu, dipilih berkas instalasi dalam bentuk MSI Installer dan mengeklik tombol “Download” yang sesuai untuk berkas instalasi tersebut untuk memulai pengunduhan.



Gambar 3.4. Opsi Pengunduhan Berkas Instalasi MySQL Community Server

b. Pengunduhan MySQL Connector/J

Pentaho Data Integration (PDI) memerlukan *driver* JDBC untuk membuat koneksi dengan berbagai basis data, termasuk MySQL. Dengan demikian, ketersediaan *driver* JDBC MySQL pada PDI merupakan suatu keharusan. *Driver* ini dapat ditemukan dalam berkas MySQL Connector/J. Berdasarkan dokumentasi resmi dari PDI 9.4.0.0-343, perangkat ini mendukung *driver* JDBC MySQL versi 5.7 dan 8.0.26. Oleh karena itu, dipilih MySQL Connector/J versi 8.0.26 yang merupakan versi terbaru dari *driver* JDBC MySQL yang didukung oleh PDI 9.4.0.0-343 dan dapat digunakan untuk terkoneksi MySQL 8.4.3. Selanjutnya, pengunduhan *driver* JDBC MySQL diawali dengan mengakses halaman web “MySQL Product Archieves – MySQL Connector/J (Archived Versions)” pada situs web resmi MySQL (Developer Zone) yang dapat diakses melalui tautan <https://downloads.mysql.com/archives/c-j/>.

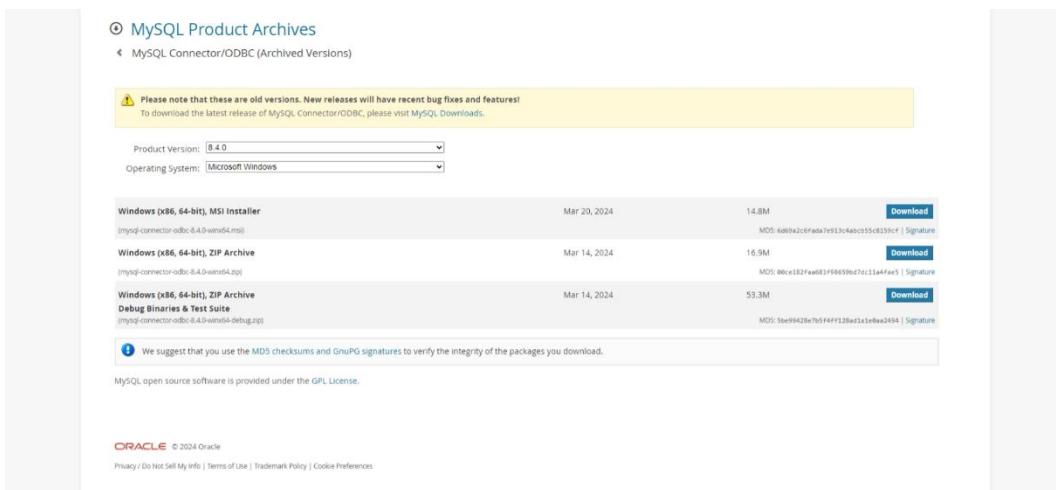


Gambar 3.5. Halaman Web “MySQL Product Archieves – MySQL Connector/J (Archived Versions)” pada Situs Web Resmi MySQL (Developer Zone)

Selanjutnya, dipilih versi 8.0.26 dan sistem operasi yang berupa “Platform Independent” pada *dropdown* yang tersedia. Setelah itu, dipilih berkas instalasi dalam bentuk ZIP Archive dan mengeklik tombol “Download” yang sesuai untuk berkas instalasi tersebut untuk memulai pengunduhan.

c. Pengunduhan MySQL Connector/ODBC

Tableau Desktop memerlukan *driver* ODBC untuk membuat koneksi dengan berbagai basis data, termasuk MySQL. Dengan demikian, ketersediaan *driver* ODBC MySQL pada Tableau Desktop merupakan suatu keharusan. Oleh karena itu, dipilih MySQL Connector/ODBC versi 8.4.0 yang merupakan versi terbaru dari *driver* ODBC MySQL yang sesuai dengan versi MySQL Server yang digunakan, yaitu versi 8.4.3. Selanjutnya, pengunduhan *driver* ODBC MySQL diawali dengan mengakses halaman web “MySQL Product Archives – MySQL Connector/ODBC (Archived Versions)” pada situs web resmi MySQL (Developer Zone) yang dapat diakses melalui tautan <https://downloads.mysql.com/archives/c-odbc/>.

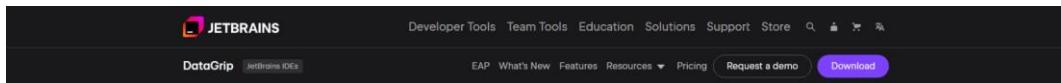


Gambar 3.6. Halaman Web “MySQL Product Archives – MySQL Connector/ODBC (Archived Versions)” pada Situs Web Resmi MySQL (Developer Zone)

Selanjutnya, dipilih versi 8.4.0 dan sistem operasi Windows (Microsoft Windows) pada *dropdown* yang tersedia. Setelah itu, dipilih berkas instalasi dalam bentuk MSI Installer dan mengeklik tombol “Download” yang sesuai untuk berkas instalasi tersebut untuk memulai pengunduhan.

d. Pengunduhan DataGrip

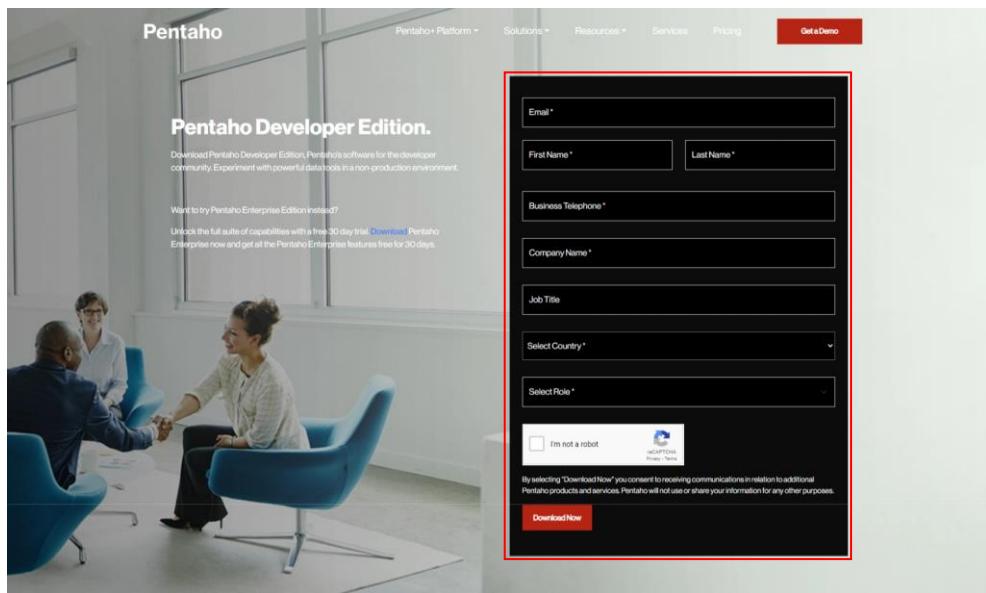
Proses pengunduhan DataGrip diawali dengan mengakses halaman web “Download Data Grip” pada situs web resmi JetBrains yang dapat diakses melalui tautan <https://www.jetbrains.com/datagrip/download/>. Kemudian, dipilih sistem operasi Windows dan berkas instalasi dengan ekstensi “.exe”, serta mengeklik tombol “Download” untuk memulai pengunduhan.



Gambar 3.7. Halaman Web “Download Data Grip” pada Situs Web Resmi JetBrains

3.1.3. Pengunduhan Pentaho Data Integration 9.4.0.0-343 (Developer/Community Edition)

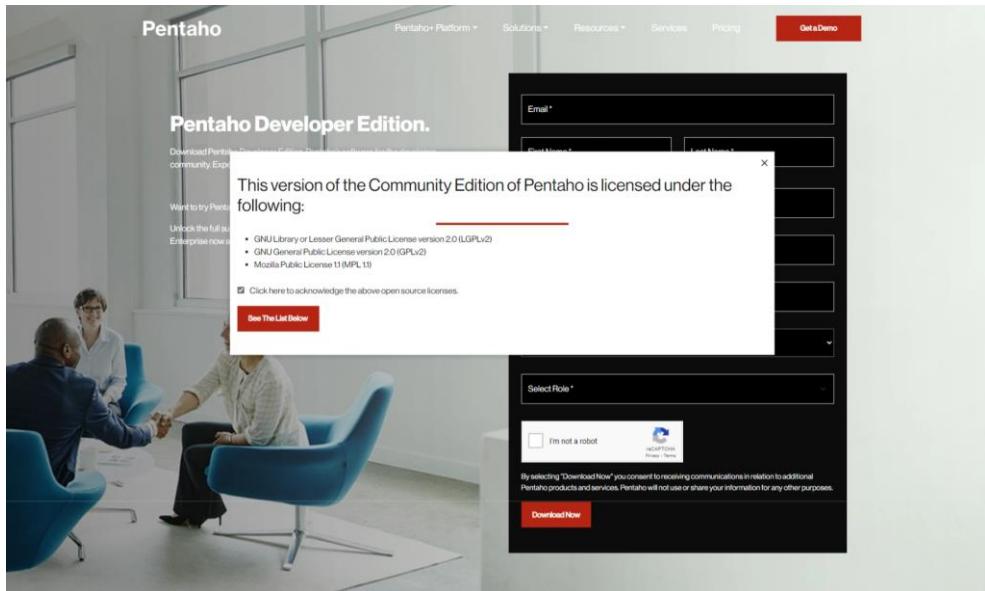
Proses pengunduhan Pentaho Data Integration (PDI) edisi pengembang (*developer*) atau komunitas (*community*) diawali dengan mengakses halaman web “Pentaho Developer Edition” pada situs web resmi Pentaho yang dapat diakses melalui tautan <https://pentaho.com/pentaho-developer-edition/>. Kemudian, melengkapi informasi pribadi pada setiap kolom (*field*) dalam *frame* yang memiliki tombol “Download Now” dan mengeklik tombol tersebut untuk mengakses daftar berkas instalasi Pentaho edisi pengembang (*developer*) atau komunitas (*community*) yang dapat diunduh dan digunakan secara gratis.



Gambar 3.8. Halaman Web “Pentaho Developer Edition” pada Situs Web Resmi Pentaho

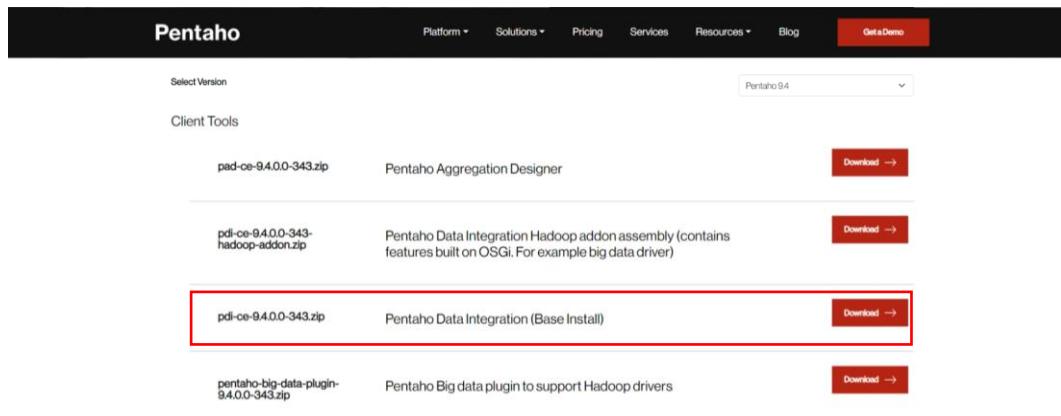
Setelah mengeklik tombol “Download Now”, diberikan informasi terkait lisensi dari Pentaho edisi pengembang (*developer*) atau komunitas (*community*) yang dapat dipahami lebih lanjut dengan mengeklik setiap hipertaut lisensi dalam daftar lisensi yang dicantumkan. Kemudian, dilakukan penyetujuan terhadap seluruh lisensi dengan mengeklik *check button* pada opsi “Click here to acknowledge the above open source

licenses” agar dapat mengeklik tombol “See The List Below” untuk mendapatkan daftar produk Pentaho edisi pengembang (*developer*) atau komunitas (*community*).



Gambar 3.9. Daftar Lisensi Produk Pentaho Edisi Pengembang (*Developer*) atau Komunitas (*Community*)

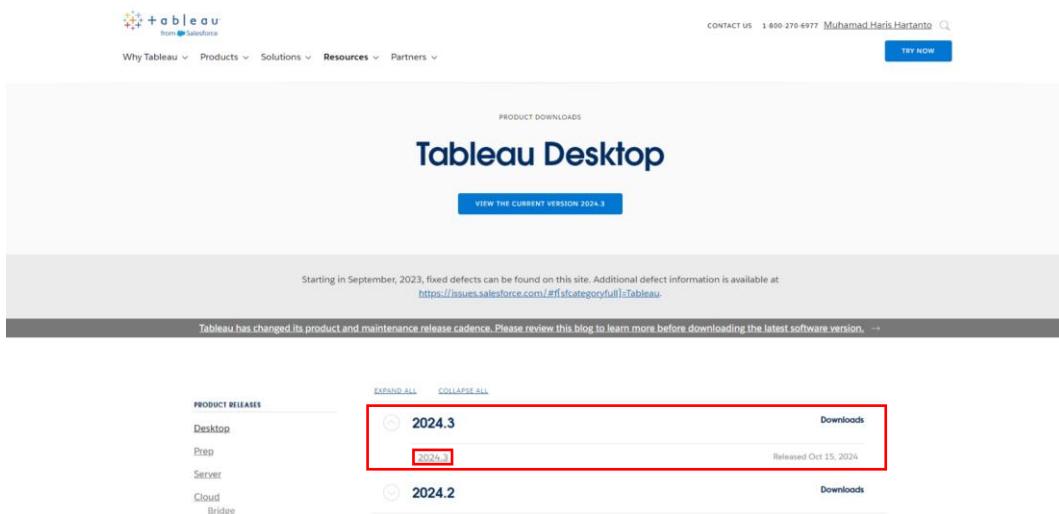
Dalam daftar produk yang diberikan, dipilih Pentaho versi 9.4 dan produk Pentaho Data Integration (Base Install) dengan nama berkas “pdi-ce-9.4.0.0-343.zip” pada daftar “Client Tools” dan mengeklik tombol “Download” yang mengakses tautan <https://privatefilesbucket-community-edition.s3.us-west-2.amazonaws.com/9.4.0.0-343/ce/client-tools/pdi-ce-9.4.0.0-343.zip> untuk memulai pengunduhan.



Gambar 3.10. Daftar Produk Pentaho Edisi Pengembang (*Developer*) atau Komunitas (*Community*) 9.4

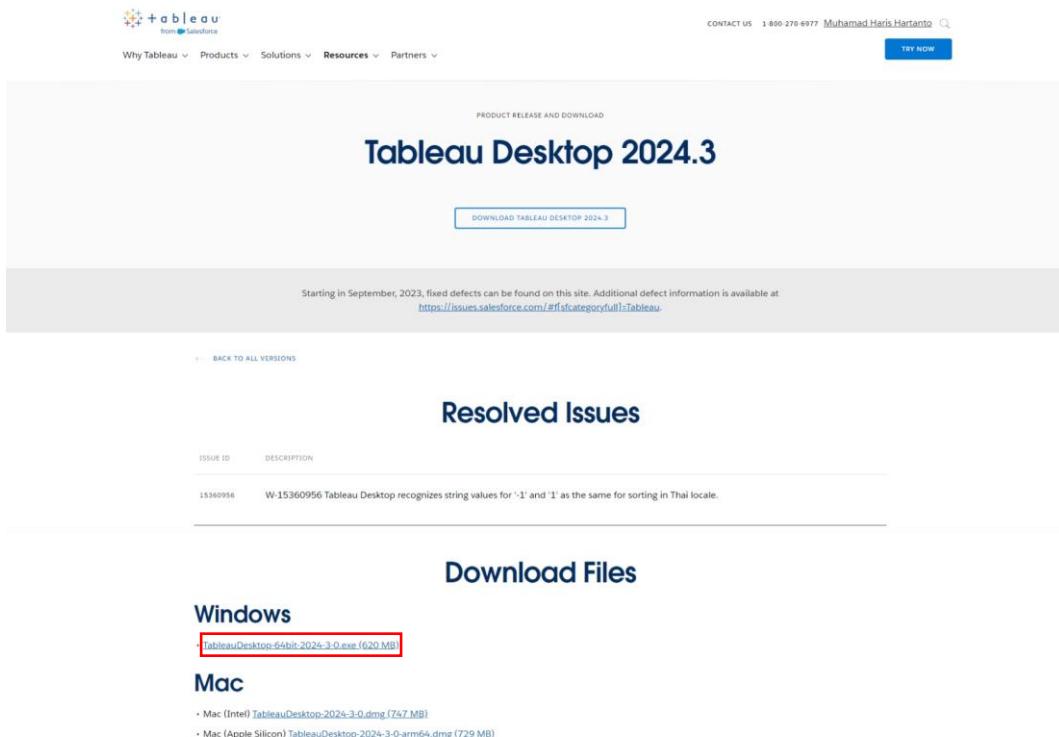
3.1.4. Pengunduhan Tableau Desktop 2024.3

Proses pengunduhan Tableau Desktop diawali dengan mengakses halaman web “PRODUCT DOWNLOADS – Tableau Desktop” pada situs web resmi Tableau yang dapat diakses melalui tautan <https://www.tableau.com/support/releases>. Kemudian, dipilih versi terbaru dari perangkat ini, yakni versi 2024.3.



Gambar 3.11. Halaman Web “PRODUCT DOWNLOADS – Tableau Desktop” pada Situs Web Resmi Tableau

Selanjutnya, pada halaman web “PRODUCT RELEASE AND DOWNLOAD - Tableau Desktop 2024.3”, dipilih berkas instalasi untuk sistem operasi Windows dengan mengeklik hipertaut yang terdapat dalam teks “TableauDesktop-64bit-2024-3-0.exe (620 MB)” untuk memulai pengunduhan.



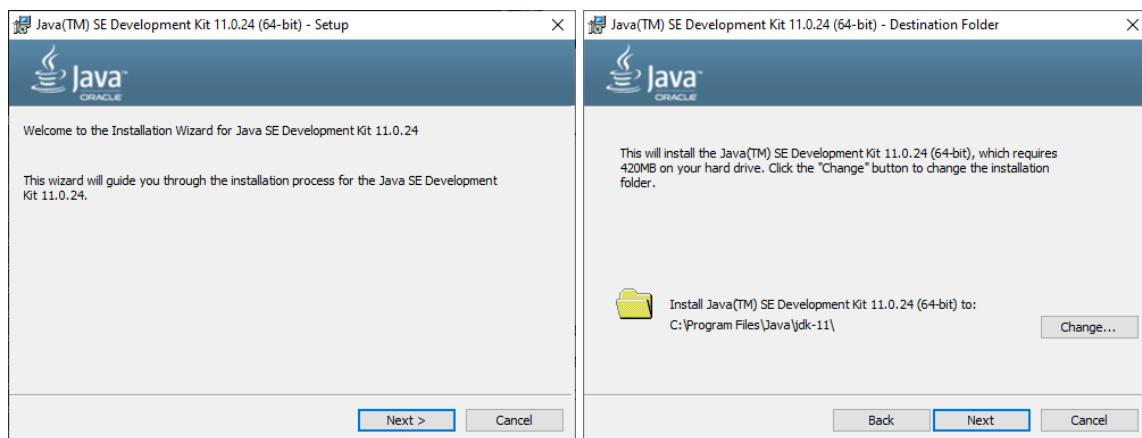
Gambar 3.12. Halaman Web “PRODUCT RELEASE AND DOWNLOAD - Tableau Desktop 2024.3” pada Situs Web Resmi Tableau

3.2. Instalasi *Software Data Warehouse*

3.2.1. Instalasi dan Konfigurasi Java Development Kit (JDK) 11

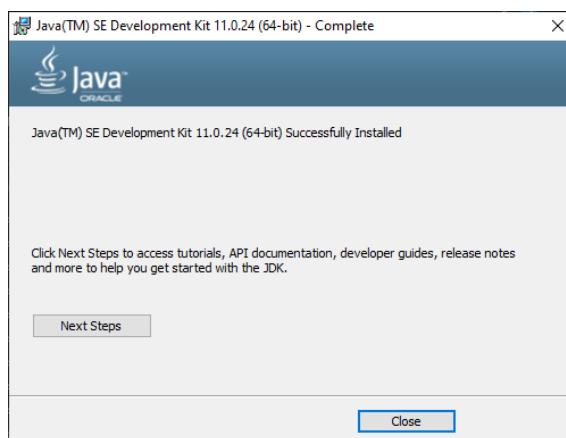
a. Instalasi Java Development Kit (JDK) 11

Proses instalasi JDK pada komputer dengan sistem operasi Windows menggunakan berkas instalasi (*installer*) dengan ekstensi “.exe” dimulai dengan membuka atau menjalankan berkas instalasi. Kemudian, mengeklik tombol “Next >” untuk menentukan lokasi penyimpanan JDK di dalam direktori komputer. Secara bawaan (*default*), JDK akan *diinstal* pada direktori “C:\Program Files\Java\jdk-11” yang membutuhkan alokasi ruang penyimpanan sebesar 420 MB pada *hard drive* komputer.



Gambar 3.13. Tahapan Pertama dan Kedua Instalasi JDK 11

Setelah lokasi instalasi ditentukan, proses instalasi dapat dimulai dengan mengeklik tombol “Next”. Apabila berhasil, akan ditampilkan halaman “Complete” yang menandakan bahwa JDK 11 telah terinstal dengan sukses.



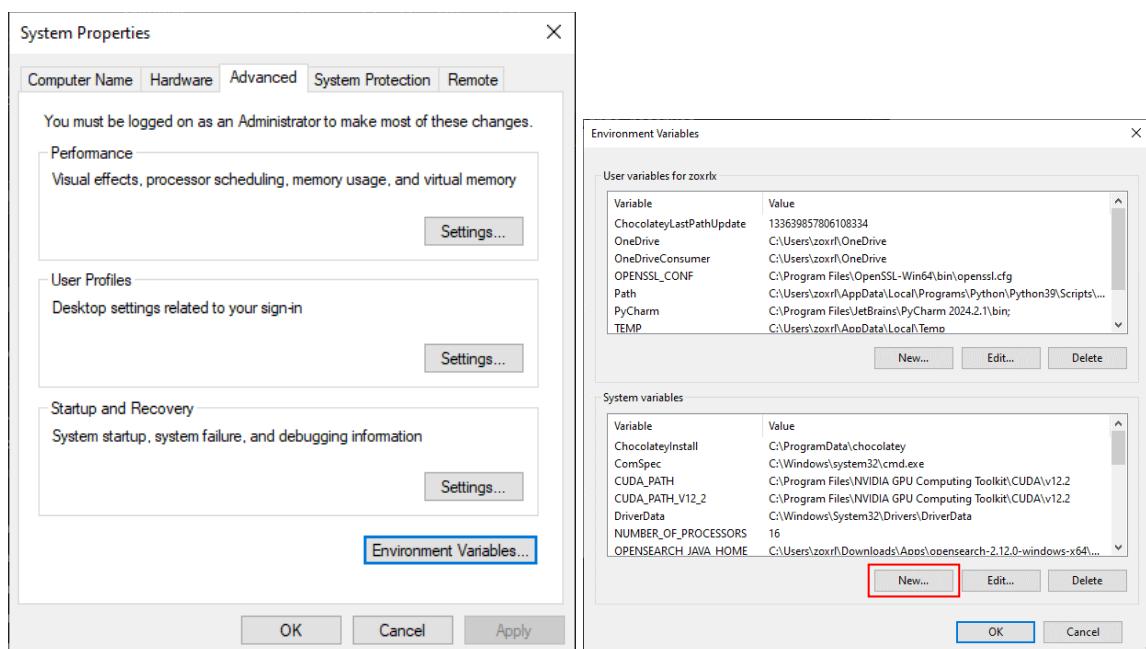
Gambar 3.14. Tahapan Terakhir Instalasi JDK 11

Pada halaman ini, juga diberikan informasi tambahan mengenai JDK 11, seperti dokumentasi *Application Programming Interface* (API), panduan pengembang, dan sebagainya yang dapat diakses melalui tombol “Next Steps”. Tombol ini akan mengarahkan pengguna ke halaman web “Java Platform Standard Edition 11 Documentation” pada situs resmi Dokumentasi Oracle

(<https://docs.oracle.com/en/java/javase/11/>). Proses instalasi diakhiri dengan mengeklik tombol “Close” yang akan menutup jendela (windows) instalasi JDK.

b. Konfigurasi *Environtment Variable* untuk JDK 11

Proses pengonfigurasian *environment variable* diawali dengan mengakses menu “Advanced” pada jendela pengaturan “System Properties”. Kemudian, mengakses jendela “Environment Variables” dengan mengeklik tombol “Environment Variables...” yang terdapat pada *dialog box*. Pada halaman *environment variable*, penambahan variabel (*variable*) untuk JDK 11 dilakukan dengan mengeklik tombol “New..” pada *frame* dengan label “System Variables” yang akan membuka halaman baru untuk menambahkan nama dan nilai (*value*) dari variabel yang akan ditambahkan.



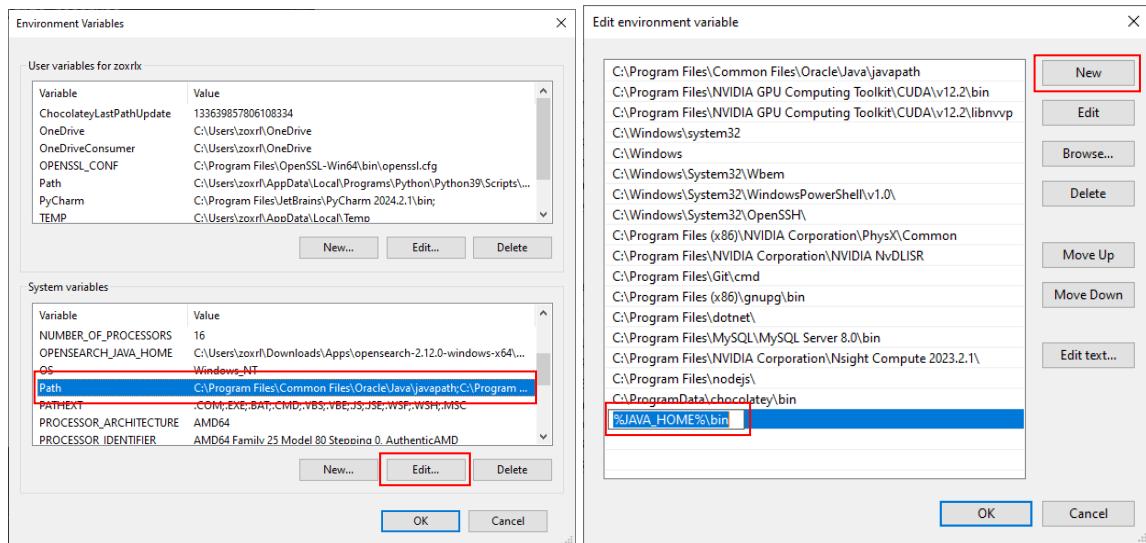
Gambar 3.15. Tahapan Pertama dan Kedua Pengonfigurasian JDK 11

Nilai nama variabel (*variable name*) diisi dengan nama JAVA_HOME, sementara nilai variabel (*variable value*) diisi dengan lokasi direktori JDK diinstal pada tahap sebelumnya, yaitu pada “C:\Program Files\Java\jdk-11”. Setelah itu, mengeklik tombol “OK” untuk menyelesaikan proses penambahan variabel baru.



Gambar 3.16. Tahapan Ketiga Pengonfigurasian JDK 11

Selanjutnya, dilakukan pembaruan pada variabel “Path” dalam *system variables* dengan mengeklik tombol “Edit...” pada *frame* dengan label “System Variables” yang akan membuka halaman baru untuk mengedit nilai (*value*) dari variabel “Path”.



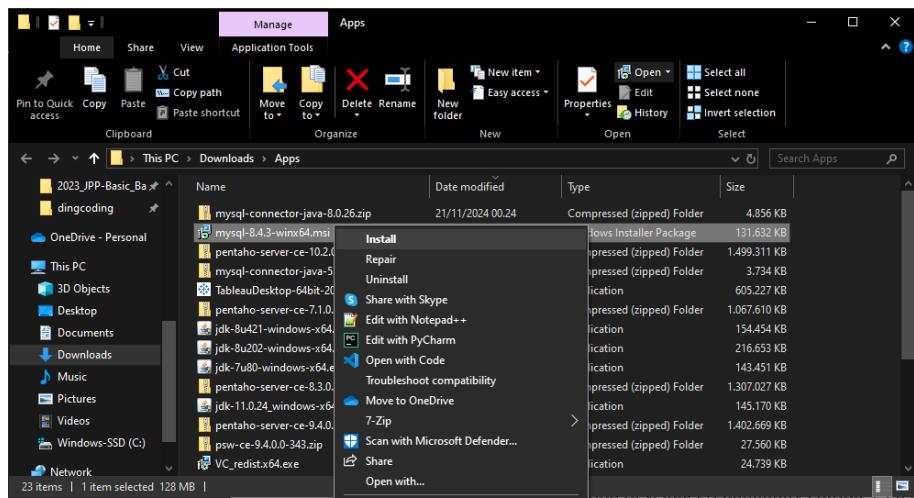
Gambar 3.17. Tahapan Keempat dan Terakhir Pengonfigurasian JDK 11

Ditambahkan lokasi direktori dari berkas-berkas yang dapat dijalankan Java (Java *executable files*) dengan mengeklik tombol “New” dan menambahkan nilai “%JAVA_HOME%\bin”. Setelah itu, mengeklik tombol “OK” untuk menyelesaikan proses penambahan nilai variabel baru dan mengeklik kembali tombol “OK” pada halaman *environment variable* untuk menyimpan seluruh aksi penambahan dan pembaruan variabel yang telah dilakukan, serta menutup jendela pengaturan “System Properties” untuk mengakhiri proses konfigurasi *environment variable* JDK.

3.2.2. Instalasi MySQL

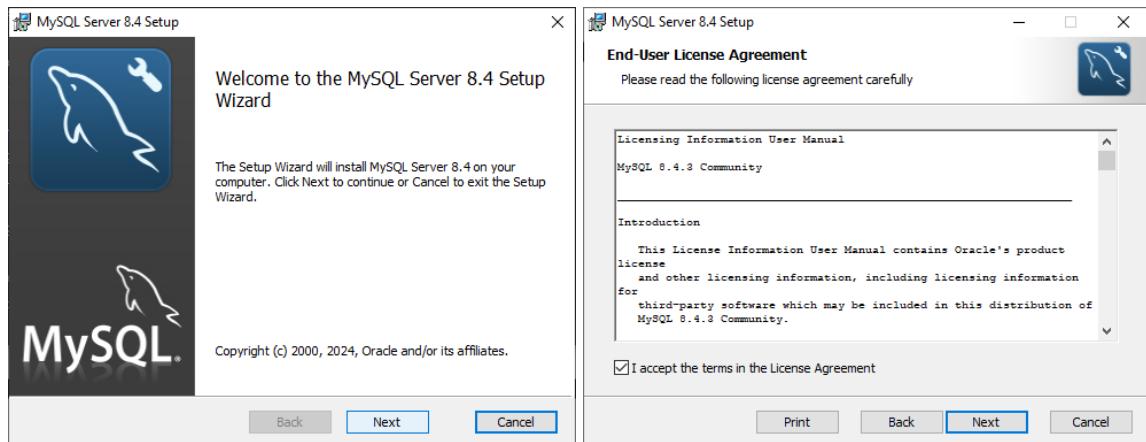
a. Instalasi MySQL Server 8.4.3 (*Community*)

Proses instalasi MySQL Server edisi komunitas (*community*) versi 8.4.3 pada komputer bersistem operasi Windows menggunakan berkas instalasi (*installer*) dalam bentuk MSI Installer yang berekstensi “.msi” diawali dengan membuka atau menjalankan berkas instalasinya.



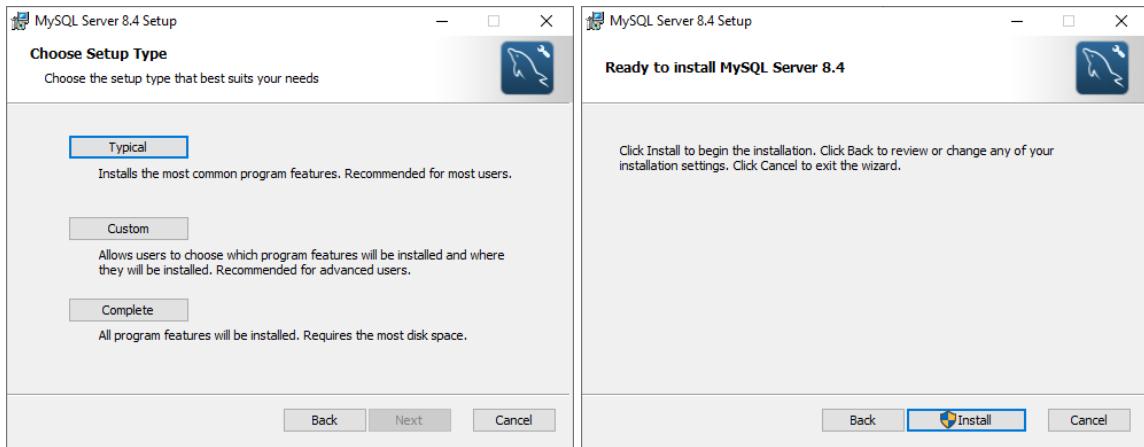
Gambar 3.18. Tahapan Pertama Instalasi MySQL Server 8.4.3 (*Community*)

Pada jendela awal instalasi MySQL Server yang ditampilkan, diklik tombol “Next” untuk mengakses dan membaca, serta menyetujui persetujuan lisensi dari MySQL Server 8.4.3 Community dengan mengeklik *check box* penyetujuan. Kemudian, mengeklik tombol “Next” untuk melanjutkan proses instalasi ke tahap pengaturan (*setup*) instalasi.



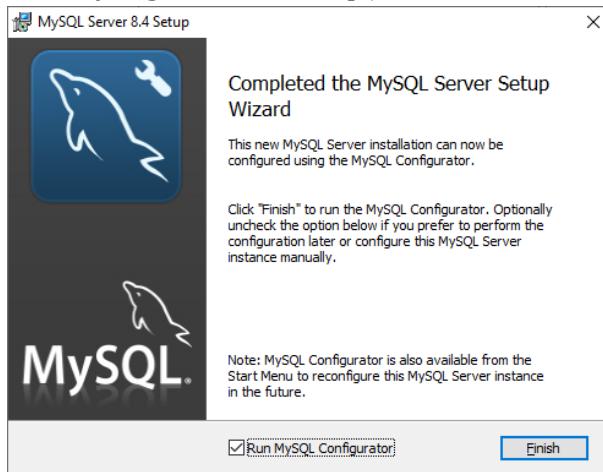
Gambar 3.19. Tahapan Kedua dan Ketiga Instalasi MySQL Server 8.4.3 (*Community*)

Pada halaman pengaturan (*setup*), dipilih opsi “Complete” yang akan menginstal seluruh fitur program dengan alokasi penyimpanan sebesar 831 MB. Kemudian, diklik tombol “Install” untuk memulai proses instalasi.



Gambar 3.20. Tahapan Keempat dan Kelima Instalasi MySQL Server 8.4.3 (Community)

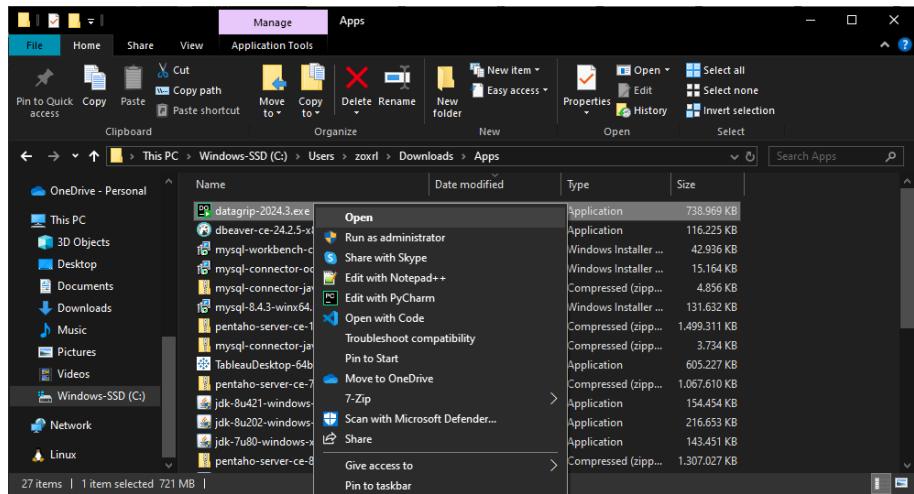
Ketika proses instalasi MySQL Server telah selesai, maka akan muncul jendela (*window*) seperti yang ditunjukkan pada gambar 3.21. Proses instalasi diakhiri dengan mengeklik tombol “Finish” yang akan menutup jendela instalasi MySQL Server.



Gambar 3.21. Tahapan Terakhir Instalasi MySQL Server 8.4.3 (Community)

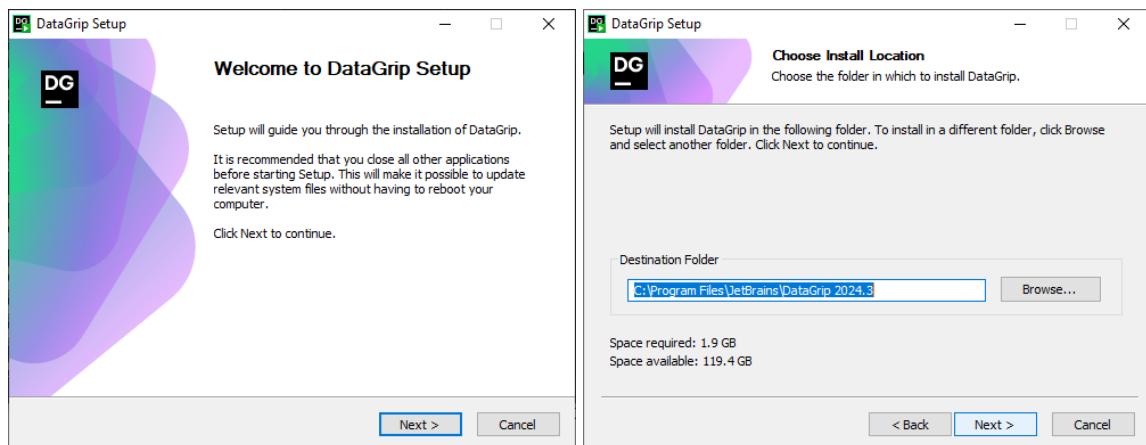
b. Instalasi DataGrip 2024.3

Proses instalasi DataGrip versi 2024.3 pada komputer bersistem operasi Windows menggunakan berkas instalasi (*installer*) yang berekstensi “.exe” diawali dengan melakukan membuka atau menjalankan berkas instalasi.



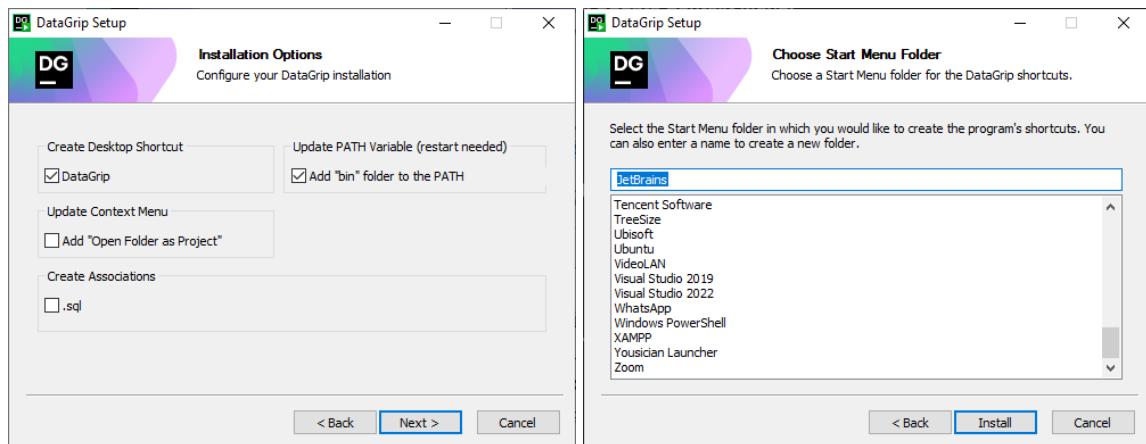
Gambar 3.22. Tahapan Pertama Instalasi DataGrip 2024.3

Selanjutnya, diklik tombol “Next >” untuk menentukan lokasi penyimpanan DataGrip di dalam direktori komputer. Secara bawaan (*default*), DataGrip akan diinstal pada direktori “C:\Program Files\JetBrains\DataGrip 2024.3”.



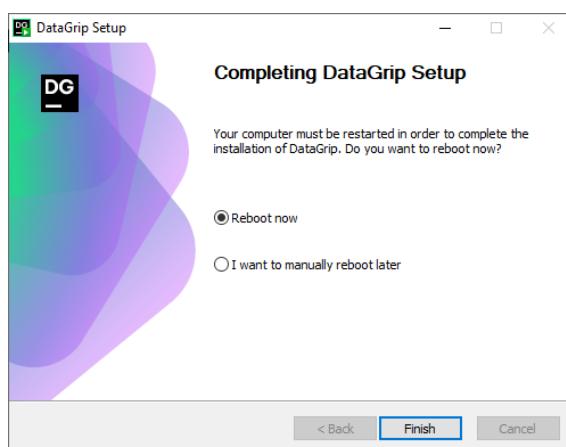
Gambar 3.23. Tahapan Kedua dan Ketiga Instalasi DataGrip 2024.3

Setelah lokasi instalasi ditentukan, diklik kembali tombol “Next >” untuk menentukan opsi instalasi, di mana dipilih opsi “Create Desktop Shortcut” untuk membuat *shortcut* DataGrip pada *desktop*. Selain itu, dipilih opsi “Update PATH Variable” untuk menambahkan folder “bin” dari direktori instalasi DataGrip ke variabel pengguna (*user*). Kemudian, diklik tombol “Next >” sekali lagi untuk memilih lokasi *shortcut* DataGrip di menu Start.



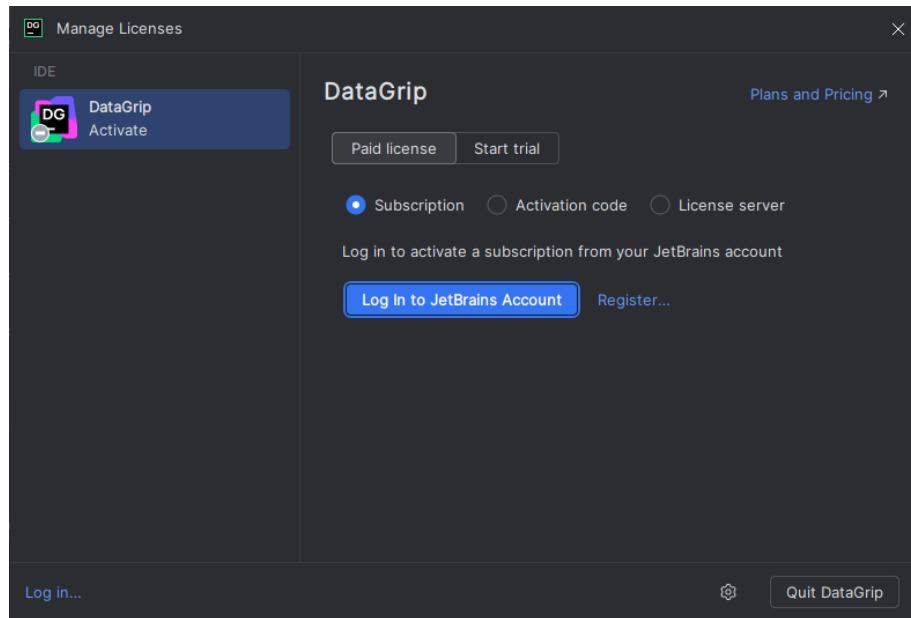
Gambar 3.24. Tahapan Keempat dan Kelima Instalasi DataGrip 2024.3

Apabila konfigurasi instalasi telah sesuai, proses instalasi dapat dimulai dengan mengeklik tombol “Install”. Kemudian, dipilih opsi “Reboot now” pada halaman “Completing DataGrip Setup” untuk melakukan *restart* komputer guna menyelesaikan proses instalasi.



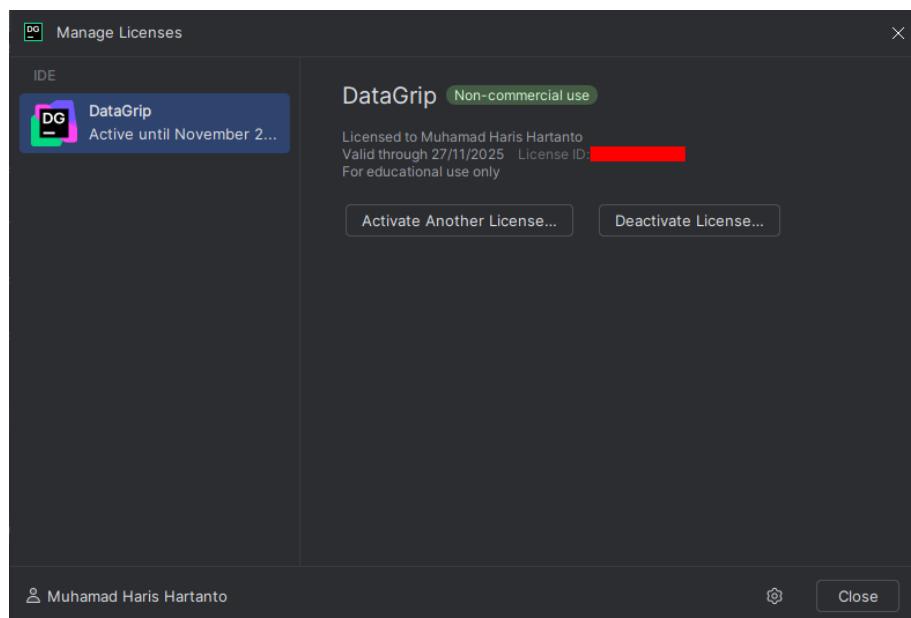
Gambar 3.25. Tahapan Terakhir Instalasi DataGrip 2024.3

Setelah instalasi selesai, DataGrip dapat diakses melalui *shortcut* yang terdapat pada *desktop* ataupun direktori “JetBrains” pada menu Start. Ketika pertama kali membuka DataGrip, akan diarahkan ke halaman “Manage Licenses” untuk mengaktifkan lisensi dengan tiga opsi, yakni langganan (*subscription*) yang tertaut dengan akun JetBrains, kode aktivasi, dan server. Selain itu, juga tersedia opsi *trial* gratis selama 30 hari yang dapat dipilih jika belum memiliki lisensi. Pada halaman ini, dipilih proses aktivasi lisensi melalui opsi langganan dengan mengeklik tombol “Log In to JetBrains” untuk masuk ke akun JetBrains melalui *browser* yang akan terbuka secara otomatis.



Gambar 3.26. Opsi Aktivasi Lisensi DataGrip 2024.3

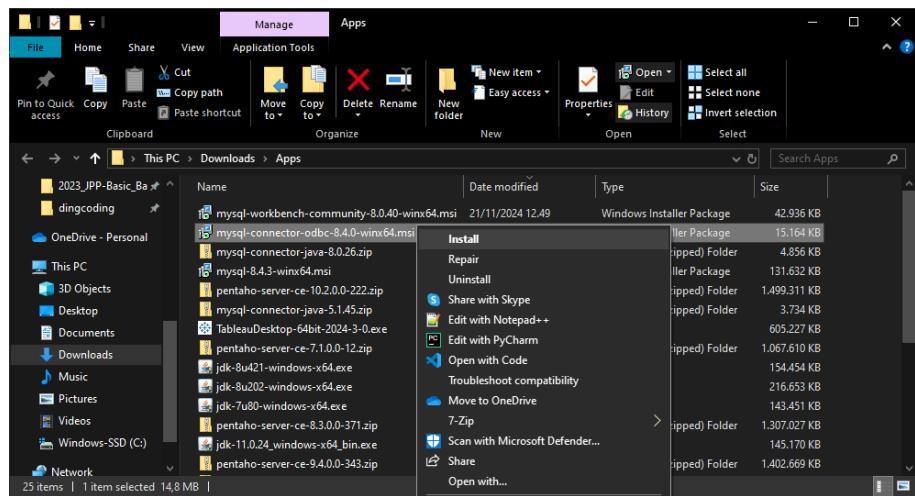
Setelah sukses masuk ke akun JetBrains pada *browser*, maka status lisensi pada halaman “Manage Licenses” akan secara otomatis aktif.



Gambar 3.27. Informasi DataGrip 2024.3 dengan Lisensi yang Telah Aktif

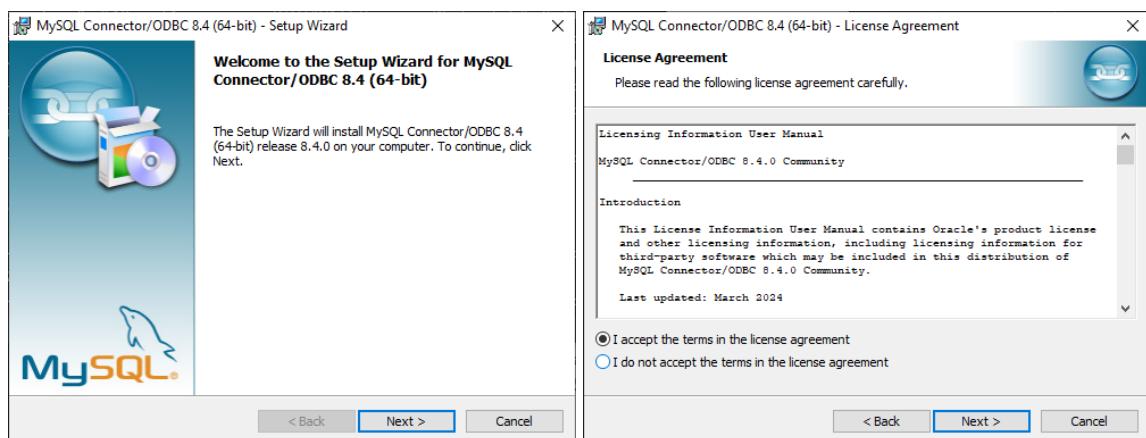
c. Instalasi *Driver ODBC MySQL 8.4.0*

Proses instalasi *driver ODBC MySQL* versi 8.4.0 pada komputer dengan sistem operasi Windows menggunakan berkas instalasi (*installer*) dalam bentuk MSI Installer dengan berkas yang berekstensi “.msi” diawali dengan melakukan membuka atau menjalankan berkas instalasi.



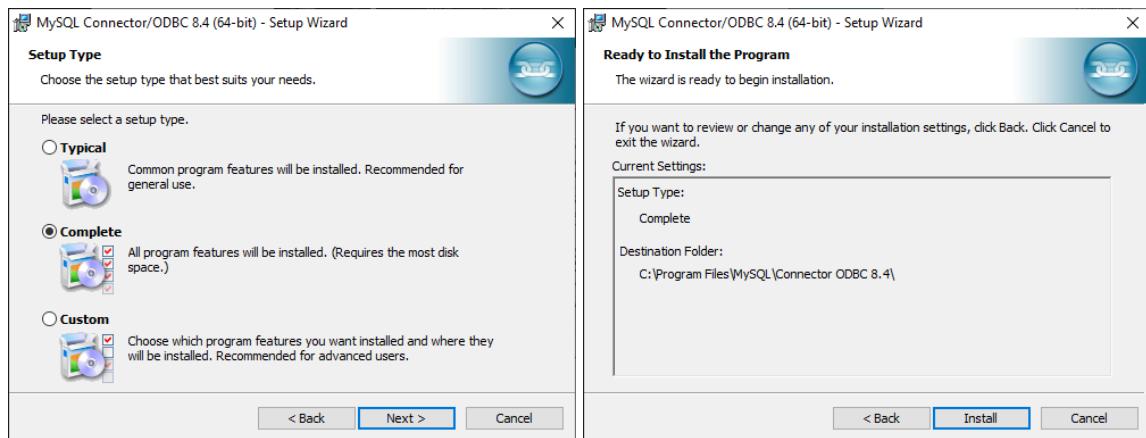
Gambar 3.28. Tahapan Pertama Instalasi *Driver ODBC MySQL 8.4.0*

Selanjutnya, diklik tombol “Next” untuk mengakses dan membaca, serta menyetujui persetujuan lisensi dari MySQL Connector/ODBC 8.4.



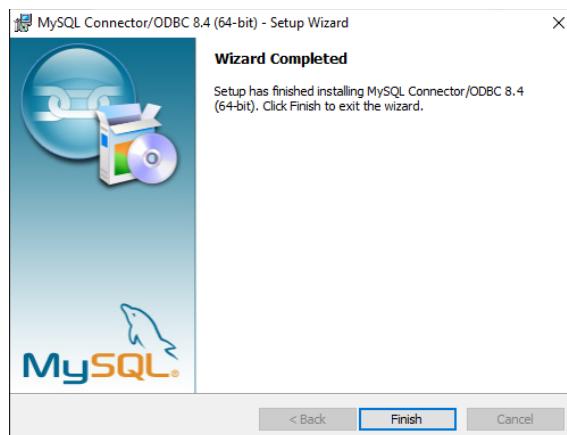
Gambar 3.29. Tahapan Kedua dan Ketiga Instalasi *Driver ODBC MySQL 8.4.0*

Setelah itu, diklik kembali tombol “Next >” untuk menentukan *setup type*, di mana dipilih opsi “Complete” untuk menginstal seluruh fitur program yang tersedia. Kemudian, diklik tombol “Next >” sekali lagi untuk mengakses halaman peninjauan instalasi yang memberikan informasi mengenai konfigurasi instalasi yang telah diatur sebelumnya.



Gambar 3.30. Tahapan Keempat dan Kelima Instalasi *Driver ODBC MySQL 8.4.0*

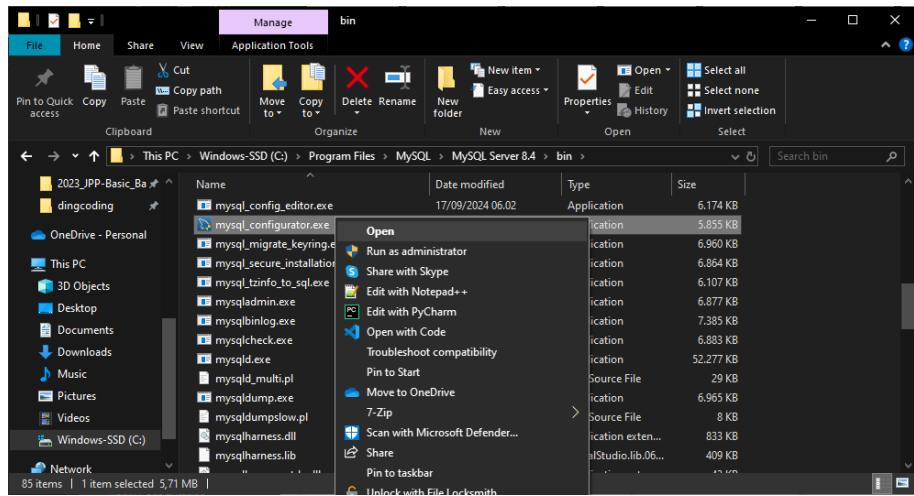
Apabila konfigurasi instalasi telah sesuai, proses instalasi dapat dimulai dengan mengeklik tombol “Install”. Kemudian, ditampilkan halaman “Wizard Complete” yang menandakan bahwa *driver ODBC MySQL* atau *MySQL Connector/ODBC* telah terinstal dengan sukses. Proses instalasi diakhiri dengan mengeklik tombol “Finish” yang akan menutup jendela (*window*) instalasi *MySQL Connector/ODBC*.



Gambar 3.31. Tahapan Terakhir Instalasi Driver ODBC MySQL 8.4.0

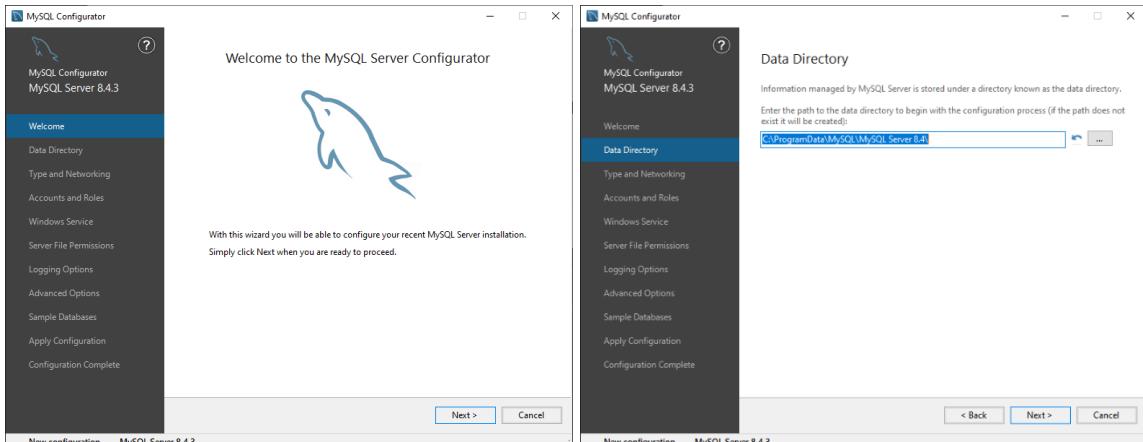
d. Konfigurasi MySQL Server 8.4.3 (*Community*)

Proses pengonfigurasian MySQL Server edisi komunitas (*community*) versi 8.4.3 pada komputer bersistem operasi Windows diawali dengan mengakses MySQL Configurator melalui *shortcut* yang terdapat dalam menu Start atau menjalankan berkas “mysql_configurator.exe” dalam direktori “bin” yang dapat ditemukan di dalam lokasi direktori MySQL Server diinstal.



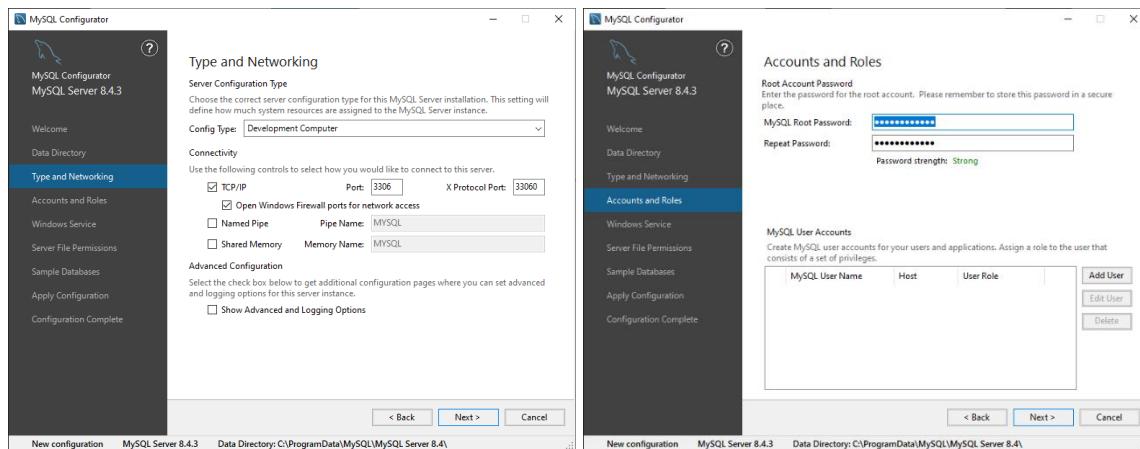
Gambar 3.32. Tahapan Pertama Pengonfigurasian MySQL Server 8.4.3 (Community)

Pada jendela (*window*) awal instalasi MySQL Server yang ditampilkan, diklik tombol “Next >” untuk memulai serangkaian tahapan pengonfigurasian yang diawali dengan tahap “Data Directory” yang menentukan lokasi direktori data. Dalam tahap pertama ini, diinisialisasikan jalur direktori data yang secara bawaan (*default*) berupa “C:\ProgramData\MySQL\MySQL Server 8.4\”. Kemudian, diklik kembali tombol “Next >” untuk melanjutkan pengonfigurasian ke tahapan selanjutnya.



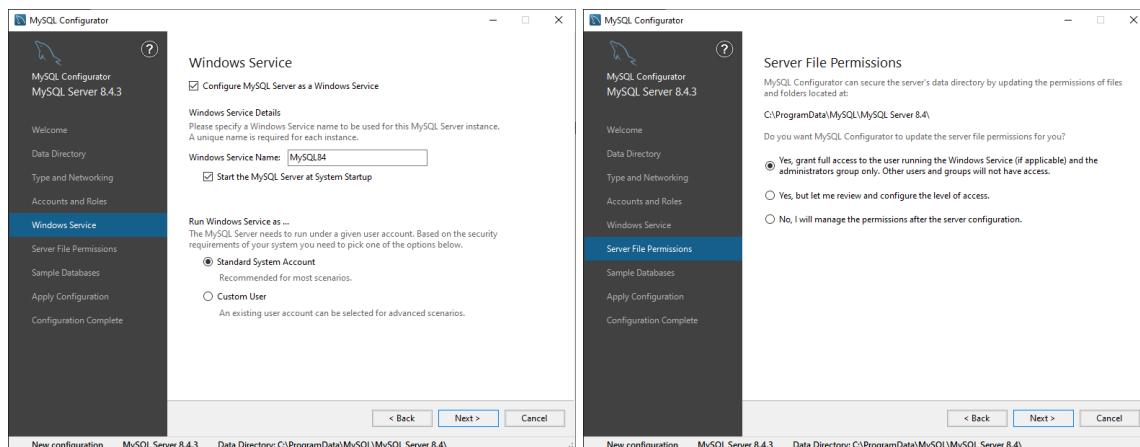
Gambar 3.33. Tahapan Kedua dan Ketiga Pengonfigurasian MySQL Server 8.4.3 (Community)

Pada tahap yang kedua, yakni “Type and Networking” yang mengatur tipe konfigurasi server. Dipilih tipe konfigurasi (*config type*) yang berupa “Development Computer” dan konektivitas (*connectivity*) seperti yang ditunjukkan pada gambar 3.34. Kemudian, diklik tombol “Next >” untuk melanjutkan ke tahap “Account and Roles” yang mengatur pembuatan kata sandi (*password*) untuk pengguna (*user*) “Root” MySQL Server. Setelah itu, diklik kembali tombol “Next >” untuk melanjutkan pengonfigurasian ke tahapan selanjutnya.



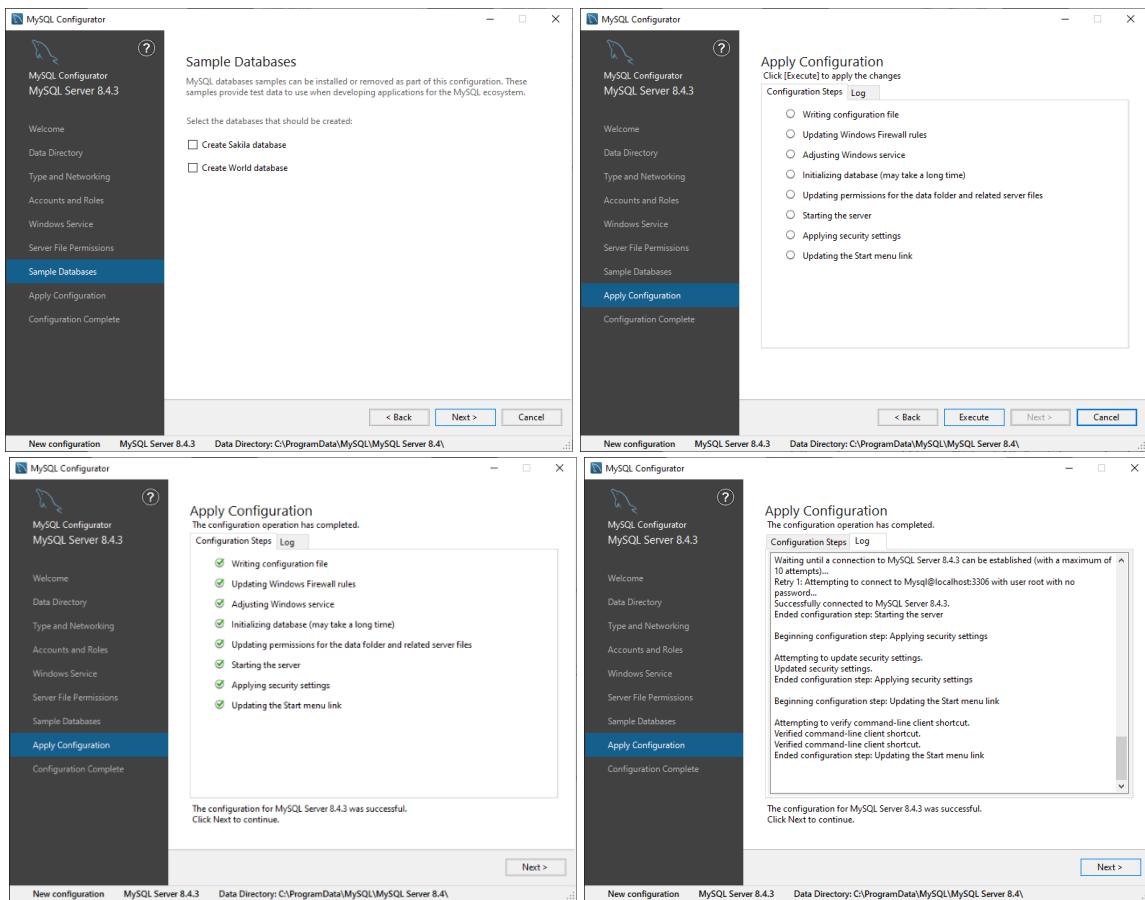
Gambar 3.34. Tahapan Keempat dan Kelima Pengonfigurasian MySQL Server 8.4.3 (*Community*)

Pada tahap “Windows Service”, diinisialisasi nama *instance* MySQL Server dalam layanan sistem operasi komputer (Windows service) yang secara bawaan diberi nama dengan format “MySQL<versi>”, seperti MySQL84. Kemudian, dipilih opsi “Standard System Account” pada pengaturan “Run Windows Service as ...”. Setelah itu, mengeklik tombol “Next >” untuk melanjutkan ke tahap “Server File Permissions” yang mengatur izin akses berkas server. Pada tahap ini, diberikan akses penuh terhadap semua pengguna (user) dan grup administrator yang menjalankan layanan Windows (Windows service) dengan mengeklik pilihan pertama yang tersedia. Setelah itu, diklik kembali tombol “Next >” untuk melanjutkan pengonfigurasian ke tahapan selanjutnya.



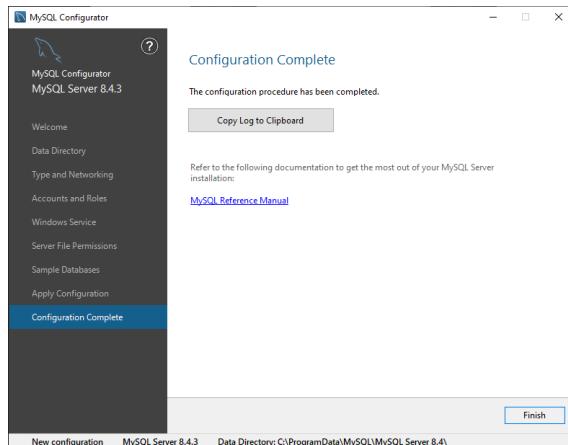
Gambar 3.35. Tahapan Keenam dan Ketujuh Pengonfigurasian MySQL Server 8.4.3 (*Community*)

Pada tahap “Sample Databases”, diberikan opsi contoh/sampel basis data yang ingin dibentuk ataupun tidak dengan langsung mengklik tombol “Next >” untuk melewati pembentukan basis data. Kemudian, pengonfigurasian yang telah diatur dalam serangkaian tahapan sebelumnya dapat diaplikasikan dengan mengeklik tombol “Execute” pada tahap “Apply Configuration”.



Gambar 3.36. Tahapan Kedelapan dan Kesembilan Pengonfigurasian MySQL Server 8.4.3 (*Community*)

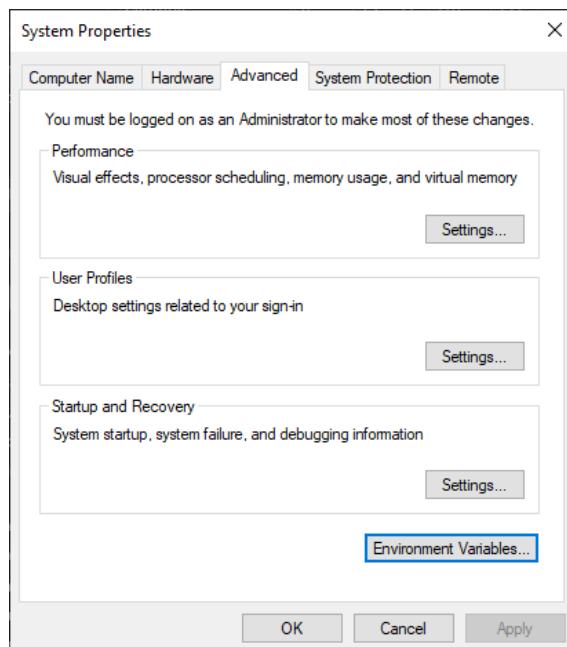
Dalam tahap “Apply Configuration” ini, juga terdapat *log* yang mencatat aktivitas dari serangkaian proses pengaplikasian konfigurasi MySQL Server yang dapat diakses melalui tab “Logs”. Apabila proses pengaplikasian konfigurasi telah berhasil dijalankan, diklik tombol “Next >” untuk mengakses jendela (*window*) terakhir “Configuration Complete” yang menyatakan bahwa proses konfigurasi telah selesai. Dalam tahap ini, diberikan opsi untuk menyalin *log* dari pengaplikasian konfigurasi MySQL Server dengan mengeklik tombol “Copy Log to Clipboard”. Selain itu, juga diberikan informasi mengenai dokumentasi resmi dari MySQL Server yang dapat diakses melalui hipertaut “MySQL Reference Manual” yang akan mengarahkan pengguna ke halaman web “MySQL 8.4 Reference Manual” pada situs resmi MySQL (Developer Zone) yang dapat diakses melalui tautan <https://dev.mysql.com/doc/refman/8.4/en/>. Proses instalasi diakhiri dengan mengeklik tombol “Finish” yang akan menutup jendela (*windows*) MySQL Configurator.



Gambar 3.37. Tahapan Terakhir Pengonfigurasian MySQL Server 8.4.3 (*Community*)

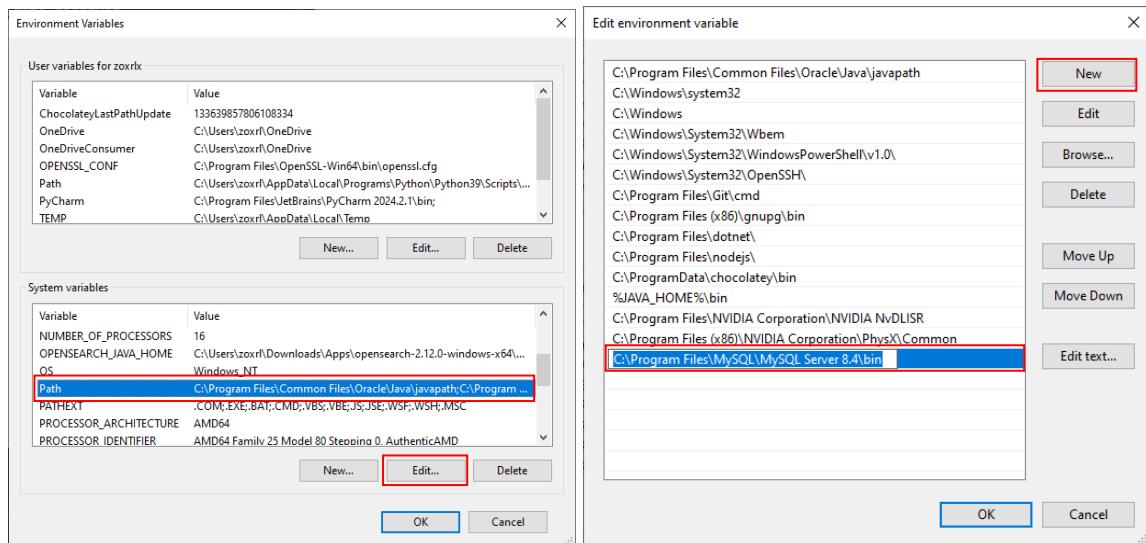
e. Konfigurasi *Environment Variable* untuk MySQL Server 8.4.3

Proses pengonfigurasian *environment variable* diawali dengan mengakses menu “Advanced” pada jendela pengaturan “System Properties”. Kemudian, mengakses jendela “Environment Variables” dengan mengeklik tombol “Environment Variables...” yang terdapat pada *dialog box*.



Gambar 3.38. Tahapan Pertama Pengonfigurasian *Environment Variable* untuk MySQL Server 8.4.3

Pada halaman *environment variable*, penambahan variabel (*variable*) untuk MySQL dilakukan dengan memilih variabel “Path” dalam *system variables* dan mengeklik tombol “Edit...” pada *frame* dengan label “System Variables” untuk mengakses halaman “Edit environment variable” guna mengedit nilai dari variabel “Path”. Kemudian, ditambahkan lokasi direktori berkas-berkas yang dapat dijalankan MySQL dengan mengeklik tombol “New” dan menambahkan nilai yang berupa lokasi direktori “bin” MySQL, yakni “C:\Program Files\MySQL\MySQL Server 8.4\bin”.



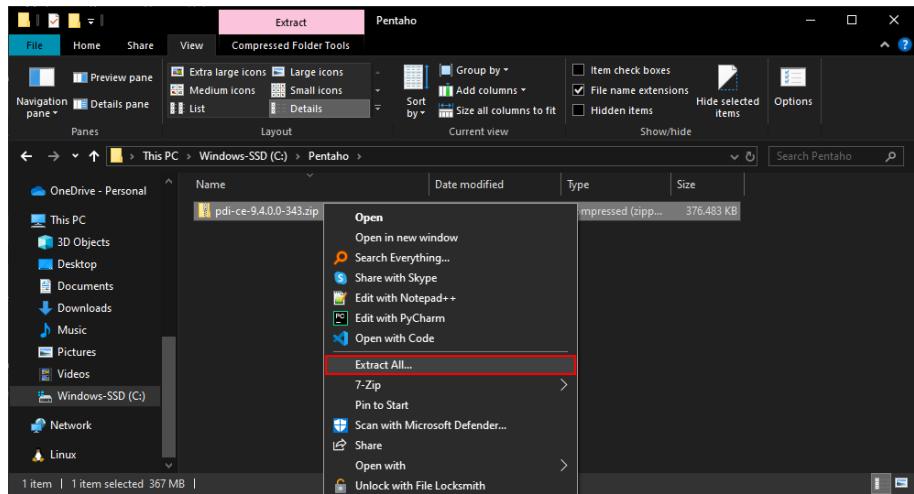
Gambar 3.39. Tahapan Kedua dan Terakhir Pengonfigurasian *Environment Variable* untuk MySQL Server 8.4.3

Setelah itu, mengeklik tombol “OK” untuk menyelesaikan proses penambahan nilai variabel baru dan mengeklik kembali tombol “OK” pada halaman “Environment Variables” untuk menyimpan seluruh aksi penambahan dan pembaruan variabel yang telah dilakukan, serta menutup jendela pengaturan “System Properties” untuk mengakhiri proses konfigurasi *environment variable* untuk MySQL Server.

3.2.3. Instalasi Pentaho Data Integration (PDI) 9.4.0.0-343 (Developer/Community Edition)

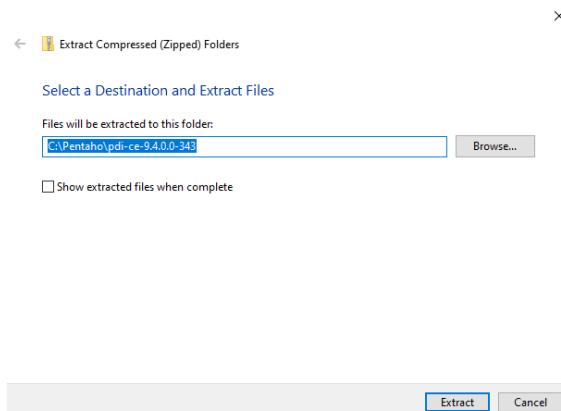
a. Instalasi Pentaho Data Integration (PDI)

Proses instalasi PDI 9.4 edisi pengembang (*developer*) atau komunitas (*community*) pada komputer dengan sistem operasi Windows menggunakan berkas instalasi dalam bentuk *compressed archive* dengan berkas berekstensi “.zip” diawali dengan mengekstrak berkas instalasi tersebut. Proses ekstraksi dilakukan dengan melakukan *secondary click* atau mengeklik kanan (*right-click*) berkas instalasi untuk membuka menu konteks (*context menu*). Kemudian, mengeklik opsi “Extract All...” untuk memulai ekstraksi berkas yang secara otomatis akan menampilkan *dialog box* “Extract Compressed (Zipped) Folders”.



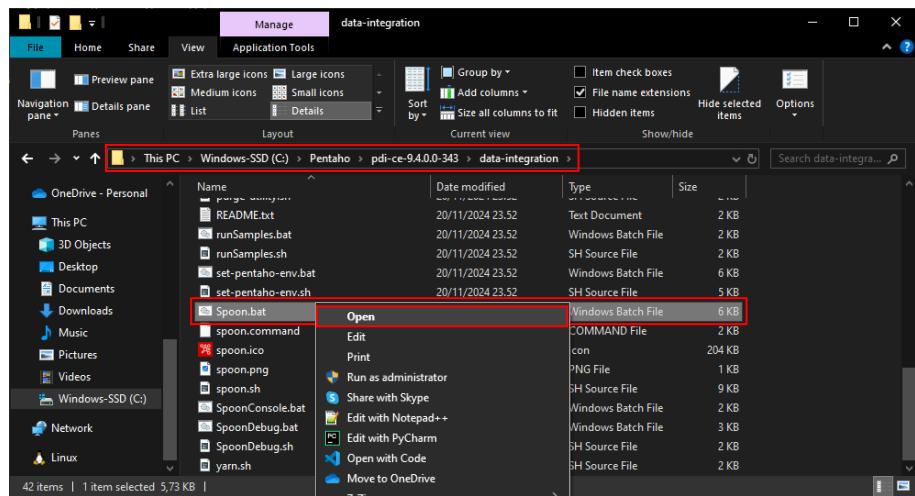
Gambar 3.40. Tahapan Pertama Instalasi PDI 9.4

Kemudian, menentukan lokasi direktori untuk menyimpan hasil ekstraksi berkas. Secara bawaan (*default*), lokasi direktori diinisialisasi di tempat yang sama dengan berkas *compressed archive* yang akan diekstrak. Proses ekstraksi kemudian dijalankan dengan mengeklik tombol “Extract”.



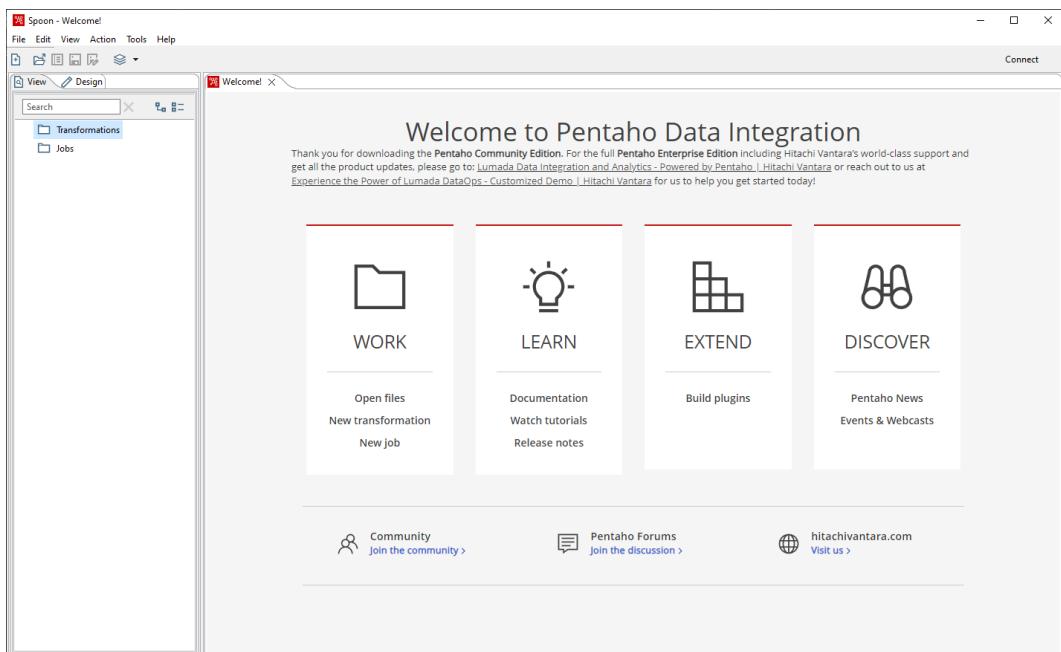
Gambar 3.41. Tahapan Terakhir Instalasi PDI 9.4

Setelah proses ekstraksi berkas instalasi berhasil dilakukan, akan muncul direktori atau folder baru dengan nama yang sama dengan berkas instalasi dalam lokasi direktori yang telah ditentukan sebelumnya.



Gambar 3.42. Cara Mengakses PDI 9.4

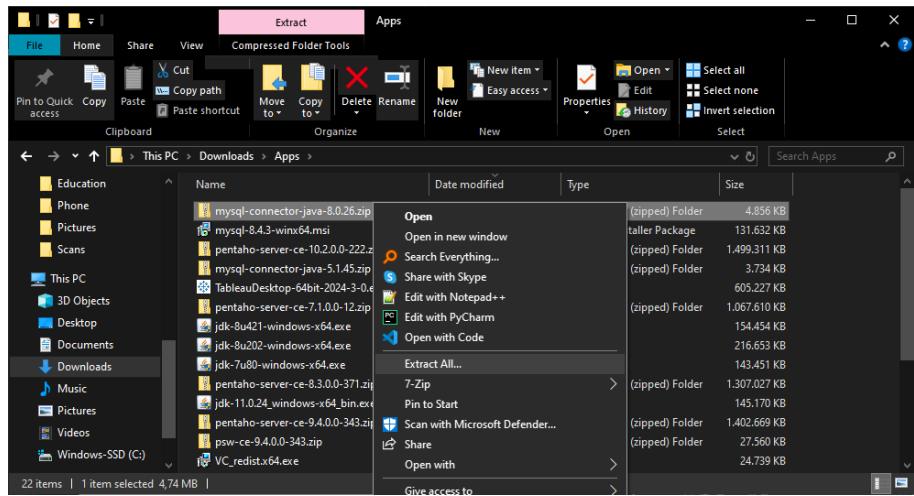
Pentaho Data Integration dapat dioperasikan dengan mengeksekusi atau membuka berkas “Spoon.bat” dalam direktori “data-integration” yang dapat ditemukan di dalam direktori hasil ekstraksi berkas instalasi.



Gambar 3.43. Tampilan Awal PDI 9.4

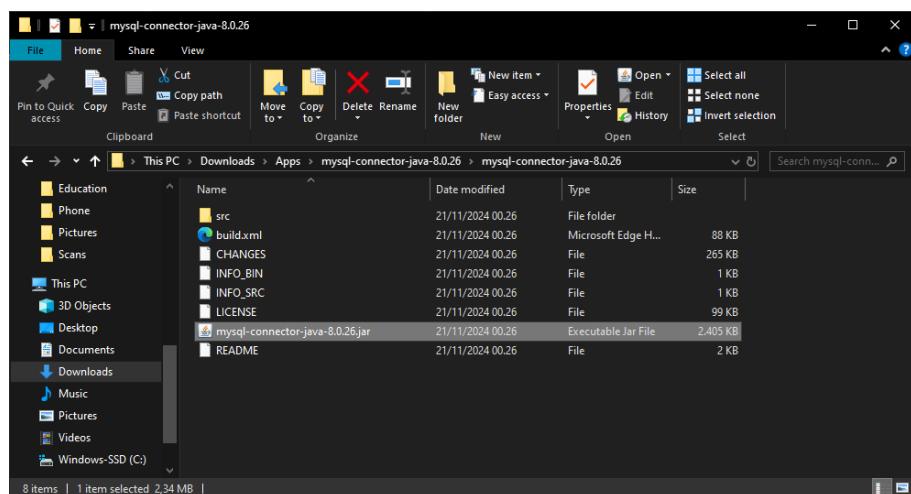
b. Pemasangan *Driver JDBC MySQL* pada Pentaho Data Integration (PDI) 9.4.0.0-343 (Developer/Community Edition)

Proses pemasangan *driver JDBC MySQL* pada PDI 9.4 edisi pengembang (*developer*) atau komunitas (*community*) dimulai dengan mengakses lokasi berkas MySQL Connector/J yang telah diunduh. Kemudian, berkas tersebut diekstrak dengan menggunakan fitur ekstraksi yang tersedia dalam sistem operasi.



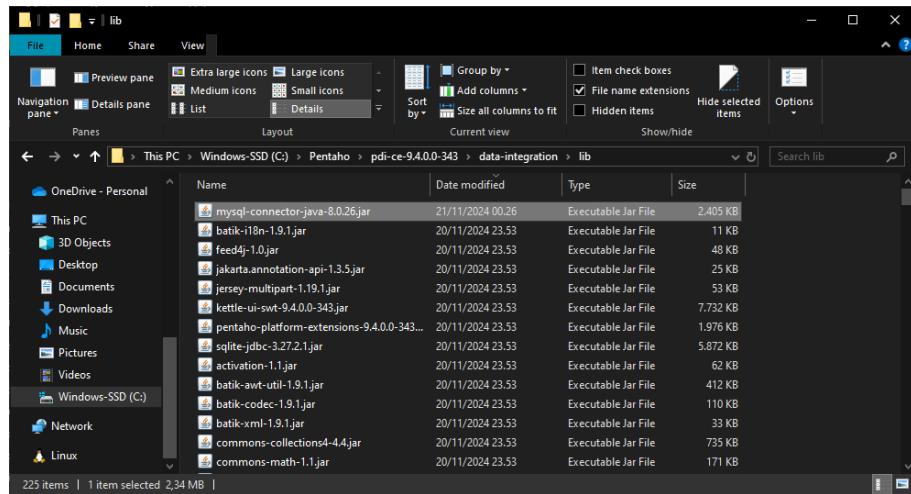
Gambar 3.44. Tahapan Pertama Pemasangan Driver JDBC MySQL pada PDI 9.4 (Developer/Community Edition)

Selanjutnya, dilakukan penyalinan (*copy*) terhadap berkas “mysql-connector-java-8.0.26.jar” yang terdapat di dalam direktori hasil ekstraksi berkas MySQL Connector/J.



Gambar 3.45. Tahapan Kedua Pemasangan Driver JDBC MySQL pada PDI 9.4 (Developer/Community Edition)

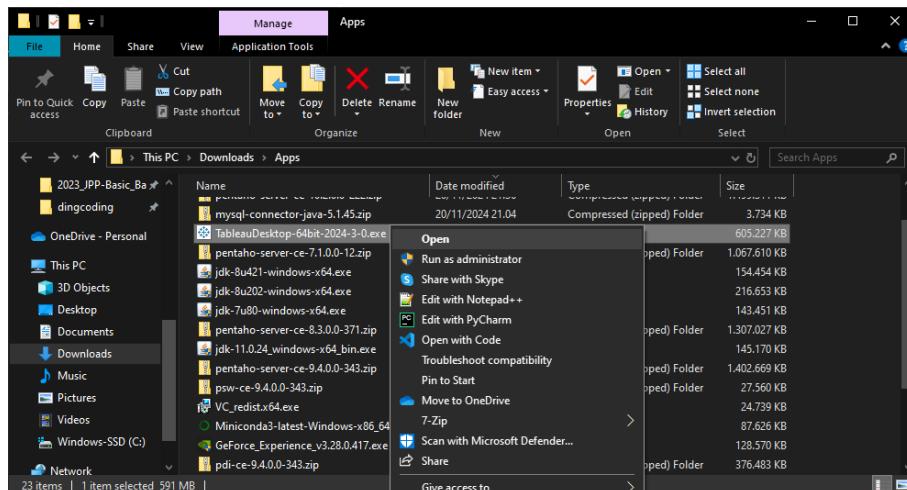
Setelah itu, berkas “mysql-connector-java-8.0.26.jar” yang telah disalin (*copy*), ditempelkan (*paste*) ke direktori “lib” Pentaho Data Integration (PDI) yang terdapat pada jalur direktori “...\\pdi-ce-9.4.0.0-343\\data-integration\\lib”.



Gambar 3.46. Tahapan Terakhir Pemasangan *Driver JDBC MySQL* pada PDI 9.4 (*Developer/Community Edition*)

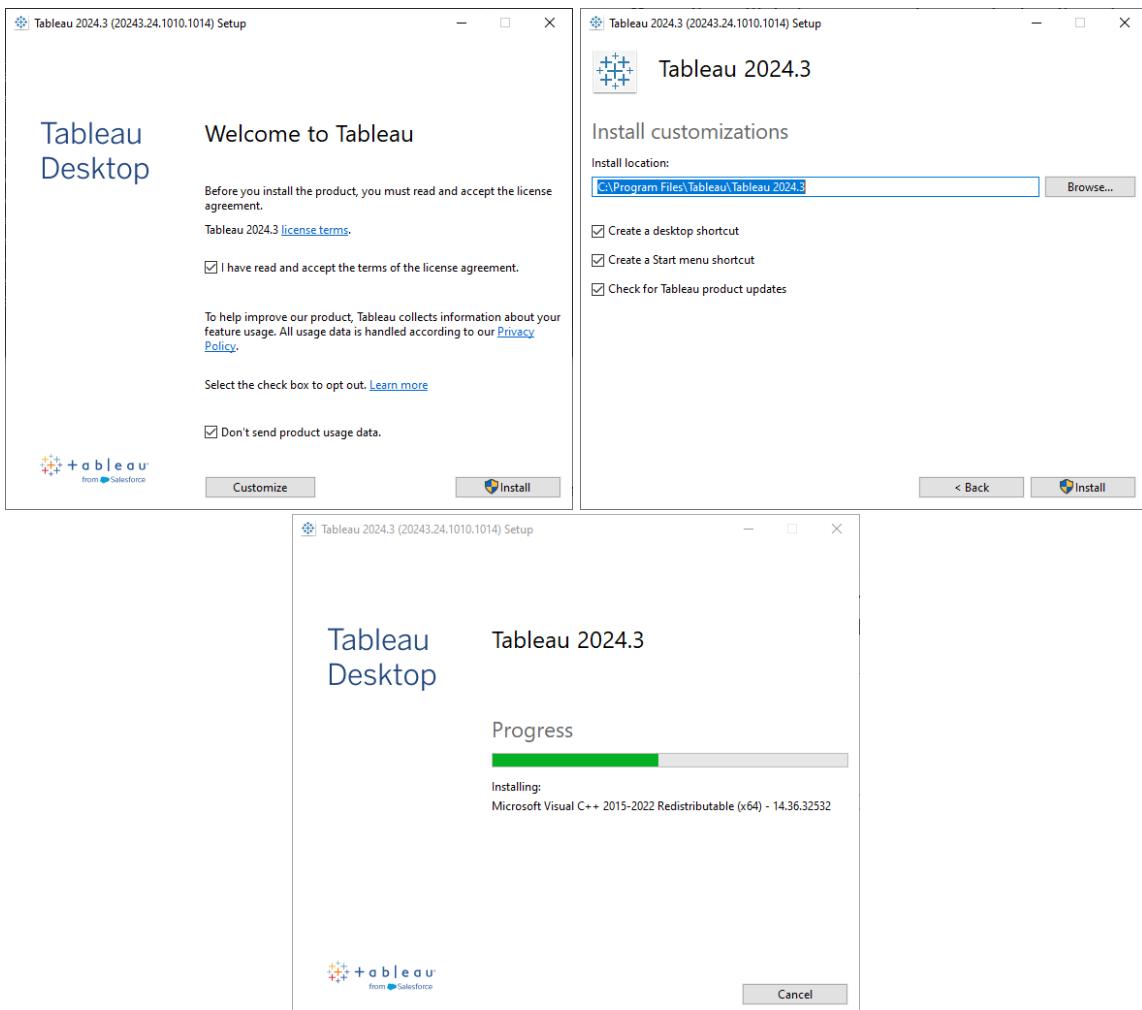
3.2.4. Instalasi Tableau Desktop 2024.3

Proses instalasi Tableau Desktop 2024.3 pada komputer dengan sistem operasi Windows menggunakan berkas instalasi (*installer*) dengan ekstensi “.exe” dimulai dengan membuka atau menjalankan berkas instalasi.



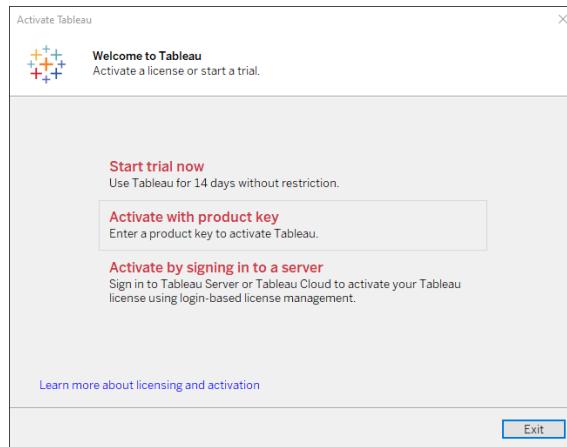
Gambar 3.47. Tahapan Pertama Instalasi Tableau Desktop 2024.3

Selanjutnya, diberikan informasi mengenai persetujuan persyaratan/ketentuan lisensi dari Tableau Desktop yang dapat dibaca lebih lanjut dengan mengeklik hipertaut yang terdapat dalam teks “license agreement”. Kemudian, dilakukan penyetujuan terhadap lisensi tersebut dengan mengeklik *check button* pada opsi “I have read and accept the terms of the license agreement” agar dapat mengeklik tombol “See The List Below” untuk mendapatkan daftar produk Pentaho edisi pengembang (*developer*) atau komunitas (*community*). Selain itu, terdapat menu kustomisasi instalasi yang dapat diakses dengan mengeklik tombol “Customize”.



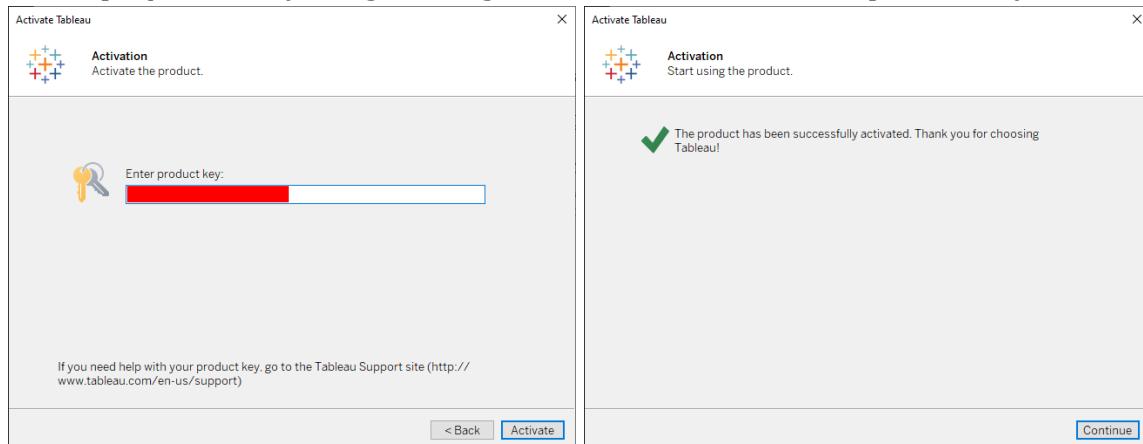
Gambar 3.48. Tahapan Kedua, Ketiga, dan Terakhir Instalasi Tableau Desktop 2024.3

Pada menu kustomisasi, dapat diatur lokasi instalasi dari Tableau Desktop yang secara bawaan (*default*) akan diinstal pada direktori “C:\Program Files\Tableau\Tableau 2024.3”. Dalam menu ini juga terdapat opsi untuk membuat *shortcut* di *desktop* atau di menu Start, serta pemeriksaan pembaruan dari produk Tableau (Tableau Desktop). Setelah itu, apabila konfigurasi instalasi telah sesuai, proses instalasi dapat dimulai dengan mengeklik tombol “Install”.



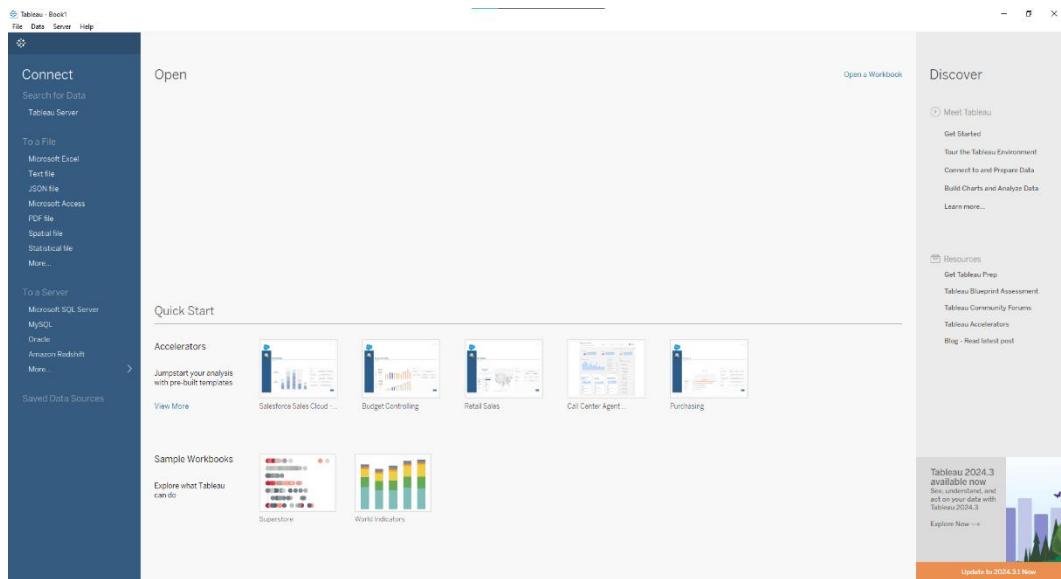
Gambar 3.49. Opsi Aktivasi Licensi Tableau Desktop 2024.3

Setelah instalasi selesai, Tableau Desktop akan terbuka secara otomatis dan diarahkan ke halaman “Activate Tableau” untuk mengaktifkan lisensi dengan tiga opsi, yakni *trial* gratis selama 14 hari, *product key*, dan server. Pada halaman ini, dipilih proses aktivasi lisensi melalui opsi *product key* dengan mengeklik tombol “Activate with product key”.



Gambar 3.50. Tahapan Aktivasi Licensi Tableau Desktop 2024.3 Menggunakan Kode Aktivasi

Setelah itu, mengisikan kolom yang tersedia dengan *product key* Tableau Desktop yang dimiliki dan mengeklik tombol “Activate” untuk mengaktivasi produk. Apabila *product key* yang digunakan valid, maka akan ditampilkan informasi yang menyatakan bahwa produk telah berhasil diaktivasi dan Tableau Desktop dapat dioperasikan.



Gambar 3.51. Tampilan Awal Tableau Desktop 2024.3

3.3. Proses *Extract, Transform, and Load (ETL)* pada Pentaho Data Integration (PDI)

Proses ETL yang dilakukan dalam proyek ini terdiri dari beberapa tahapan yang terstruktur, mulai dari pengumpulan data dengan memanfaatkan *Application Programming Interface* (API), prapemrosesan data, pembuatan basis data *Online Transaction Processing* (OLTP), hingga transformasinya ke dalam basis data *Online Analytical Processing* (OLAP) dengan memanfaatkan basis data MySQL dan menggunakan PDI. Penjelasan dari setiap tahapan ini diterangkan lebih lanjut dalam uraian di bawah ini.

a. Pengumpulan Data

Langkah pertama dalam ETL adalah pengumpulan data yang relevan untuk proyek ini. Berikut adalah penjelasan mengenai data yang dikumpulkan:

- Informasi Geografis Kecamatan di Kota Surabaya

Informasi ini diperoleh melalui proses *web scraping* dari halaman web Wikidata Kota Surabaya (<https://www.wikidata.org/wiki/Q11462>). Informasi yang dikumpulkan mencakup nama kecamatan beserta dengan informasi geografis yang berupa nilai koordinat bujur (*longitude*) dan lintang (*latitude*) dalam derajat desimal atau *decimal degree* (dd). Informasi geografis tersebut diperoleh dari situs web Geographic Names Server (<https://geonames.nga.mil/>) yang direferensikan dalam setiap halaman web Wikidata setiap kecamatan di Kota Surabaya. Nilai koordinat dari setiap kecamatan yang diperoleh nantinya akan digunakan sebagai referensi lokasi ketika akan mendapatkan data cuaca.

Selain itu, dikumpulkan juga data elevasi setiap kecamatan melalui API *Elevation Open-Meteo* (<https://open-meteo.com/en/docs/elevation-api>) dengan menggunakan setiap nilai koordinat bujur (*longitude*) dan lintang (*latitude*) dalam derajat desimal atau *decimal degree* (dd) yang telah diperoleh dari situs web Wikidata dan Geographic Names Server.

- Data Historis Cuaca Harian

Data cuaca harian dikumpulkan melalui API *Historical Weather* Open-Meteo (<https://open-meteo.com/en/docs/historical-weather-api>). Parameter cuaca yang dikumpulkan meliputi suhu rata-rata, kelembapan rata-rata, kecepatan angin, arah angin dominan, jumlah curah hujan, dan durasi siang hari. Data ini dikumpulkan dalam periode waktu yang sesuai dengan rentang tanggal penelitian, yakni mulai dari 1 Januari 2000 hingga 30 November 2024.

- Kode WMO 4677

Setiap kondisi data cuaca harian yang dikumpulkan dari Open-Meteo diinterpretasikan berdasarkan kode meteorologi yang dikeluarkan oleh World Meteorological Organization (WMO). Deskripsi lengkap dari setiap kode cuaca ini diperoleh melalui proses *web scraping* dari halaman web “WMO Meteorological codes” yang terdapat dalam situs web The CEDA Artefacts Service.

b. Transformasi Data

Transformasi data merupakan bagian penting dalam ETL yang melibatkan beberapa tahapan untuk memastikan data siap digunakan dalam basis data OLTP. Tahapan ini meliputi:

- Data historis cuaca harian yang dikumpulkan melalui API *Historical Weather* Open-Meteo berbentuk JavaScript Object Notation (JSON) sehingga diperlukan pengubahan format data menjadi bentuk *tabular* menggunakan *step* seperti “table output” agar dapat disesuaikan dengan tabel yang terdapat dalam basis data MySQL.
- Data elevasi setiap kecamatan yang diperoleh melalui API *Elevation* Open-Meteo digabungkan dengan data nama dan informasi geografis yang berupa nilai koordinat bujur (*longitude*) dan lintang (*latitude*) dalam derajat desimal. Proses penggabungan ini melibatkan penggunaan *step* “JSON input” dan “Select values”. Selain itu, dilakukan juga penyeragaman nama kecamatan di Kota Surabaya yang diperoleh dari situs web Wikidata dengan nama resmi yang tercantum dalam Peraturan Menteri Dalam Negeri (Permendagri) Nomor 72 Tahun 2019 melalui berkas CSV yang diperoleh melalui situs web Kode Wilayah Indonesia (<https://kodewilayah.id/>). Proses penyeragaman ini dilakukan dengan menggunakan *step* “Fuzzy match”. Setelah itu, dilakukan penggabungan data
- Data kode WMO 4677 yang diperoleh ditransformasikan ke dalam bentuk *tabular* agar dapat disesuaikan dengan tabel yang terdapat dalam basis data MySQL.

Selanjutnya, proses transformasi data dari basis data OLTP ke basis data OLAP memerlukan langkah-langkah lebih lanjut yang menyesuaikan data transaksional yang tersimpan dalam basis data OLTP ke dalam format yang lebih cocok untuk analisis dan agregasi di basis data OLAP. Berikut ini adalah tahapan dan penjelasan bagaimana proses transformasi ini dilakukan:

- Ekstraksi data dari basis data OLTP ke dalam basis data OLAP yang melibatkan pembacaan data dari tabel-tabel basis data OLTP yang bersifat transaksional seperti data cuaca harian, tanggal, informasi geografis kecamatan (*latitude*, *longitude*, elevasi), dan deskripsi kode WMO 4677.

- Penyesuaian data yang telah diekstrak dari basis data OLTP agar sesuai dengan skema yang akan digunakan dalam basis data OLAP. Penyesuaian ini bertujuan untuk mengagregasi data, memperbaiki format, dan melakukan perhitungan yang diperlukan untuk analisis lebih lanjut. Berikut adalah beberapa tahapan transformasi:

- Pembuatan Tabel Dimensi Waktu

Pada basis data OLAP, sering kali diperlukan dimensi waktu yang lebih terstruktur seperti tahun, bulan, dan hari. Dalam transformasi ini, data tanggal di OLTP (yang berupa *timestamp*) perlu diubah menjadi format yang lebih mudah untuk analisis multidimensi. Proses ini dapat dilakukan dengan menambah kolom-kolom baru untuk tahun, bulan, dan hari yang merupakan pemecahan dari tanggal yang berformat “yyyy-mm-dd”.

- Pembuatan Tabel Dimensi Lokasi

Pada basis data OLTP, data lokasi seperti provinsi, kota, dan kecamatan disimpan dalam tabel hierarkis terpisah. Dalam basis data OLAP, data lokasi disusun menjadi satu tabel dimensi lokasi yang mencakup semua atribut lokasi (seperti nama provinsi, kota, kecamatan, dan koordinat). Hal ini dilakukan untuk mendukung analisis geografis secara efisien.

- Pembuatan Tabel Dimensi Kondisi Cuaca

Dalam basis data OLTP, kondisi cuaca disimpan menggunakan kode numerik (*weather_code*) yang kurang informatif untuk analisis langsung. Pada basis data OLAP, kode ini diubah menjadi tabel dimensi yang berisi pasangan kode dan deskripsi (*weather_code_desc*).

- Pembuatan Tabel Dimensi Curah Hujan

Pada OLTP, data curah hujan dicatat sebagai nilai numerik dalam tabel *weather_daily* bersamaan dengan parameter cuaca lainnya. Dalam basis data OLAP, dibentuk tabel dimensi baru yang dapat merepresentasikan intensitas hujan berdasarkan jumlah curah hujan yang tercatat sesuai dengan kriteria yang telah ditetapkan Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) yang tercantum dalam halaman web “Probabilistik Curah Hujan 20 mm (tiap 24 jam)” pada situs web resmi BMKG (<https://www.bmkg.go.id/cuaca/probabilistik-curah-hujan.bmkg>). Tabel dimensi yang dibentuk akan terdiri dari atribut intensitas hujan (*rainfall_intensity*), curah hujan minimum (*pcp_min*), dan maksimum (*pcp_max*). Pembentukan tabel dimensi ini bertujuan untuk membantu dalam analisis pola curah hujan berdasarkan kategori tertentu.

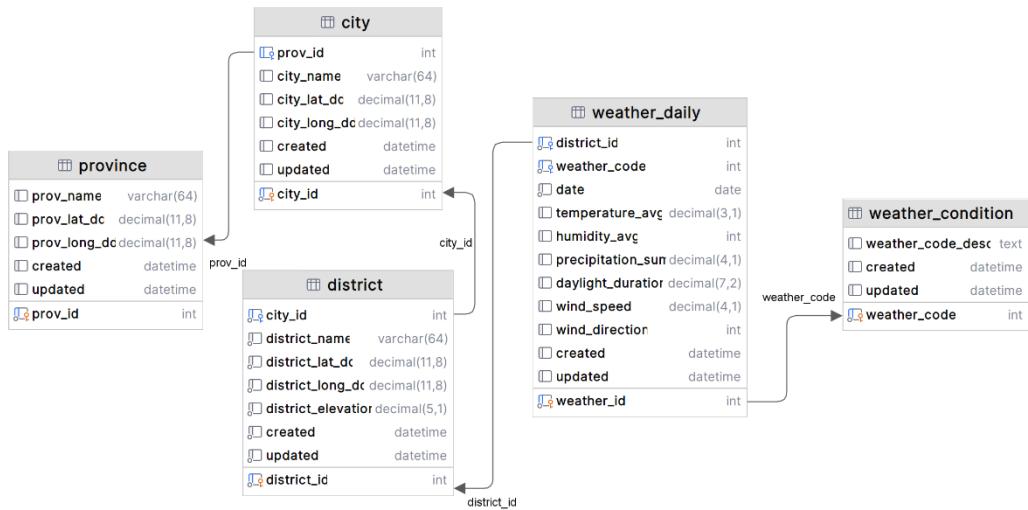
- Pembuatan Tabel Fakta Cuaca Harian

Dalam basis data OLAP, atribut suhu rata-rata, kelembapan rata-rata, kecepatan angin, arah angin dominan, jumlah curah hujan, dan durasi siang hari digabungkan menjadi satu ke dalam tabel fakta (*fact_weather*) yang terhubung dengan tabel dimensi waktu, lokasi, curah hujan, dan kondisi cuaca. Selain itu, ditambahkan atribut “*daylight_percentage*” yang merepresentasikan persentase durasi siang hari dalam satu hari. Nilai dari atribut tersebut didapatkan melalui

perhitungan nilai atribut “daylight_duration” yang dibagi dengan durasi dalam sehari dan dikalikan dengan seratus.

Setelah transformasi selesai, data kemudian dimuat ke dalam basis data OLAP yang telah dibentuk dengan konsep *star schema*. Pada skema ini, tabel fakta akan dikelilingi oleh beberapa tabel dimensi seperti tabel waktu, tabel lokasi, tabel kondisi cuaca (yang mencakup deskripsi kode WMO), dan tabel rainfall (yang mencakup tingkat curah hujan).

c. Pembuatan Basis Data *Online Transaction Processing* (OLTP)

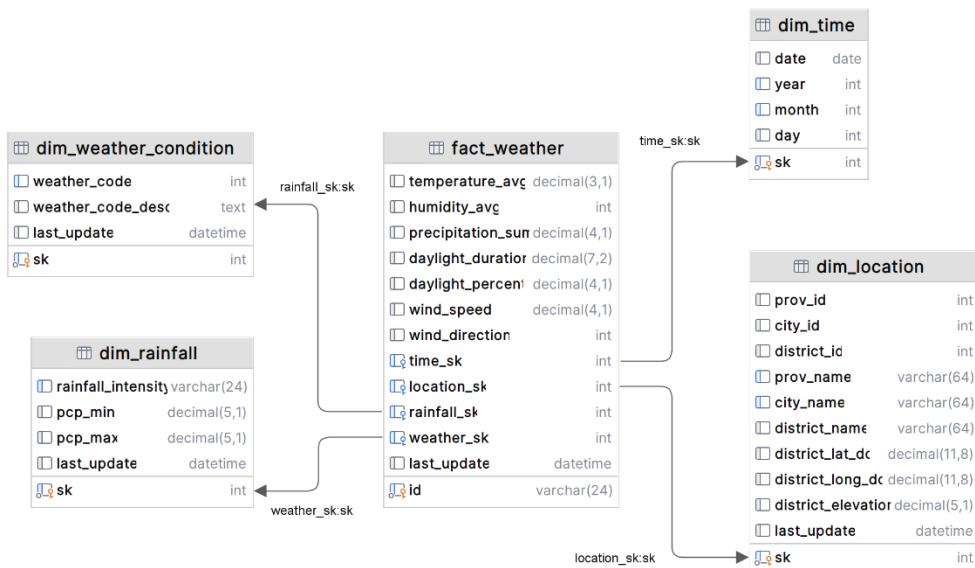


Gambar 3.52. Struktur Basis Data OLTP dalam Proyek

Gambar 3.52 menampilkan struktur basis data *Online Transaction Processing* (OLTP) dengan menerapkan *snowflake schema* yang terdiri dari lima tabel utama, yaitu “weather_daily”, “weather_condition”, “district”, “city”, dan “province”. Inisialisasi basis data OLTP dilakukan menggunakan perintah SQL pada MySQL. Setiap tabel telah didefinisikan secara rinci dengan penetapan primary key, foreign key, serta tipe data yang sesuai untuk setiap atribut. Selain itu, relasi antar tabel juga telah diatur dengan jelas untuk menjaga integritas data. Data hasil transformasi dari pengumpulan informasi cuaca dan lokasi akan disimpan di basis data OLTP.

d. Pembuatan Basis Data *Online Analytical Processing* (OLAP)

Gambar 3.53 menunjukkan struktur basis data OLAP yang menerapkan konsep *star schema* yang terdiri dari lima tabel, yaitu “dim_time”, “dim_location”, “dim_weather_condition”, “dim_rainfall”, dan “fact_weather”. Setiap tabel dimensi menghubungkan data hasil transformasi dari basis data OLTP ke tabel fakta yang berisi informasi utama cuaca yang telah diolah. Untuk inisialisasi basis data OLAP ini, akan digunakan perintah SQL dalam MySQL. Setiap data yang ditransformasikan dari OLTP ke masing-masing tabel dimensi disimpan ke dalam basis data OLAP.



Gambar 3.53. Struktur Basis Data OLTP dalam Proyek

3.4. Visualisasi Dasbor pada Tableau Desktop

Proses visualisasi dasbor yang dilakukan dalam proyek ini menggunakan Tableau sebagai alat utama untuk menyajikan data secara interaktif dan mudah dipahami. Tahapan dalam pembuatan visualisasi ini terdiri dari beberapa langkah yang terstruktur, dimulai dari menyambungkan koneksi dengan MySQL sebagai basis data serta, memanfaatkan fitur visualisasi data dalam Tableau untuk menghasilkan dasbor yang interaktif. Penjelasan lebih lanjut dari setiap tahapan ini akan diuraikan dalam bagian berikut:

a. Membuat Koneksi dengan MySQL

Tahap pertama dalam pembuatan visualisasi adalah menyambungkan Tableau dengan MySQL sebagai basis data. Proses ini melibatkan pengaturan koneksi antara Tableau dan server MySQL menggunakan kredensial yang sesuai, seperti alamat server, nama basis data, serta username dan password. Setelah koneksi berhasil dibuat, Tableau dapat mengakses data yang tersimpan di MySQL untuk digunakan dalam visualisasi.

b. Pembuatan Dasbor

Setelah koneksi dengan MySQL terhubung, langkah berikutnya adalah membuat visualisasi dasbor di Tableau. Pada tahap ini, data yang diambil dari MySQL akan diproses dan dianalisis untuk menghasilkan berbagai jenis grafik dan tabel. Tableau menyediakan berbagai jenis visualisasi, seperti grafik batang, garis, peta, dan diagram lainnya yang dapat dipilih berdasarkan tujuan analisis. Dasbor interaktif dapat dibuat dengan menambahkan filter, parameter, dan alat navigasi untuk memudahkan pengguna dalam menggali informasi lebih lanjut. Setiap elemen visualisasi dapat disesuaikan dengan kebutuhan analisis untuk memberikan wawasan yang jelas dan mudah dipahami.

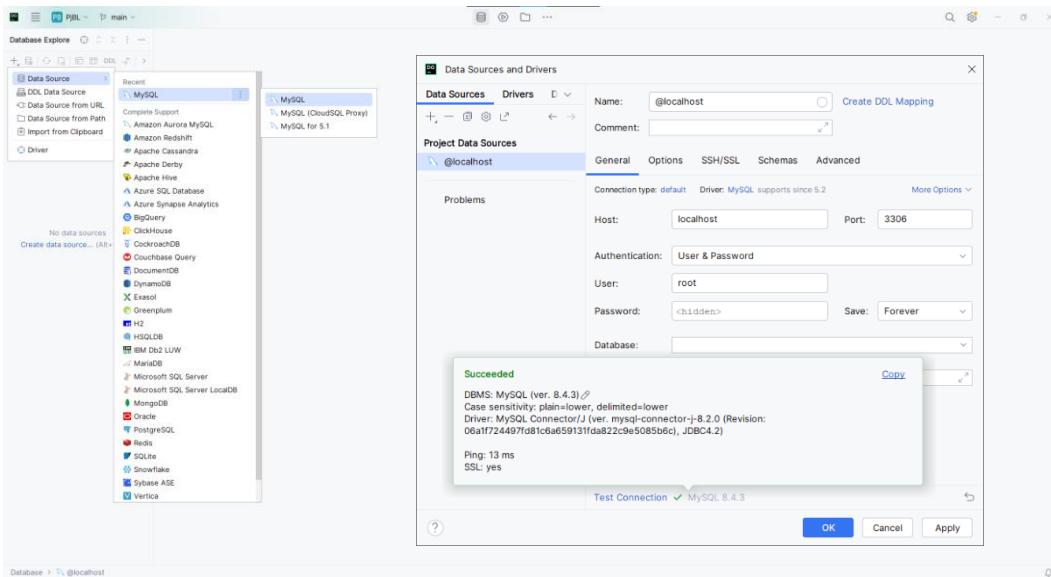
BAB IV

IMPLEMENTASI STUDI KASUS MENGGUNAKAN MYSQL, PENTAHO DATA INTEGRATION, DAN TABLEAU

4.1. Pengonfigurasian Koneksi Basis Data MySQL

4.1.1. DataGrip

Pengonfigurasian koneksi basis data MySQL dalam DataGrip dilakukan dengan menambahkan sumber data (*data source*) yang berupa MySQL pada menu “Data Sources and Drivers” yang dapat diakses dengan menekan tombol dengan ikon ataupun menekan Ctrl+Alt+Shift+S pada papan tombol (*keyboard*).



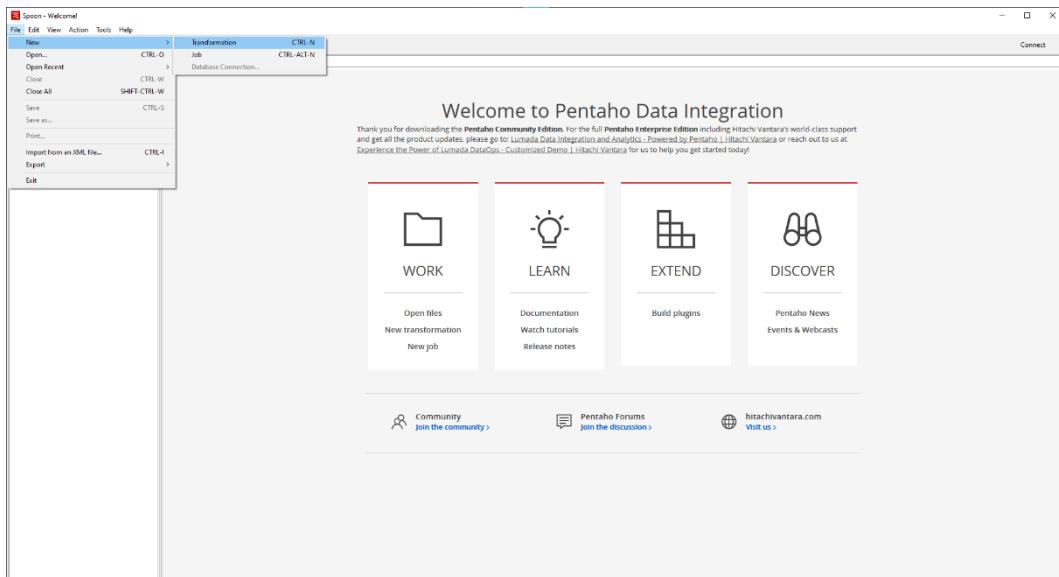
Gambar 4.1. Pengonfigurasian Koneksi Basis Data MySQL dalam DataGrip

Pada tab “General” dalam jendela menu “Data Sources and Drivers”, diisikan nilai-nilai parameter konfigurasi utama yang terdiri dari *host*, *port*, jenis otentikasi, *user*, dan *password* dari server MySQL yang ingin dihubungkan. Kemudian, diklik hipertaut “Test Connection” untuk menguji koneksi berdasarkan parameter-parameter yang telah diisikan. Apabila koneksi yang dikonfigurasi telah berhasil terhubung ke basis data, maka diklik tombol “Apply” untuk menyimpan koneksi basis data. Setelah itu, mengeklik tombol “OK” untuk keluar dari jendela menu “Data Sources and Drivers”.

4.1.2. Pentaho Data Integration

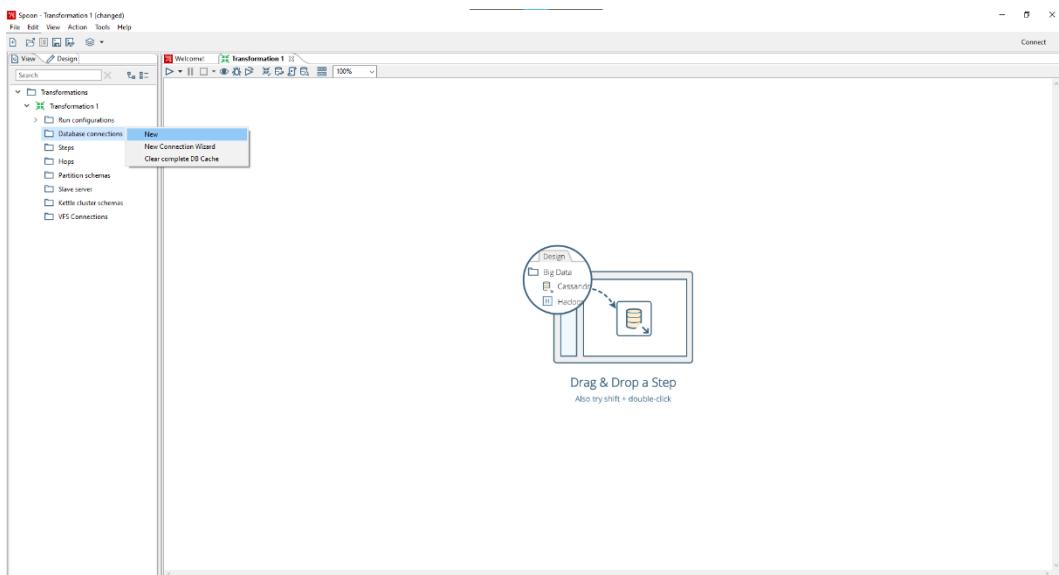
Pengonfigurasian koneksi basis data MySQL dalam Pentaho Data Integration (PDI) dapat dilakukan dengan dua cara berbeda, yakni melalui berkas *transformation* dan repositori basis data (*database repository*) dalam menu “Repository Manager”. Dalam proyek ini, diimplementasikan pengonfigurasian koneksi melalui berkas *transformation* yang cocok untuk seluruh versi PDI. Proses ini pengonfigurasian ini, diawali dengan membuat sebuah *transformation* baru dengan menekan Ctrl+N pada papan tombol

(keyboard) atau mengeklik opsi “Transformation” yang terdapat dalam sub-menu “New” dalam menu “File”.



Gambar 4.2. Tahapan Pertama Pengonfigurasian Koneksi Basis Data MySQL dalam PDI

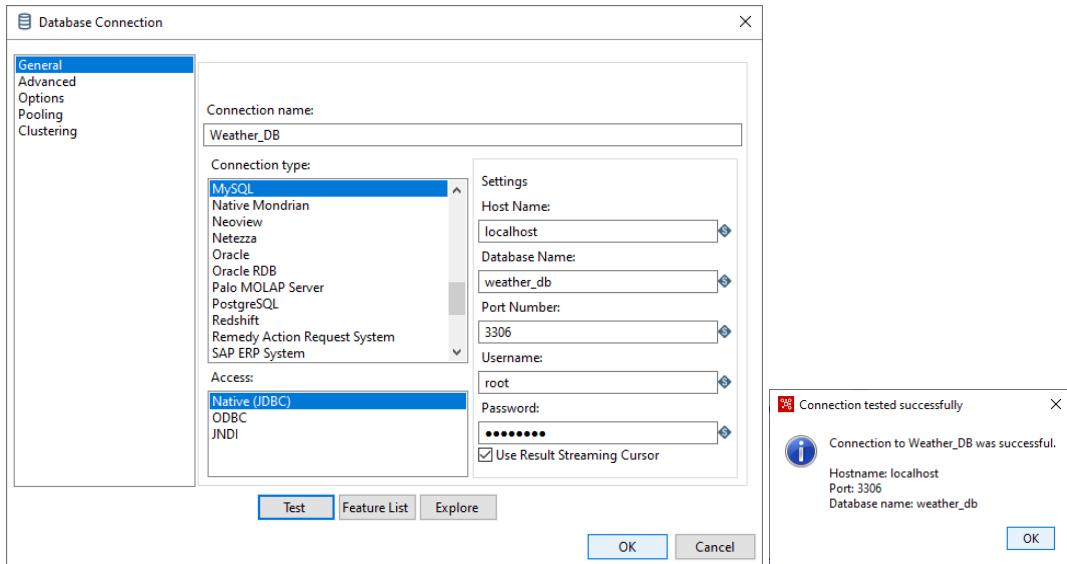
Selanjutnya, mengakses direktori “Database connections” yang terdapat dalam tab “View” dan mengeklik kanan pada direktori tersebut, serta memilih opsi “New” untuk menambahkan koneksi baru.



Gambar 4.3. Tahapan Kedua Pengonfigurasian Koneksi Basis Data MySQL dalam PDI

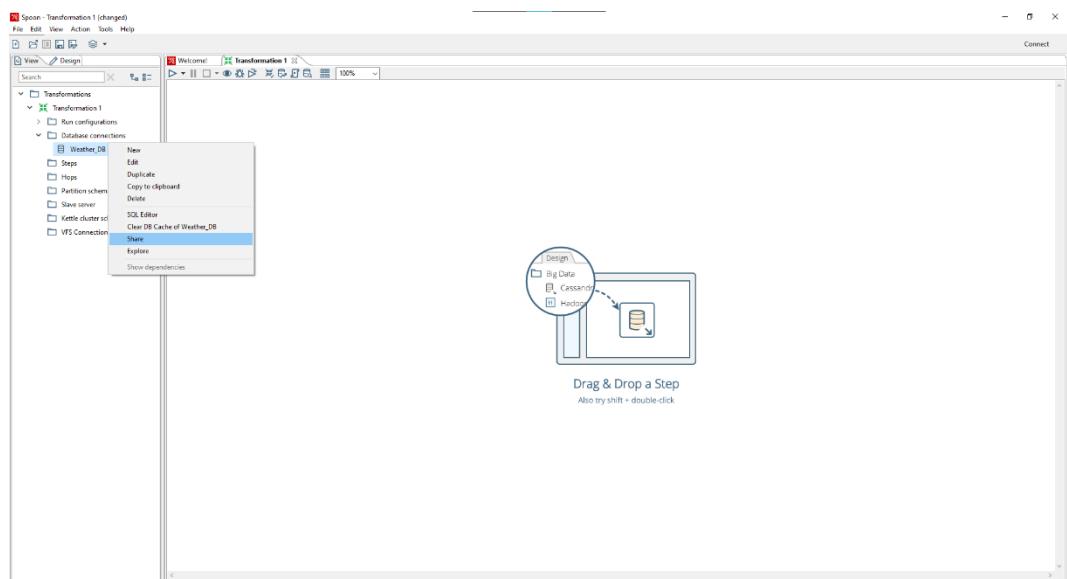
Pada jendela menu “Database Connection”, ditambahkan diisikan nilai-nilai parameter konfigurasi pada kolom (*field*) “Connection name:”, “Connection type:”, “Access:”, dan “Settings” sesuai dengan nilai parameter yang tertaut dengan server MySQL yang ingin dihubungkan. Kemudian, diklik tombol “Test” untuk menguji koneksi. Apabila koneksi

yang dibuat telah berhasil terhubung ke basis data, diklik tombol “OK” untuk menyimpan konfigurasi koneksi basis data.



Gambar 4.4. Tahapan Ketiga Pengonfigurasian Koneksi Basis Data MySQL dalam PDI

Setelah itu, diaktifkan opsi *sharing* koneksi basis data dengan mengeklik kanan nama koneksi basis data yang terdapat pada direktori “Database connections”. Kemudian, dipilih dan diklik opsi “Share”. Opsi ini diaktifkan agar koneksi basis data yang telah dibuat dapat digunakan dalam *transformation* lainnya.

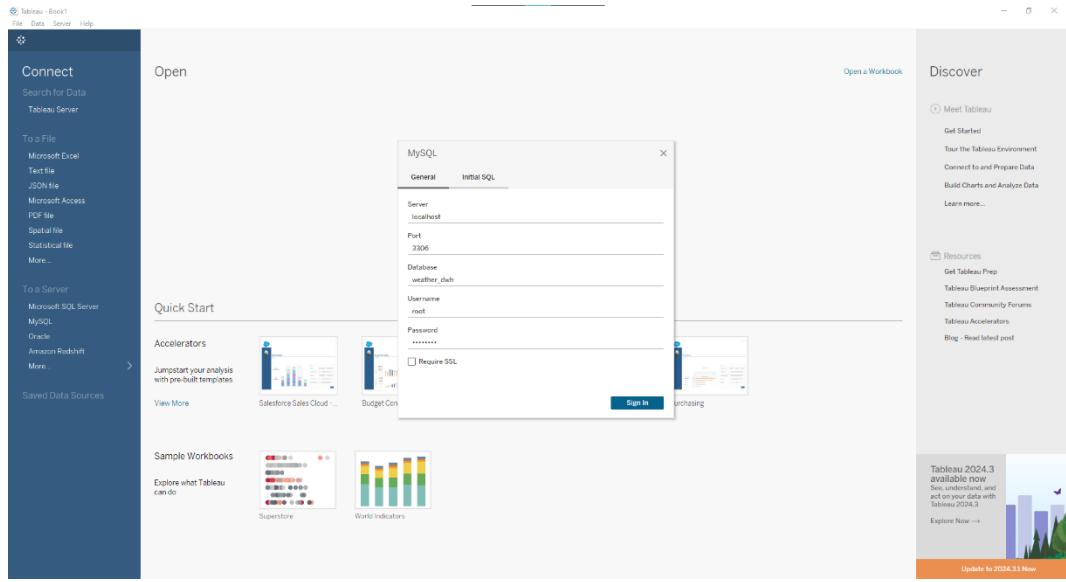


Gambar 4.5. Tahapan Terakhir Pengonfigurasian Koneksi Basis Data MySQL dalam PDI

4.1.3. Tableau Desktop

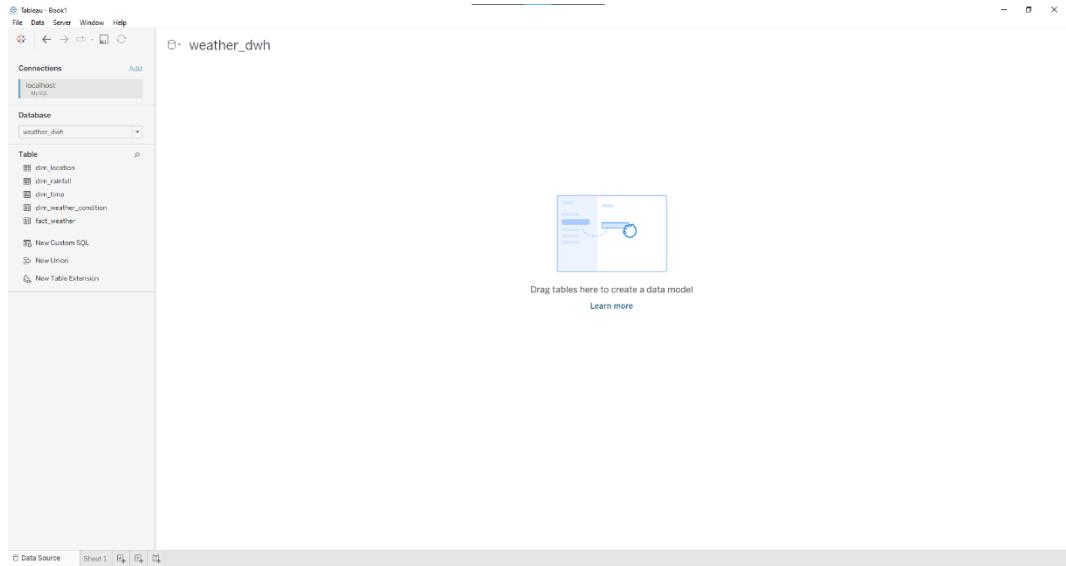
Pengonfigurasian koneksi basis data MySQL dalam Tableau Desktop dilakukan dengan memilih opsi “MySQL” pada sub-daftar “To a Server” dalam daftar “Connect” yang terdapat dalam panel di sebelah kiri dari halaman awal Tableau Desktop. Setelah itu,

diinisialisasikan server, *port*, skema atau basis data (*database*), *username*, dan *password* dari server MySQL yang ingin dihubungkan.



Gambar 4.6. Pengonfigurasian Koneksi Basis Data MySQL dalam Tableau Desktop

Setelah itu, diklik tombol “Sign In” untuk mengakses *workbook* dengan sumber data yang berasal dari tabel-tabel dalam basis data yang telah dipilih.



Gambar 4.7. Tampilan Awal Tableau Desktop Setelah Terkoneksi dengan Basis Data MySQL

4.2. Penyiapan Basis Data

4.2.1. Pembuatan Basis Data *Online Transaction Processing* (OLTP)

Skema atau basis data *Online Transaction Processing* (OLTP) yang dibuat dalam proyek ini terdiri dari 5 tabel, yaitu “province”, “city”, “district”, “weather_condition”, dan “weather_daily”. Kelima tabel ini saling terhubung dengan *foreign key* dan memanfaatkan *trigger* untuk mengelola pencatatan waktu secara otomatis pada setiap operasi *insert* dan

update. Penjelasan lebih lanjut mengenai tabel-tabel dalam basis data OLTP dijelaskan dalam uraian di bawah ini.

a. Tabel “province”

Tabel ini memiliki enam kolom, antara lain:

- prov_id (int): kode unik untuk setiap provinsi dan sebuah *primary key*,
- prov_name (varchar(64)): nama provinsi,
- prov_lat_dd (decimal(11, 8)): koordinat lintang (*latitude*) provinsi dalam derajat desimal atau *decimal degree* (dd),
- prov_long_dd (decimal(11, 8)): koordinat bujur (*longitude*) provinsi dalam derajat desimal atau *decimal degree* (dd),
- created (datetime): waktu saat data provinsi pertama kali ditambahkan,
- updated (datetime): waktu saat data provinsi terakhir diperbarui.

Trigger pada tabel ini berfungsi untuk mengotomatiskan pengisian dan pembaruan kolom yang menyimpan informasi waktu ketika terdapat perubahan data. *Trigger* “trigger_province_insert” dijalankan setiap kali terdapat data baru yang dimasukkan ke dalam tabel “province” melalui operasi *insert*. *Trigger* ini secara otomatis mengisi kolom “created” dan “updated” dengan nilai tanggal dan waktu melalui fungsi *sysdate()* saat data tersebut dimasukkan sehingga tidak perlu dilakukan secara manual. Sementara itu, *trigger* “trigger_province_update” dijalankan setiap kali data dalam tabel “province” diperbarui melalui operasi *update*. Kegunaan dari *trigger* ini adalah untuk memperbarui kolom “updated” dengan tanggal dan waktu terbaru yang memastikan bahwa setiap perubahan pada data selalu tercatat kapan terakhir kali dilakukan.

b. Tabel “city”

Tabel ini memiliki tujuh kolom, antara lain:

- city_id (int): kode unik untuk setiap kota dan sebuah *primary key*,
- prov_id (int): kode unik untuk setiap provinsi dan sebuah *foreign key* yang terhubung dengan kolom “prov_id” di tabel “province”,
- city_name (varchar(64)): nama kota,
- city_lat_dd (decimal(11, 8)): koordinat lintang (*latitude*) kota dalam derajat desimal atau *decimal degree* (dd),
- city_long_dd (decimal(11, 8)): koordinat bujur (*longitude*) kota dalam derajat desimal atau *decimal degree* (dd),
- created (datetime): waktu saat data kota pertama kali ditambahkan,
- updated (datetime): waktu saat data kota terakhir diperbarui.

Trigger pada tabel ini berfungsi untuk mengotomatiskan pengisian dan pembaruan kolom yang menyimpan informasi waktu ketika terdapat perubahan data. *Trigger* “trigger_city_insert” dijalankan setiap kali terdapat data baru yang dimasukkan ke dalam tabel “city” melalui operasi *insert*. *Trigger* ini secara otomatis mengisi kolom “created” dan “updated” dengan nilai tanggal dan waktu melalui fungsi *sysdate()* saat data tersebut dimasukkan sehingga tidak perlu dilakukan secara manual. Sementara

itu, *trigger* “trigger_city_update” dijalankan setiap kali data dalam tabel “city” diperbarui melalui operasi *update*. Kegunaan dari *trigger* ini adalah untuk memperbarui kolom “updated” dengan tanggal dan waktu terbaru yang memastikan bahwa setiap perubahan pada data selalu tercatat kapan terakhir kali dilakukan.

c. Tabel “district”

Tabel ini memiliki delapan kolom, antara lain:

- district_id (int): kode unik untuk setiap kecamatan dan sebuah *primary key*,
- city_id (int): kode unik untuk setiap kota dan sebuah *foreign key* yang terhubung dengan kolom “city_id” di tabel “city”,
- district_name (varchar(64)): nama kecamatan,
- district_lat_dd (decimal(11, 8)): koordinat lintang (*latitude*) kecamatan dalam derajat desimal atau *decimal degree* (dd),
- district_long_dd (decimal(11, 8)): koordinat bujur (*longitude*) kecamatan dalam derajat desimal atau *decimal degree* (dd),
- district_elevation (decimal(5, 1)): ketinggian kecamatan dari permukaan laut (dalam meter),
- created (datetime): waktu saat data kecamatan pertama kali ditambahkan,
- updated (datetime): waktu saat data kecamatan terakhir diperbarui.

Trigger pada tabel ini berfungsi untuk mengotomatiskan pengisian dan pembaruan kolom yang menyimpan informasi waktu ketika terdapat perubahan data. *Trigger* “trigger_district_insert” dijalankan setiap kali terdapat data baru yang dimasukkan ke dalam tabel “district” melalui operasi *insert*. *Trigger* ini secara otomatis mengisi kolom “created” dan “updated” dengan nilai tanggal dan waktu melalui fungsi *sysdate()* saat data tersebut dimasukkan sehingga tidak perlu dilakukan secara manual. Sementara itu, *trigger* “trigger_district_update” dijalankan setiap kali data dalam tabel “district” diperbarui melalui operasi *update*. Kegunaan dari *trigger* ini adalah untuk memperbarui kolom “updated” dengan tanggal dan waktu terbaru yang memastikan bahwa setiap perubahan pada data selalu tercatat kapan terakhir kali dilakukan.

d. Tabel “weather_condition”

Tabel ini memiliki empat kolom, yaitu:

- weather_code (int): kode unik untuk setiap kondisi cuaca sesuai dengan World Meteorological Organization (WMO) Code 4677 dan sebuah *primary key*,
- weather_code_desc (text): deskripsi kondisi cuaca (misalnya, cerah, berawan, hujan) untuk setiap WMO Code 4677,
- created (datetime): waktu saat data kondisi cuaca pertama kali ditambahkan,
- updated (datetime): waktu saat data kondisi cuaca terakhir diperbarui.

Trigger pada tabel ini berfungsi untuk mengotomatiskan pengisian dan pembaruan kolom yang menyimpan informasi waktu ketika terdapat perubahan data. *Trigger* “trigger_weather_code_insert” dijalankan setiap kali terdapat data baru yang dimasukkan ke dalam tabel “weather_code” melalui operasi *insert*. *Trigger* ini secara

otomatis mengisi kolom “created” dan “updated” dengan nilai tanggal dan waktu melalui fungsi `sysdate()` saat data tersebut dimasukkan sehingga tidak perlu dilakukan secara manual. Sementara itu, *trigger* “trigger_weather_code_update” dijalankan setiap kali data dalam tabel “weather_code” diperbarui melalui operasi *update*. Kegunaan dari *trigger* ini adalah untuk memperbarui kolom “updated” dengan tanggal dan waktu terbaru yang memastikan bahwa setiap perubahan pada data selalu tercatat kapan terakhir kali dilakukan.

e. Tabel “weather_daily”

Tabel ini memiliki 12 kolom, yaitu:

- `weather_id` (int): kode unik untuk setiap catatan cuaca harian dan sebuah *primary key*,
- `district_id` (int): kode unik untuk setiap kecamatan dan sebuah *foreign key* yang terhubung dengan kolom “`district_id`” di tabel “`district`”,
- `weather_code` (int): kode unik untuk setiap kondisi cuaca dan sebuah *foreign key* yang terhubung dengan kolom “`weather_code`” di tabel “`weather_condition`”,
- `date` (date): tanggal pencatatan cuaca,
- `temperature_avg` (decimal(3, 1)): suhu rata-rata harian (dalam derajat celsius),
- `humidity_avg` (int): kelembaban rata-rata harian (dalam persen),
- `precipitation_sum` (decimal(4, 1)): jumlah curah hujan harian (dalam milimeter),
- `daylight_duration` (decimal(7, 2)): durasi siang hari (dalam detik).
- `wind_speed` (decimal(4, 1)): kecepatan angin rata-rata harian (dalam meter per detik),
- `wind_direction` (int): arah angin dominan dalam satu hari (dalam derajat),
- `created` (datetime): waktu saat data cuaca harian pertama kali ditambahkan,
- `updated` (datetime): waktu saat data cuaca harian terakhir diperbarui.

Trigger pada tabel ini berfungsi untuk mengotomatiskan pengisian dan pembaruan kolom yang menyimpan informasi waktu ketika terdapat perubahan data. *Trigger* “trigger_weather_daily_insert” dijalankan setiap kali terdapat data baru yang dimasukkan ke dalam tabel “weather_daily” melalui operasi *insert*. *Trigger* ini secara otomatis mengisi kolom “created” dan “updated” dengan nilai tanggal dan waktu melalui fungsi `sysdate()` saat data tersebut dimasukkan sehingga tidak perlu dilakukan secara manual. Sementara itu, *trigger* “trigger_weather_daily_update” dijalankan setiap kali data dalam tabel “weather_daily” diperbarui melalui operasi *update*. Kegunaan dari *trigger* ini adalah untuk memperbarui kolom “updated” dengan tanggal dan waktu terbaru yang memastikan bahwa setiap perubahan pada data selalu tercatat kapan terakhir kali dilakukan.

Untuk menginisialisasi basis data OLTP dengan tabel dan kolom seperti yang telah dijelaskan di atas, dapat digunakan kode SQL yang ditunjukkan pada Gambar 4.8 pada MySQL. Kode ini mencakup pembuatan tabel-tabel yang diperlukan beserta dengan setiap kolomnya dan *primary key*, *foreign key*, serta sejumlah *trigger* yang akan mengotomatisasi pencatatan waktu pada setiap perubahan data di setiap tabel.

```

create table if not exists province
(
    prov_id      int          not null primary key,
    prov_name    varchar(64)   null,
    prov_lat_dd  decimal(11, 8) null,
    prov_long_dd decimal(11, 8) null,
    created      datetime     null,
    updated      datetime     null
);

create table if not exists city
(
    city_id      int          not null primary key,
    prov_id      int          null,
    city_name    varchar(64)   null,
    city_lat_dd  decimal(11, 8) null,
    city_long_dd decimal(11, 8) null,
    created      datetime     null,
    updated      datetime     null,
    constraint city_province_fk
        foreign key (prov_id) references province (prov_id)
            on update cascade
);
create definer = root@localhost trigger trigger_city_insert
before insert
on city
for each row
begin
    set new.created = sysdate();
    set new.updated = sysdate();
end;

create definer = root@localhost trigger trigger_city_updated
before update
on city
for each row
begin
    set new.updated = sysdate();
end;

create table if not exists district
(
    district_id  int          not null primary key,
    city_id      int          not null,
    district_name varchar(64)  not null,
    district_lat_dd decimal(11, 8) not null,
    district_long_dd decimal(11, 8) not null,
    district_elevation decimal(5, 1) not null,
    created      datetime     not null,
    updated      datetime     not null,
    constraint district_city_fk
        foreign key (city_id) references city (city_id)
);

```

```

        on update cascade
);

create definer = root@localhost trigger trigger_district_insert
    before insert
    on district
    for each row
begin
    set new.created = sysdate();
    set new.updated = sysdate();
end;

create definer = root@localhost trigger trigger_district_update
    before update
    on district
    for each row
begin
    set new.updated = sysdate();
end;

create definer = root@localhost trigger trigger_province_insert
    before insert
    on province
    for each row
begin
    set new.created = sysdate();
    set new.updated = sysdate();
end;

create definer = root@localhost trigger trigger_province_update
    before update
    on province
    for each row
begin
    set new.updated = sysdate();
end;

create table if not exists weather_condition
(
    weather_code      int      not null primary key,
    weather_code_desc text     null,
    created          datetime null,
    updated          datetime null
);

create definer = root@localhost trigger trigger_weather_code_insert
    before insert
    on weather_condition
    for each row
begin
    set new.created = sysdate();
    set new.updated = sysdate();
end;

```

```

create definer = root@localhost trigger trigger_weather_code_update
  before update
  on weather_condition
  for each row
begin
  set new.updated = sysdate();
end;

create table if not exists weather_daily
(
    weather_id      int auto_increment
      primary key,
    district_id     int          not null,
    weather_code     int          not null,
    date            date         not null,
    temperature_avg decimal(3, 1) null,
    humidity_avg    int          null,
    precipitation_sum decimal(4, 1) null,
    daylight_duration decimal(7, 2) null,
    wind_speed       decimal(4, 1) null,
    wind_direction   int          null,
    created          datetime     null,
    updated          datetime     null,
    constraint weather_daily_code_fk
        foreign key (weather_code) references weather_condition
(weather_code)
        on update cascade,
    constraint weather_daily_location_fk
        foreign key (district_id) references district (district_id)
        on update cascade
);

create definer = root@localhost trigger trigger_weather_daily_insert
  before insert
  on weather_daily
  for each row
begin
  set new.created = sysdate();
  set new.updated = sysdate();
end;

create definer = root@localhost trigger trigger_weather_daily_updated
  before update
  on weather_daily
  for each row
begin
  set new.updated = sysdate();
end;

```

Gambar 4.8. Kode SQL Inisialisasi Basis Data OLTP

4.2.2. Pembuatan Basis Data *Online Analytical Processing* (OLAP)

Skema atau basis data *Online Analytical Processing* (OLAP) yang dibuat dalam proyek ini terdiri dari 5 tabel, antara lain “dim_location”, “dim_rainfall”, “dim_time”, “dim_weather_condition”, dan “fact_weather”. Basis data ini dirancang untuk menyimpan dan menganalisis data cuaca dari berbagai dimensi, seperti lokasi, intensitas hujan, waktu, dan kondisi cuaca. Tabel-tabel tersebut terhubung dengan *foreign key* dan memanfaatkan indeks untuk mempercepat pencarian data. Penjelasan lebih lanjut mengenai tabel-tabel dalam basis data OLAP dijelaskan dalam uraian di bawah ini.

a. Tabel “dim_location”

Indeks “loc_city_key” dan “loc_prov_key” dibuat untuk mempercepat pencarian berdasarkan nama kota dan nama provinsi. Tabel ini memiliki 11 kolom, antara lain:

- sk (int): kode unik yang digunakan untuk mengidentifikasi setiap lokasi dan sebuah *surrogate key* atau *primary key*,
- prov_id (int): kode unik untuk setiap provinsi dan sebuah *business key* yang berasal dari data sumber (OLTP),
- city_id (int): kode unik untuk setiap kota dan sebuah *business key* yang berasal dari data sumber (OLTP),
- district_id (int): kode unik untuk setiap kecamatan dan sebuah *business key* yang berasal dari data sumber (OLTP),
- prov_name (varchar(64)): nama provinsi,
- city_name (varchar(64)): nama kota,
- district_name (varchar(64)): nama kecamatan,
- district_lat_dd (decimal(11, 8)): koordinat lintang (*latitude*) provinsi dalam derajat desimal atau *decimal degree* (dd),
- district_long_dd (decimal(11, 8)): koordinat bujur (*longitude*) provinsi dalam derajat desimal atau *decimal degree* (dd),
- district_elevation (decimal(5, 1)): ketinggian kecamatan dari permukaan laut (dalam meter),
- last_update (datetime): waktu saat data lokasi terakhir diperbarui.

b. Tabel “dim_rainfall”

Indeks “rainfall_key” dibuat untuk mempercepat pencarian berdasarkan intensitas hujan. Tabel ini memiliki lima kolom, yaitu:

- sk (int): kode unik yang digunakan untuk mengidentifikasi setiap intensitas hujan dan sebuah *surrogate key* atau *primary key*,
- rainfall_intensity (varchar(24)): deskripsi intensitas hujan (misalnya, ringan, sedang, lebat, dsb.),
- pcp_min (decimal(5, 1)): curah hujan minimum dalam satu hari,
- pcp_max (decimal(5, 1)): curah hujan maksimum,
- last_update (datetime): waktu saat data intensitas hujan terakhir diperbarui.

c. Tabel dim_time

Indeks “month_key” dan “year_key” dibuat untuk mempercepat pencarian berdasarkan bulan dan tahun. Tabel ini memiliki lima kolom, yaitu:

- sk (int): kode unik yang digunakan untuk mengidentifikasi setiap waktu dan sebuah *surrogate key* atau *primary key*,
- date (date): tanggal dalam format yyyy-MM-dd,
- year (int): tahun (yyyy),
- month (int): bulan (MM),
- day (int): hari (dd).

d. Tabel “dim_weather_condition”

Indeks “weather_code_key” dibuat untuk mempercepat pencarian berdasarkan kode cuaca. Tabel ini memiliki empat kolom, yaitu:

- sk (int): kode unik yang digunakan untuk mengidentifikasi setiap kondisi cuaca dan sebuah sebuah *surrogate key* atau *primary key*,
- weather_code (int): kode unik untuk setiap kondisi cuaca sesuai dengan World Meteorological Organization (WMO) Code 4677 dan sebuah *business key* yang berasal dari data sumber (OLTP),
- weather_code_desc (text): deskripsi kondisi cuaca (misalnya, cerah, berawan, hujan) untuk setiap WMO Code 4677,
- last_update (datetime): waktu saat data kondisi cuaca terakhir diperbarui.

e. Tabel fact_weather

Indeks “location_key”, “rainfall_key”, “time_key”, dan “weather_key” dibuat untuk mempercepat pencarian berdasarkan *foreign key* yang merujuk ke tabel dimensi terkait. Tabel ini memiliki 13 kolom, antara lain:

- id (varchar(24)): kode unik untuk setiap catatan cuaca harian dan sebuah *primary key*,
- temperature_avg (decimal(3, 1)): suhu rata-rata harian (dalam derajat celsius),
- humidity_avg (int): kelembaban rata-rata harian (dalam persen),
- precipitation_sum (decimal(4, 1)): jumlah curah hujan harian (dalam milimeter),
- daylight_duration (decimal(7, 2)): durasi siang hari (dalam detik),
- daylight_percent (decimal(4, 1)): durasi siang hari (dalam persen),
- wind_speed (decimal(4, 1)): kecepatan angin rata-rata harian (dalam meter per detik),
- wind_direction (int): arah angin dominan dalam satu hari (dalam derajat),
- time_sk (int): kode unik yang digunakan untuk mengidentifikasi setiap waktu dan sebuah *foreign key* yang merujuk ke tabel “dim_time”,
- location_sk (int): kode unik yang digunakan untuk mengidentifikasi setiap lokasi dan sebuah *foreign key* yang merujuk ke tabel “dim_location”,
- rainfall_sk (int): kode unik yang digunakan untuk mengidentifikasi setiap intensitas hujan dan sebuah *foreign key* yang merujuk ke tabel “dim_rainfall”,

- `weather_sk` (int): kode unik yang digunakan untuk mengidentifikasi setiap kondisi cuaca dan sebuah *foreign key* merujuk ke tabel “`dim_weather_condition`”,
- `last_update` (datetime): waktu saat data cuaca terakhir diperbarui.

Untuk menginisialisasi basis data OLAP dengan tabel dan kolom seperti yang telah dijelaskan di atas, dapat digunakan kode SQL yang ditunjukkan pada Gambar 4.9 pada MySQL. Kode ini mencakup pembuatan tabel-tabel yang diperlukan untuk analisis data cuaca beserta dengan setiap kolomnya dan *primary key* atau *surrogate key* pada tabel dimensi, *foreign key* atau *business key*, serta indeks untuk mempercepat pencarian atau kueri data.

```

create table if not exists dim_location
(
    sk           int auto_increment
    primary key,
    prov_id      int          null,
    city_id      int          null,
    district_id  int          null,
    prov_name    varchar(64)  null,
    city_name    varchar(64)  null,
    district_name varchar(64)  null,
    district_lat_dd decimal(11, 8) null,
    district_long_dd decimal(11, 8) null,
    district_elevation decimal(5, 1) null,
    last_update   datetime     null
);

create index loc_city_key
    on dim_location (city_name);

create index loc_prov_key
    on dim_location (prov_name);

create table if not exists dim_rainfall
(
    sk           int          not null
    primary key,
    rainfall_intensity varchar(24)  null,
    pcp_min       decimal(5, 1) null,
    pcp_max       decimal(5, 1) null,
    last_update   datetime     null
);

create index rainfall_key
    on dim_rainfall (rainfall_intensity);

create table if not exists dim_time
(
    sk   int auto_increment
    primary key,

```

```

        date  date null,
        year  int  null,
        month int  null,
        day   int  null
);

create index month_key
    on dim_time (month);

create index year_key
    on dim_time (year);

create table if not exists dim_weather_condition
(
    sk          int auto_increment
        primary key,
    weather_code      int      null,
    weather_code_desc text     null,
    last_update       datetime null
);

create index weather_code_key
    on dim_weather_condition (weather_code);

create table if not exists fact_weather
(
    id          varchar(24)  not null
        primary key,
    temperature_avg  decimal(3, 1) null,
    humidity_avg     int      null,
    precipitation_sum decimal(4, 1) null,
    daylight_duration decimal(7, 2) null,
    daylight_percent decimal(4, 1) null,
    wind_speed       decimal(4, 1) null,
    wind_direction   int      null,
    time_sk          int      null,
    location_sk      int      null,
    rainfall_sk      int      null,
    weather_sk        int      null,
    last_update       datetime null,
    constraint fact_weather_dim_location_fk
        foreign key (location_sk) references dim_location (sk)
            on update cascade,
    constraint fact_weather_dim_rainfall_fk
        foreign key (rainfall_sk) references dim_rainfall (sk)
            on update cascade,
    constraint fact_weather_dim_time_fk
        foreign key (time_sk) references dim_time (sk)
            on update cascade,
    constraint fact_weather_dim_weather_condition_fk
        foreign key (weather_sk) references dim_weather_condition (sk)
            on update cascade
);

```

```

create index location_key
    on fact_weather (location_sk);

create index rainfall_key
    on fact_weather (rainfall_sk);

create index time_key
    on fact_weather (time_sk);

create index weather_key
    on fact_weather (weather_sk);

```

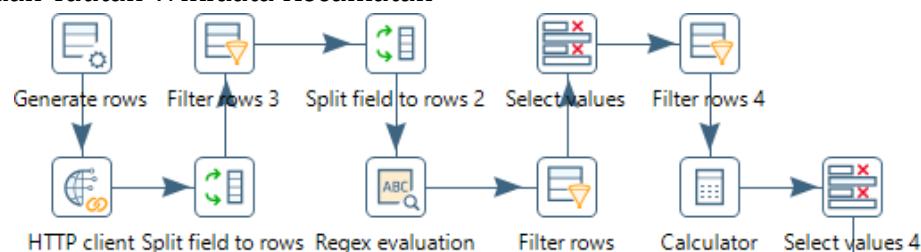
Gambar 4.9.Kode SQL Inisialisasi Basis Data OLAP

4.3. Ekstraksi Data dan Praproses Data

a. Data Kecamatan di Surabaya

Proses ekstraksi data kecamatan di Surabaya dan menyimpannya dalam basis data OLTP dibagi menjadi beberapa proses utama. Berikut adalah penjelasan dari setiap step yang digunakan:

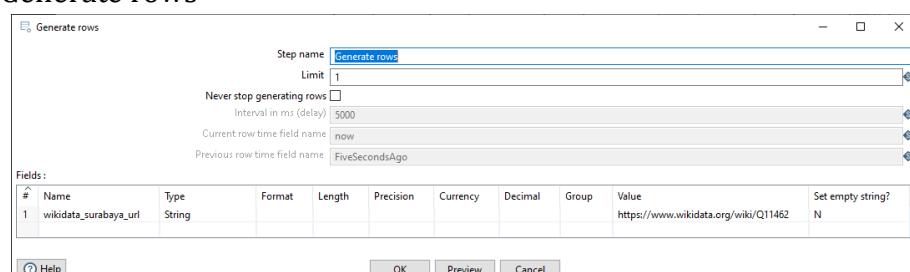
1. Nama dan Tautan Wikidata Kecamatan



Gambar 4.10. Rangkaian Step PDI untuk Ekstraksi Nama dan Tautan Wikidata Kecamatan

Gambar di atas menunjukkan proses ekstraksi data nama kecamatan di Surabaya yang bersumber dari Wikidata untuk masing-masing kecamatan. Berikut adalah penjelasan dari setiap step yang digunakan:

- Step “Generate rows”

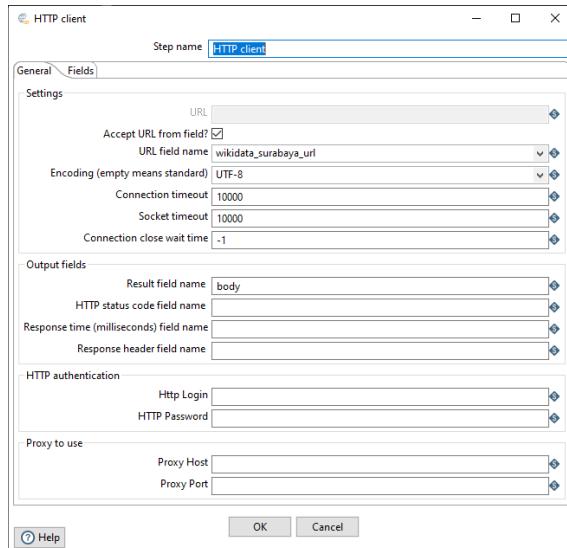


Gambar 4. 11. Konfigurasi Step “Generate Rows” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini digunakan untuk membuat baris data baru. Dalam hal ini, akan dibuat satu kolom baru bernama “wikidata_surabaya_url” yang bertipe data string dengan nilai yang dimasukkan adalah

[“https://www.wikidata.org/wiki/Q11462”](https://www.wikidata.org/wiki/Q11462). Tautan tersebut berisi profil data kota Surabaya yang juga memuat daftar kecamatan di kota tersebut.

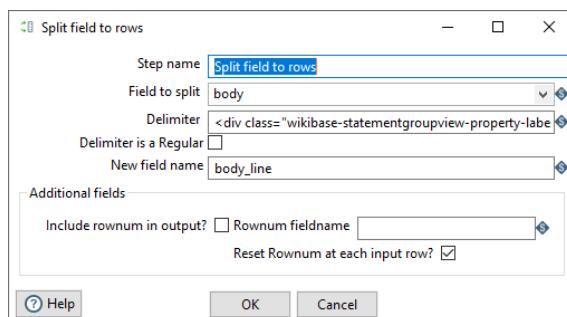
- Step “HTTP Client”



Gambar 4.12. Konfigurasi Step “HTTP Client” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini digunakan untuk melakukan *request* ke situs web agar dapat mengekstrak data dari situs web tersebut. Dalam hal ini, mengacu pada kolom “wikidata_surabaya_url” yang dibuat sebelumnya pada step “Generate rows”. Hasilnya akan disimpan dalam kolom (*field*) “body”.

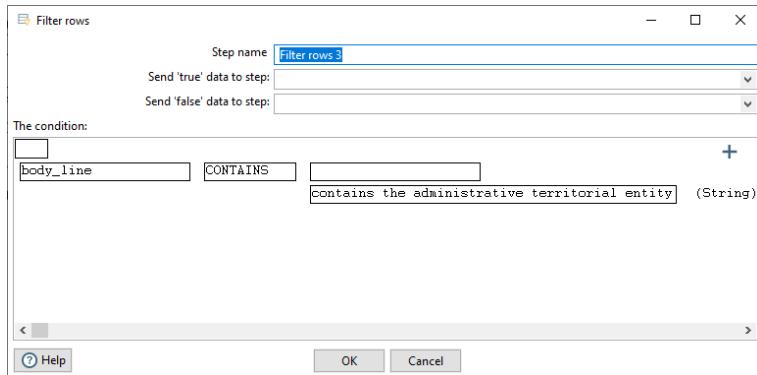
- Step “Split field to rows”



Gambar 4.13. Konfigurasi Step “Split field to rows” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step “Split field to rows” berfungsi untuk membagi sebuah *field* (dalam hal ini bernama “body”) menjadi beberapa baris, berdasarkan *delimiter* yang telah ditentukan, yaitu “<div class=“wikibase-statementgroupview-property-label”” dan menyimpannya dalam *field* baru bernama “body_line”.

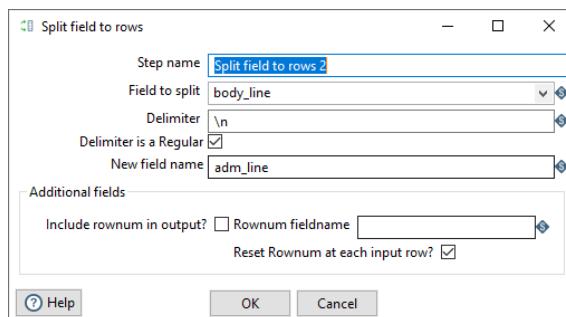
- Step “Filter rows 3”



Gambar 4.14. Konfigurasi Step “Filter rows 3” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam studi kasus ini, filter data berdasarkan nilai “contains the administrative territorial entity” berhubungan dengan karakteristik *metadata* kolom yang ingin difilter. Nilai tersebut merupakan nilai dalam kolom yang berhubungan dengan wilayah administratif.

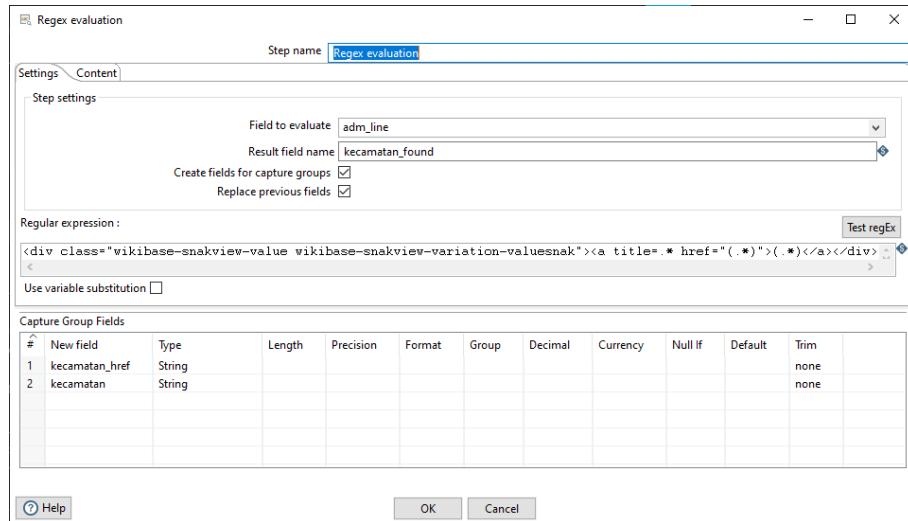
- Step “Split field to rows 2”



Gambar 4.15. Konfigurasi Step “Split field to rows 2” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini berfungsi untuk membagi sebuah *field* (dalam hal ini bernama “body_line”) menjadi baris, berdasarkan *delimiter* yang telah ditentukan, yaitu “\n” atau karakter yang merepresentasikan baris baru (*new line*) dan menyimpannya dalam *field* baru bernama “adm_line”.

- Step “Regex evaluation”

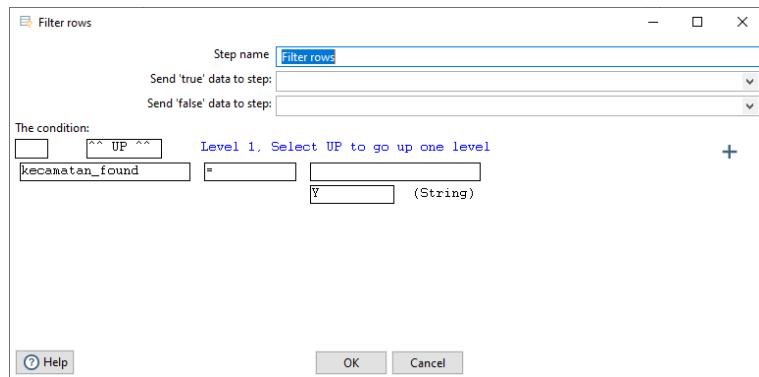


Gambar 4.16. Konfigurasi Step “Regex evaluation” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini digunakan untuk mengevaluasi apakah sebuah field cocok dengan pola tertentu *regular expression* (regex) dan mengekstrak bagian-bagian tertentu dari teks menggunakan *capture groups*. Dalam kasus ini, langkah tersebut dilakukan untuk teks yang terdapat dalam *field* “adm_line” dan hasilnya disimpan di *field* “kecamatan_found”. “Create Field for Capture Groups” dicentang agar dibuat *field* terpisah untuk setiap *capture group* yang ditemukan dalam pola regex.

Digunakan pola regex ‘`<div class="wikibase-snakview-value wikibase-snakview-variation-valuesnak">(.*)</div>`’. Pencarian pertama adalah *tag* `<div>` yang memiliki nilai “wikibase-snakview-value wikibase-snakview-variation-valuesnak”. Pencarian kedua adalah “``” yang mencari elemen `<a>` yang memiliki atribut title dengan nilai apapun. Akan diambil nilai dari ‘`href="(.*)"`’ yang menangkap nilai apapun dari atribut href sebagai nilai pertama. Kemudian, diambil nilai kedua dari “`<a>`”. Ini adalah teks yang akan muncul di dalam *tag* `<a>`. Hasilnya akan disimpan dalam dua kolom baru yaitu “kecamatan_href” dan “kecamatan”.

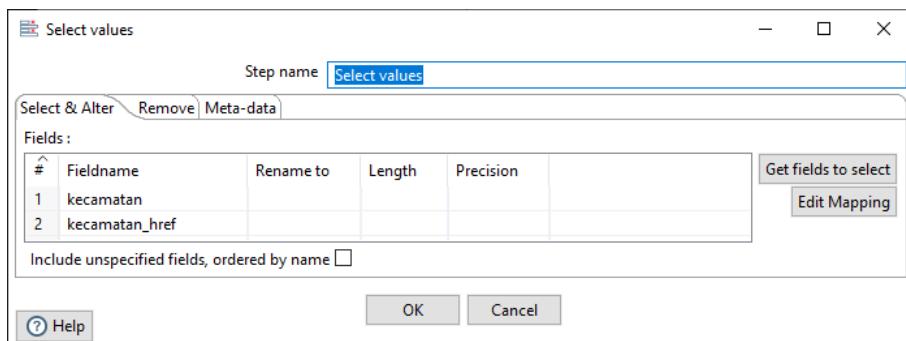
- Step “Filter rows”



Gambar 4.17.Konfigurasi Step “Filter rows” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam hal ini, data difilter berdasarkan “kecamatan_found = Y” yang berarti baris yang mengandung nilai “Y” di field “kecamatan_found” akan dipertahankan, sementara baris dengan nilai selain “Y” akan dibuang.

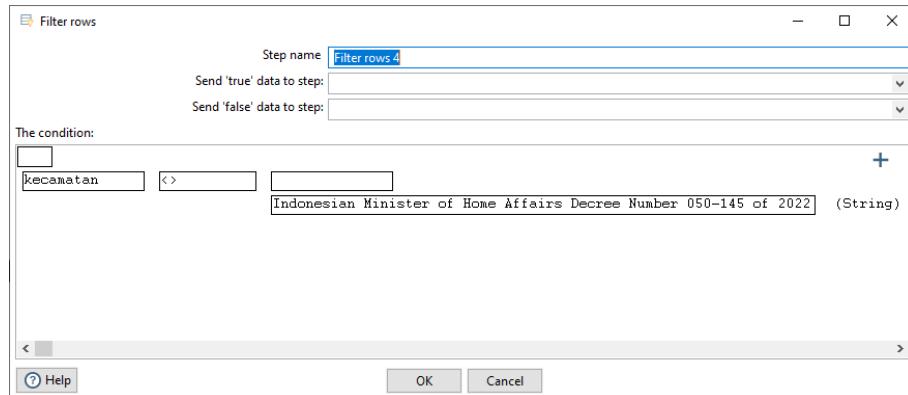
- Step “Select Values”



Gambar 4.18. Konfigurasi Step “Select Values” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini digunakan untuk memilih variabel atau *field* yang diinginkan dari semua *field*. Terdapat 2 *field* yang akan diambil yaitu “kecamatan” dan “kecamatan_href”. Sedangkan, *field* “kecamatan_found” tidak digunakan.

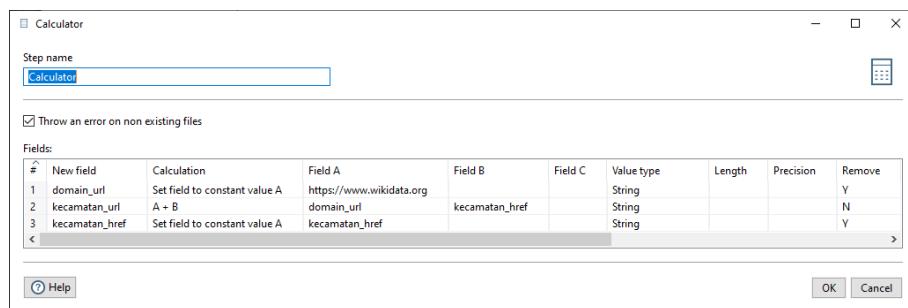
- Step “Filter rows 4”



Gambar 4.19. Konfigurasi Step “Filter rows 4” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam studi kasus ini, filter data berdasarkan nilai “Indonesian Minister of Home Affairs Decree Number 050-145 of 2022”. Dalam hal ini, filter ini akan menyaring semua baris dalam *field* kecamatan yang tidak sama dengan “Indonesian Minister of Home Affairs Decree Number 050-145 of 2022”. Baris yang mengandung nilai tersebut, akan dibuang.

- Step “Calculator”

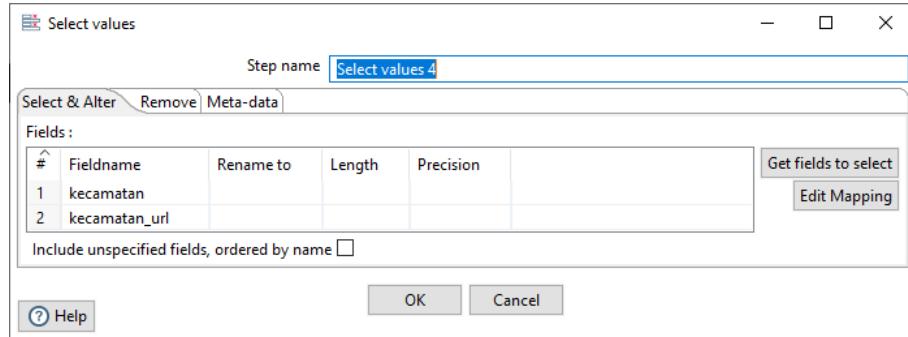


Gambar 4.20. Konfigurasi Step “Calculator” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini digunakan untuk membuat beberapa operasi atau perhitungan. Terdapat 3 operasi yang dilakukan dalam step ini, yaitu:

1. membuat *field* “domain_url” yang akan diisi dengan nilai tetap atau konstanta berupa “<https://www.wikidata.org>”,
2. membuat *field* “kecamatan_url” dengan menambahkan nilai dari *field* “domain_url” dengan nilai dari *field* “kecamatan_href”,
3. membuat *field* “kecamatan_href” yang akan diisi dengan nilai tetap atau konstanta dari *field* itu sendiri.

- Step “Select Values 4”



Gambar 4.21. Konfigurasi Step “Select Values 4” dalam Ekstraksi Nama dan Tautan Wikidata Kecamatan

Step ini digunakan untuk memilih variabel atau *field* yang diinginkan dari semua *field*. Terdapat 2 *field* yang akan dipilih, yakni “kecamatan” dan “kecamatan_url”.

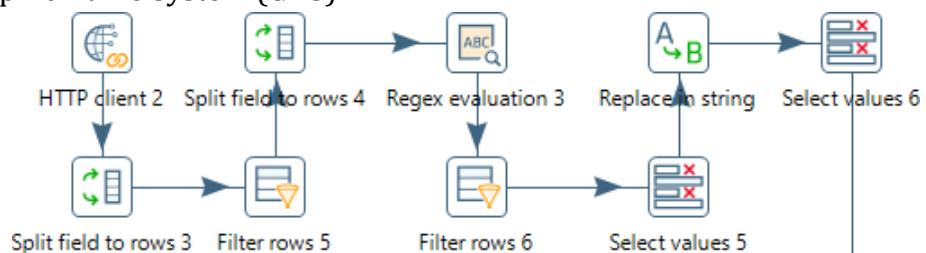
- Hasil

Examine preview data		
Rows of step: Select values 4 (31 rows)		
#	kecamatan	kecamatan_url
1	Tegalsari	https://www.wikidata.org/wiki/Q3298269
2	Genteng	https://www.wikidata.org/wiki/Q3298395
3	Bubutan	https://www.wikidata.org/wiki/Q3298203
4	Simokerto	https://www.wikidata.org/wiki/Q3298321
5	Pabean Cantian	https://www.wikidata.org/wiki/Q3298242

Gambar 4.22. Potongan Data Hasil Ekstraksi Nama dan Tautan Wikidata Kecamatan

Hasil pada proses ekstraksi pada bagian adalah nama kecamatan beserta dengan tautan Wikidata-nya.

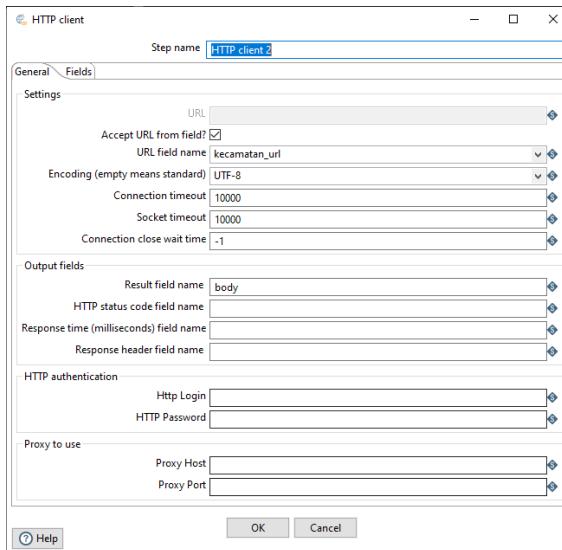
2. Geographic Name System (GNS)



Gambar 4.23. Rangkaian Step PDI untuk Ekstraksi Tautan GNS Kecamatan

Gambar di atas menunjukkan proses ekstraksi tautan GNS untuk masing-masing kecamatan. Berikut adalah penjelasan dari setiap step yang digunakan:

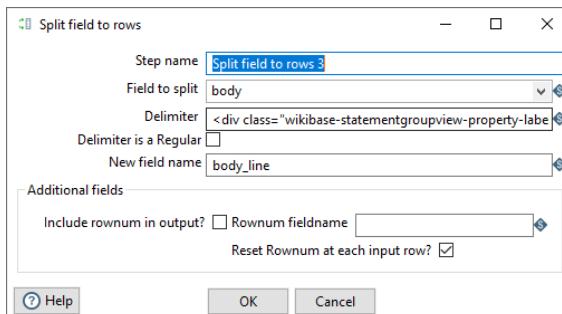
- Step “HTTP Client 2”



Gambar 4.24. Konfigurasi Step “HTTP Client 2” dalam Ekstraksi Tautan GNS Kecamatan

Step ini digunakan untuk melakukan *request* ke situs web agar dapat mengekstrak data dari situs web tersebut. Dalam hal ini, mengacu pada kolom “kecamatan_url” yang dibuat sebelumnya. Hasilnya akan disimpan dalam *field* “body”.

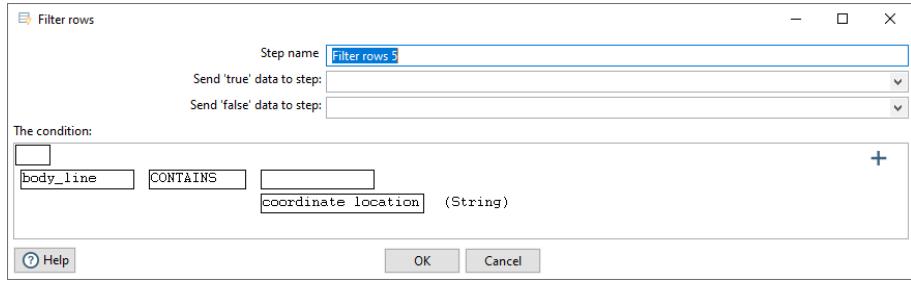
- Step “Split field to rows 3”



Gambar 4.25. Konfigurasi Step “Split field to rows 3” dalam Ekstraksi Tautan GNS Kecamatan

Step ini berfungsi untuk membagi sebuah *field* (dalam hal ini bernama “body”) menjadi beberapa baris, berdasarkan *delimiter* yang telah ditentukan, yaitu “<div class=“wikibase-statementgroupview-property-label”” dan menyimpannya dalam *field* baru bernama “body_line”.

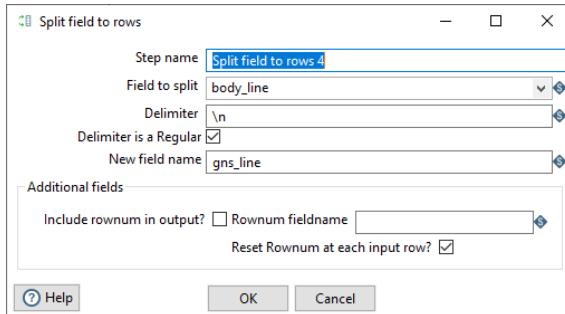
- Step “Filter rows 5”



Gambar 4.26. Konfigurasi Step “Filter rows 5” dalam Ekstraksi Tautan GNS Kecamatan

Step ini digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam hal ini, filter data yang digunakan adalah pada *field* “body_line” yang memiliki teks “coordinate location”.

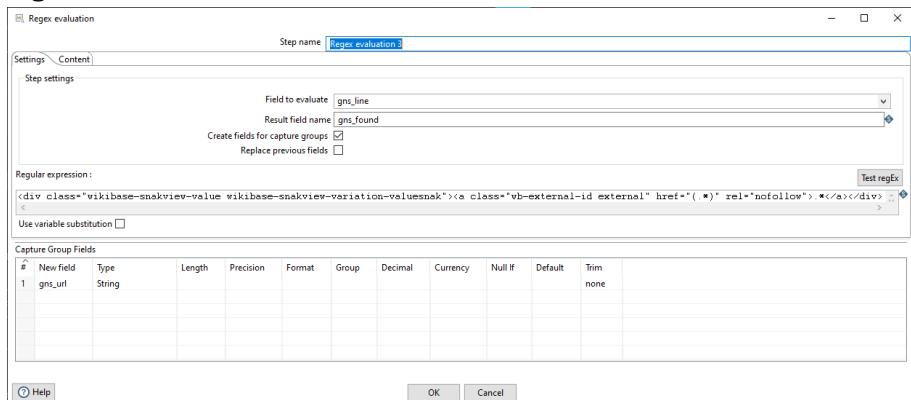
- Step “Split field to rows 4”



Gambar 4.27. Konfigurasi Step “Split field to rows 4” dalam Ekstraksi Tautan GNS Kecamatan

Step ini berfungsi untuk membagi sebuah *field* (dalam hal ini bernama “body_line”) menjadi beberapa baris, berdasarkan *delimiter* yang telah ditentukan, yaitu “\n” dan menyimpannya dalam *field* baru bernama “gns_line”.

- Step “Regex evaluation 3”



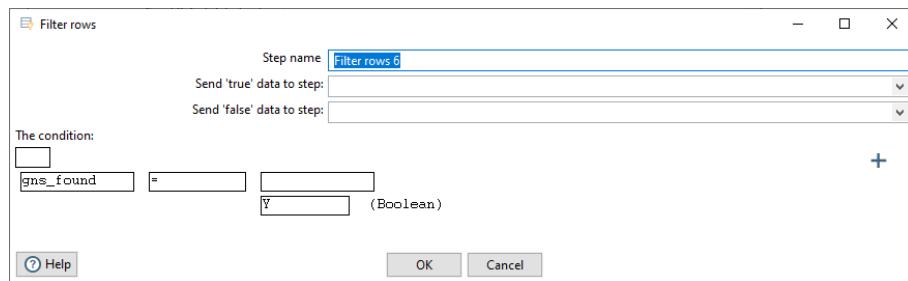
Gambar 4.28. Konfigurasi Step “Regex evaluation 3” dalam Ekstraksi Tautan GNS Kecamatan

Step ini digunakan untuk mengevaluasi apakah sebuah *field* cocok dengan pola regex dan mengekstrak bagian-bagian tertentu dari teks menggunakan *capture groups*. Dalam kasus ini, langkah tersebut dilakukan untuk *field* “gns_line” dan hasilnya disimpan di *field* “gns_found”. Create Field for Capture Groups akan

dicentang agar dibuat field terpisah untuk setiap capture group yang ditemukan dalam regex.

Digunakan regex '<div class="wikibase-snakview-value wikibase-snakview-variation-valuesnak">.*</div>'. Pencarian pertama adalah tag <div> yang memiliki nilai "wikibase-snakview-value wikibase-snakview-variation-valuesnak". Pencarian kedua adalah "<a class="wb-external-id external" yang mencari elemen <a> yang memiliki nama kelas tersebut. Akan diambil nilai dari "href="(.*)" yang menangkap nilai apapun dari atribut href. Dan lanjutannya ada pola yang disesuaikan. Hasilnya akan disimpan dalam kolom baru yaitu gns_url.

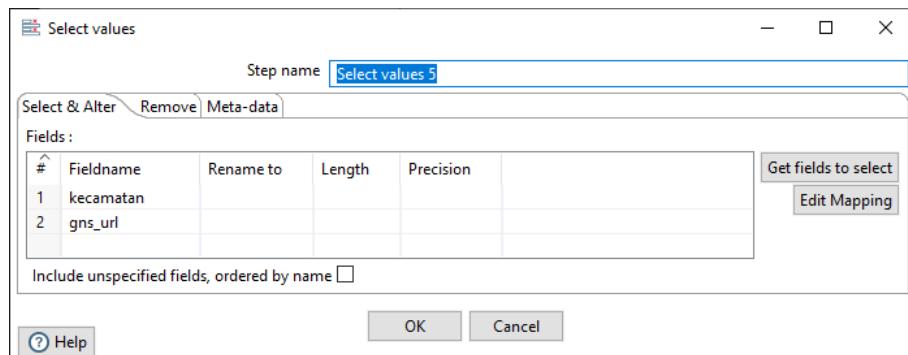
- Step “Filter rows 6”



Gambar 4.29. Konfigurasi Step “Filter rows 6” dalam Ekstraksi Tautan GNS Kecamatan

Step ini digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam studi kasus ini, filter data berdasarkan nilai dalam *field* “gns_found” yang sama dengan nilai “Y” (Boolean), jika tidak maka baris tersebut tidak digunakan.

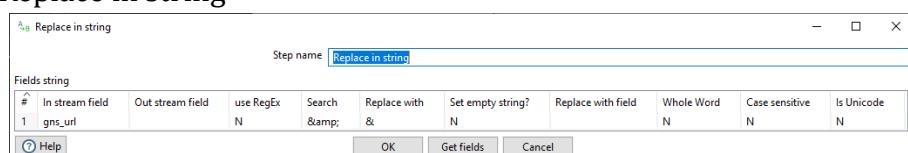
- Step “Select value 5”



Gambar 4.30. Konfigurasi Step “Select value 5” dalam Ekstraksi Tautan GNS Kecamatan

Step ini digunakan untuk memilih variabel atau *field* yang diinginkan dari semua *field*. Terdapat 2 *field* yang akan diambil yaitu “kecamatan” dan gns_url”.

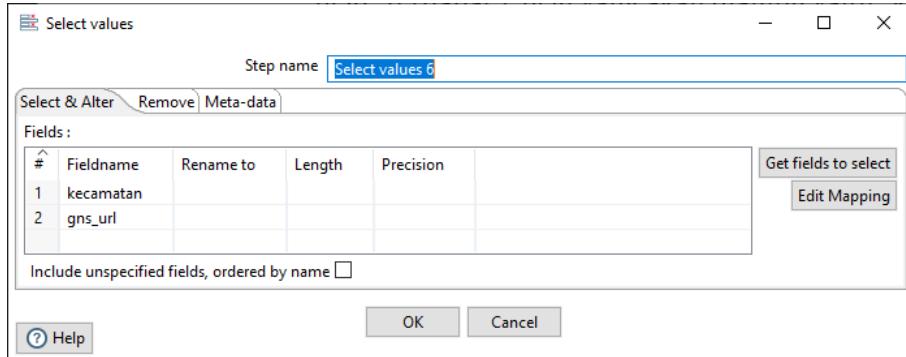
- Step “Replace in string”



Gambar 4.31. Konfigurasi Step “Replace in string” dalam Ekstraksi Tautan GNS Kecamatan

Step ini berfungsi untuk mengganti suatu nilai string ke nilai yang lain. Dalam hal ini, akan menggantikan setiap kemunculan karakter “&” dalam *field* gns_url dengan karakter *ampersand* “&” yang sebenarnya.

- Step “Select Values 6”



Gambar 4.32. Konfigurasi Step “Select Values 6” dalam Ekstraksi Tautan GNS Kecamatan

Step ini digunakan untuk memilih variabel atau *field* yang diinginkan dari semua *field*. Terdapat 2 *field* yang akan diambil yaitu “kecamatan” dan *gns_url*.

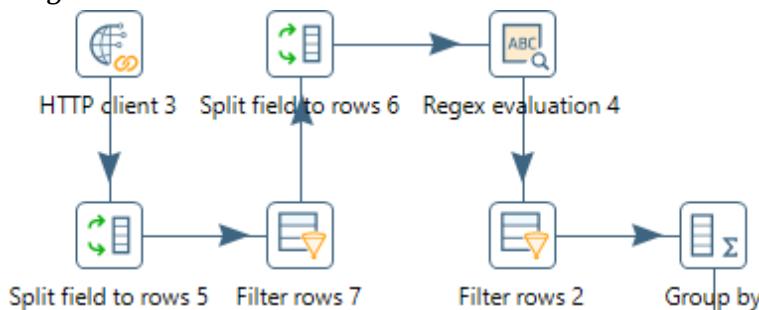
- Hasil

Examine preview data		
Rows of step: Select values 6 (31 rows)		
#	kecamatan	gns_url
1	Tegalsari	https://geonames.nga.mil/geo-ags/rest/services/RESEARCH/GIS_OUTPUT/MapServer/0/query?outFields=*&where=ufi+%3D+10290493
2	Genteng	https://geonames.nga.mil/geo-ags/rest/services/RESEARCH/GIS_OUTPUT/MapServer/0/query?outFields=*&where=ufi+%3D+10290492
3	Bubutan	https://geonames.nga.mil/geo-ags/rest/services/RESEARCH/GIS_OUTPUT/MapServer/0/query?outFields=*&where=ufi+%3D+10191204
4	Simokerto	https://geonames.nga.mil/geo-ags/rest/services/RESEARCH/GIS_OUTPUT/MapServer/0/query?outFields=*&where=ufi+%3D+14959337
5	Pabean Cantian	https://geonames.nga.mil/geo-ags/rest/services/RESEARCH/GIS_OUTPUT/MapServer/0/query?outFields=*&where=ufi+%3D+10191201

Gambar 4. 33. Potongan Data Hasil Ekstraksi Tautan GNS Kecamatan

Hasil pada proses ekstraksi pada bagian ini adalah nama kecamatan beserta dengan tautan GNS-nya.

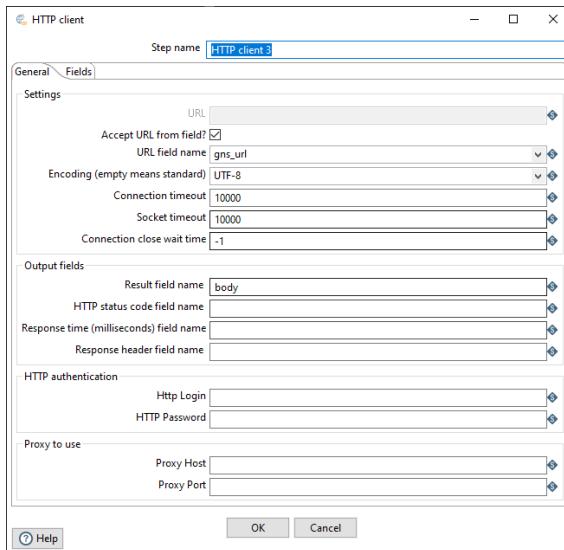
3. Latitude dan Longitude



Gambar 4.34. Rangkaian Step PDI untuk Ekstraksi Koordinat *Latitude* dan *Longitude* Kecamatan

Gambar di atas menunjukkan proses ekstraksi koordinat *latitude* dan *longitude* dari tautan GNS untuk masing-masing kecamatan. Berikut adalah penjelasan dari setiap step yang digunakan:

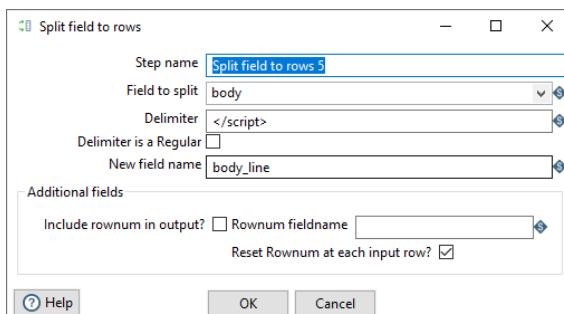
- Step “HTTP Client 3”



Gambar 4.35. Konfigurasi Step “HTTP Client 3” dalam Ekstraksi Koordinat *Latitude* dan *Longitude* Kecamatan

Step “HTTP client 3” digunakan untuk melakukan request ke situs web agar dapat mengekstrak data dari situs web. Dalam hal ini, mengacu pada kolom *gns_url* yang dibuat sebelumnya. Hasilnya akan disimpan dalam “body”.

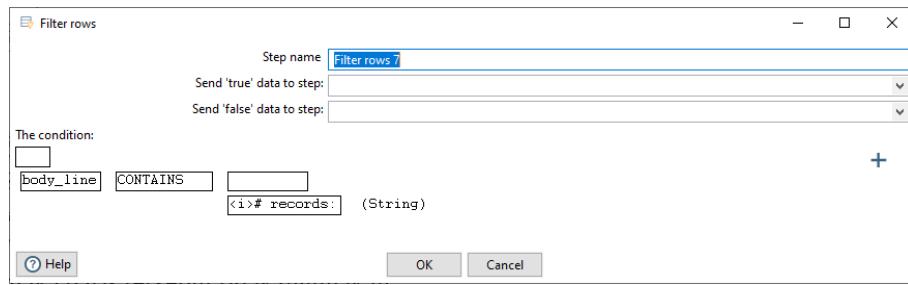
- Step “Split field to rows 5”



Gambar 4.36. Konfigurasi Step “Split field to rows 5” dalam Ekstraksi Koordinat *Latitude* dan *Longitude* Kecamatan

Step ini berfungsi untuk membagi sebuah *field* (dalam hal ini bernama “body”) menjadi beberapa baris, berdasarkan *delimiter* yang telah ditentukan, yaitu “</script>” dan menyimpannya dalam *field* baru bernama “body_line”.

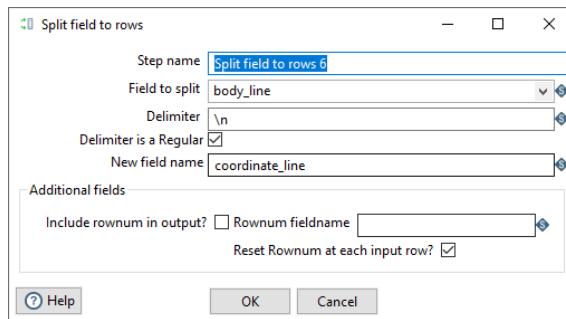
- Step “Filter rows 7”



Gambar 4.37. Konfigurasi Step “Filter rows 7” dalam Ekstraksi Koordinat *Latitude* dan *Longitude* Kecamatan

Step ini digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam studi kasus ini, filter data berdasarkan “body_line” mengandung nilai “<i># records:” atau tidak. Jika tidak mengandung, maka baris tersebut tidak digunakan.

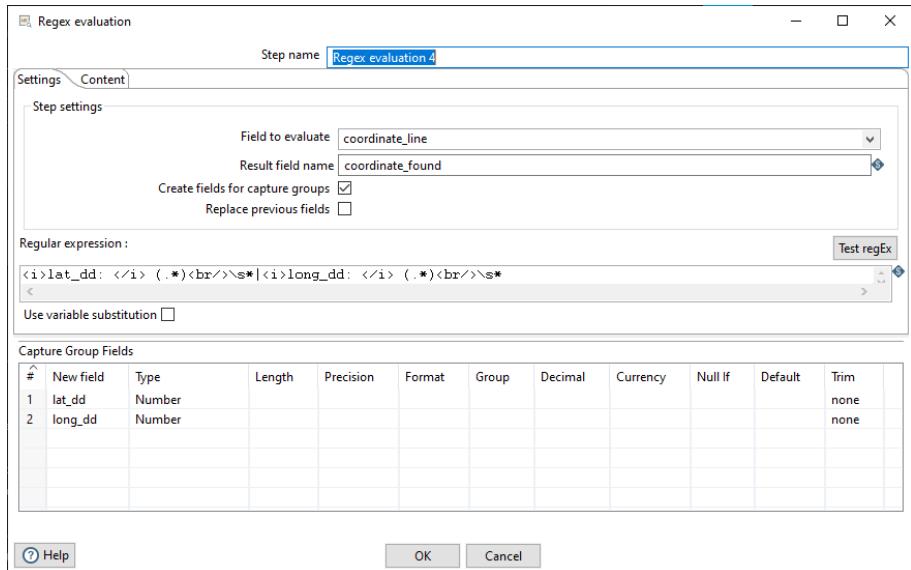
- Step “Split field to rows 6”



Gambar 4.38. Konfigurasi Step “Split field to rows 6” dalam Ekstraksi Koordinat *Latitude* dan *Longitude* Kecamatan

Step ini berfungsi untuk membagi sebuah field (dalam hal ini bernama “body_line”) menjadi beberapa baris, berdasarkan delimiter yang telah ditentukan yaitu “\n” dan menyimpannya dalam field baru bernama “coordinate_line”.

- Step “Regex evaluation 4”

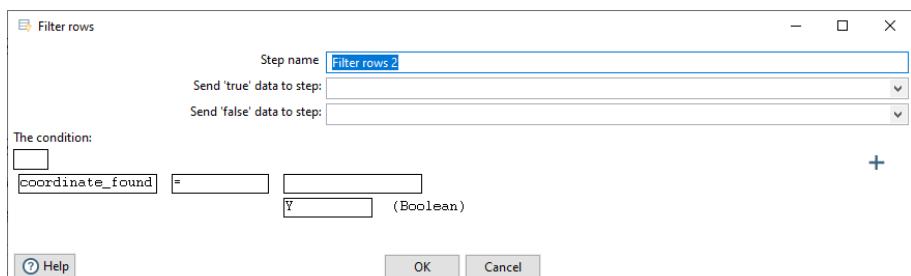


Gambar 4.39. Konfigurasi Step “Regex evaluation 4” dalam Ekstraksi Koordinat *Latitude* dan *Longitude* Kecamatan

Step ini digunakan untuk mengevaluasi apakah sebuah field cocok dengan pola regex dan mengekstrak bagian-bagian tertentu dari string menggunakan capture groups. Dalam kasus ini, langkah tersebut dilakukan untuk field “coordinate_line” dan hasilnya disimpan di field “coordinate_found”. Create Field for Capture Groups akan dicentang agar dibuat field terpisah untuk setiap capture group yang ditemukan dalam regex.

Digunakan regex “<i>lat_dd: </i> (.*)
\s*|<i>long_dd: </i> (.*)
\s*”. Regex akan membantu untuk mencari nilai dari latitude dan longitude kecamatan tersebut dengan pola yang sudah tertulis. Hasilnya akan disimpan dalam dua kolom baru yaitu lat_dd dan long_dd.

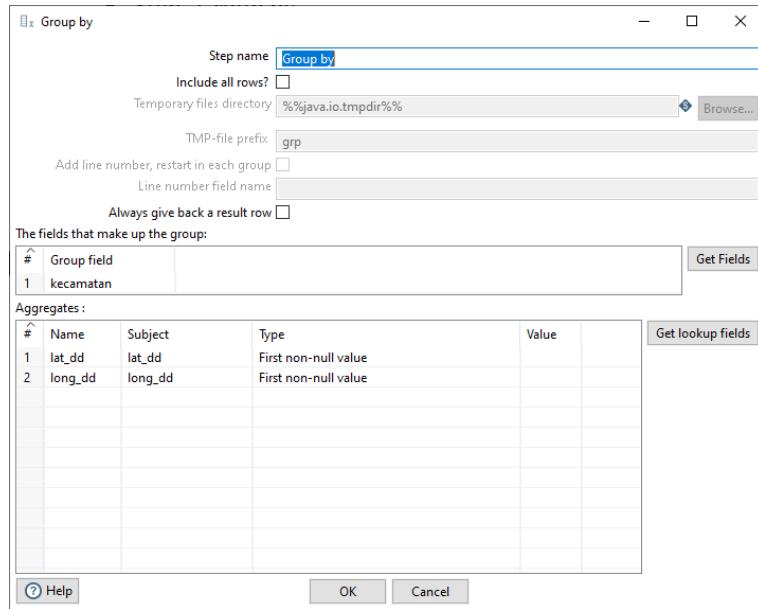
- Step “Filter rows 2”



Gambar 4.40. Konfigurasi Step “Filter rows 2” dalam Ekstraksi Koordinat *Latitude* dan *Longitude* Kecamatan

Step ini digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam studi kasus ini, filter data berdasarkan apakah terdapat nilai koordinat atau tidak di mana “coordinate_found” bernilai Y jika ada dan N jika tidak ada. Baris yang mengandung nilai N tersebut tidak digunakan.

- Step “Group by”



Gambar 4.41. Konfigurasi Step “Group by” dalam Ekstraksi Koordinat *Latitude* dan *Longitude* Kecamatan

Step ini dapat digunakan untuk mengelompokkan data berdasarkan suatu field tertentu. Dalam kasus ini, step Group By digunakan untuk mengelompokkan data berdasarkan kecamatan dan didapatkan nilai *latitude* dan *longitude*-nya yang merupakan bukan nilai *null* pertama.

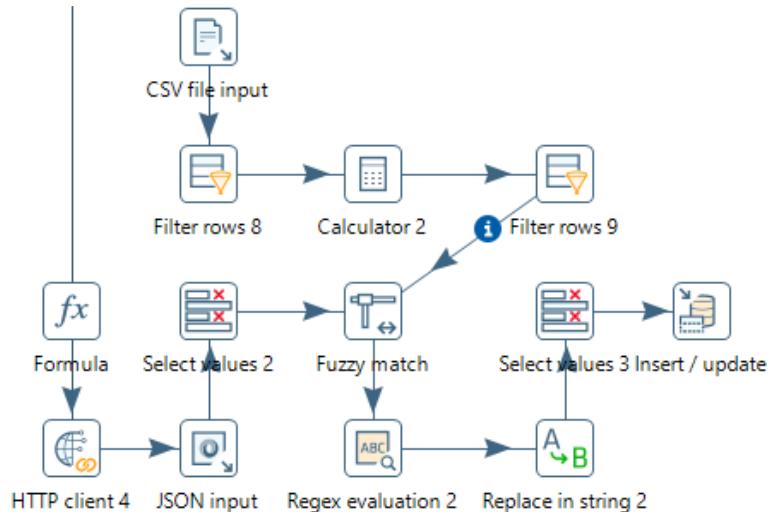
- Hasil

Examine preview data			
Rows of step: Group by (31 rows)			
#	kecamatan	lat_dd	long_dd
1	Tegalsari	-7.275556	112.733056
2	Genteng	-7.259167	112.745
3	Bubutan	-7.248889	112.727778
4	Simokerto	-7.239444	112.753333
5	Pabean Cantian	-7.214722	112.731944

Gambar 4.42. Potongan Data Hasil Ekstraksi Koordinat *Latitude* dan *Longitude* Kecamatan

Hasil pada proses ekstraksi pada bagian ini adalah nama kecamatan beserta dengan koordinat *latitude* dan *longitude*-nya.

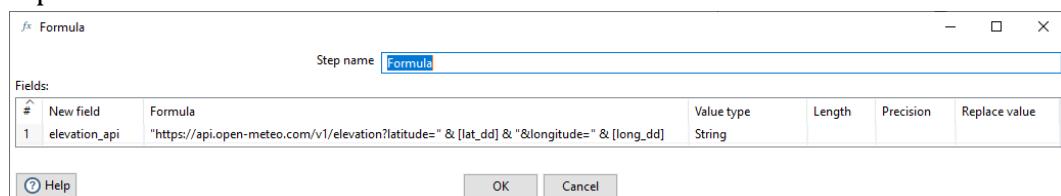
4. Elevasi dan Kode Wilayah



Gambar 4.43. Rangkaian Step PDI untuk Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Gambar di atas menunjukkan proses ekstraksi koordinat *latitude* dan *longitude* dari tautan GNS untuk masing-masing kecamatan. Berikut adalah penjelasan dari setiap step yang digunakan:

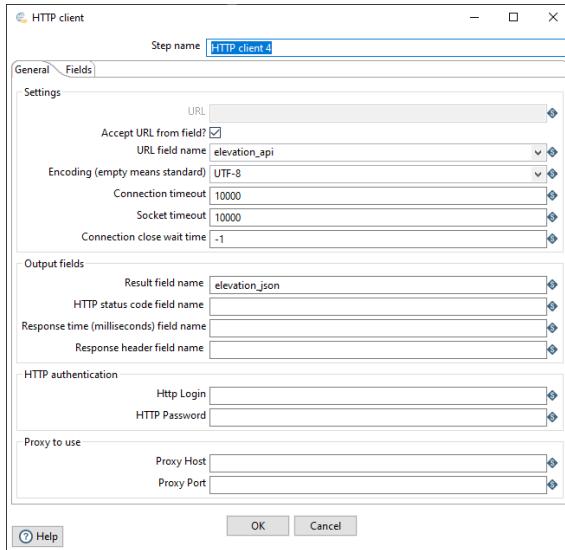
- Step “Formula”



Gambar 4.44. Konfigurasi Step “Formula” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini digunakan untuk membuat *field* baru bernama “elevation_api” yang memiliki nilai “<https://api.open-meteo.com/v1/elevation?latitude=> & [lat_dd] & “[longitude=](https://api.open-meteo.com/v1/elevation?longitude=) & [long_dd]”. Step ini berisikan tautan API Open-Meteo dan menggunakan nilai *latitude* dan *longitude* pada step sebelumnya.

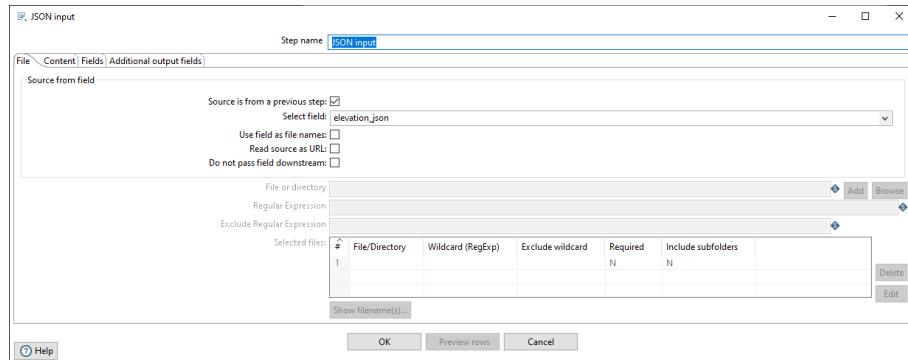
- Step “HTTP Client 4”



Gambar 4.45. Konfigurasi Step “HTTP Client 4” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

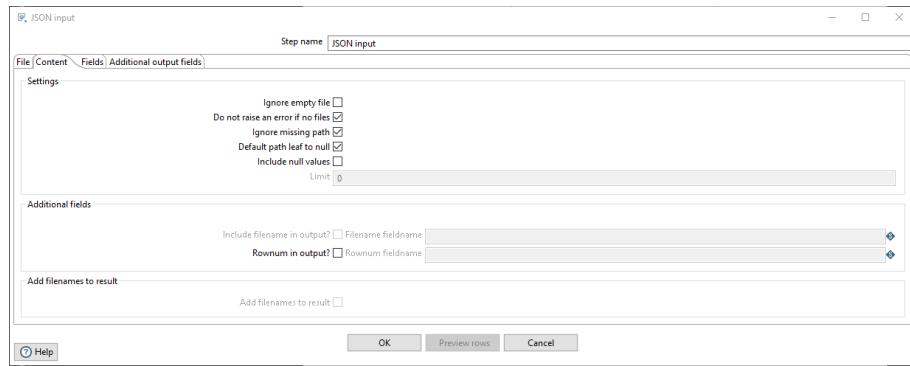
Step ini digunakan untuk melakukan *request* ke tautan agar dapat mengekstrak data dari situs web. Dalam hal ini, mengacu pada kolom “elevation_api” yang dibuat sebelumnya. Hasilnya akan disimpan dalam “elevation_json”.

- Step “JSON Input”



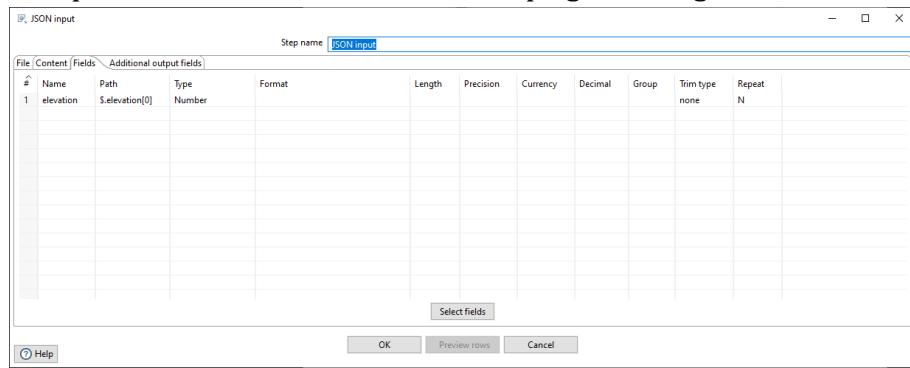
Gambar 4.46. Konfigurasi Step “JSON Input” pada tab “File” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini digunakan untuk mendefinisikan data *input* berbentuk JSON. *Field* yang digunakan adalah “elevation_json” yang didapatkan dari step “HTTP Client” sebelumnya.



Gambar 4.47. Konfigurasi Step “JSON Input” pada tab “Content” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

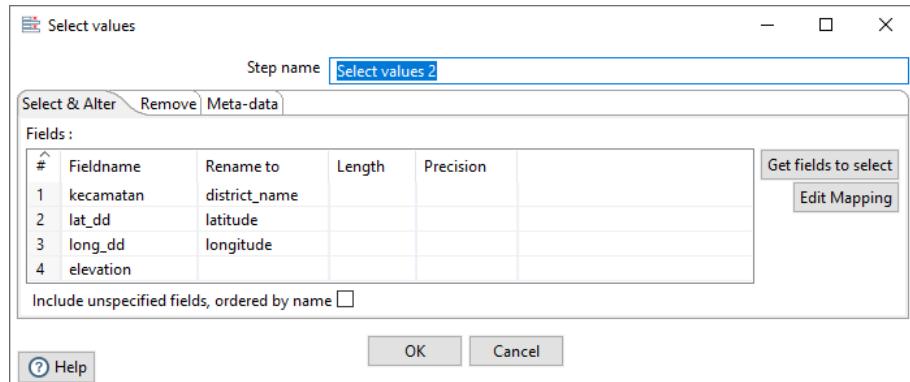
Pada tab “content”, centang pada bagian “Do not raise an error if no files”, “Ignore missing path, dan “default path leaf to null”. Sehingga ketika tidak ada *file* atau path, tidak perlu memunculkan nilai error tapi, ganti dengan nilai *null*.



Gambar 4.48. Konfigurasi Step “JSON Input” pada tab “Fields” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Pada tab “Fields” didefinisikan *field* yang diambil yaitu “elevation”. Definisikan juga path, tipe data, dan format untuk tiap field yang diambil. Path yang dituliskan mengacu pada hasil dari ekstraksi data menggunakan api pada step sebelumnya.

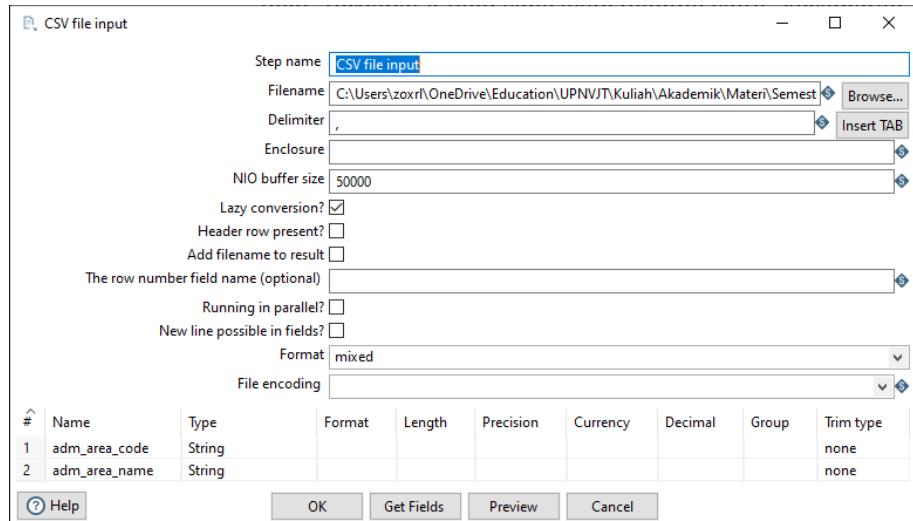
- Step “Select Values 2”



Gambar 4.49. Konfigurasi Step “Select Values 2” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini digunakan untuk memilih variabel atau field yang diinginkan dari semua field. Terdapat 4 field yang akan diambil yaitu “kecamatan” yang diubah menjadi “district_name”, “lat_dd” yang diubah menjadi “latitude”, “long_dd” yang diubah menjadi longitude, dan “elevation”.

- Step “CSV file input”



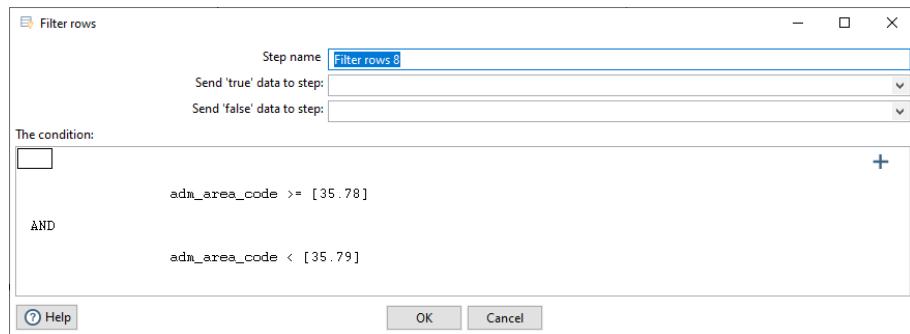
Gambar 4.50. Konfigurasi Step “CSV file input” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini berfungsi untuk menerima masukan (*input*) data berbentuk CSV. Dalam hal ini, akan digunakan file bernama “base.csv” yang berisi kode suatu daerah dari provinsi, kabupaten, hingga ke kecamatan untuk seluruh daerah di Indonesia. *Delimiter* yang dipilih adalah tanda koma (,). Bentuk data CSV-nya adalah berikut.

Examine preview data		
Rows of step: CSV file input (1000 rows)		
#	adm_area_code	adm_area_name
1	11	ACEH
2	11.01	KAB. ACEH SELATAN
3	11.01.01	Bakongan
4	11.01.01.2001	Keude Bakongan
5	11.01.01.2002	Ujung Mangki

Gambar 4.51. Bentuk Data File CSV Kode Wilayah di Indonesia (base.csv)

- Step “Filter rows 8”



Gambar 4.52. Konfigurasi Step “Filter rows 8” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam studi kasus ini, terdapat dua filter yaitu nilai dari adm_area_code lebih dari sama dengan 35.78 dan kurang dari 35.79. Hal ini dikarenakan 35.78 adalah kode kota Surabaya.

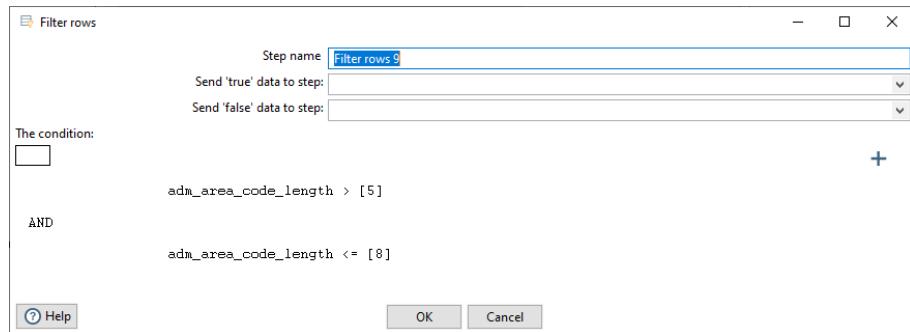
- Step “Calculator 2”



Gambar 4.53. Konfigurasi Step “Calculator 2” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini dilakukan untuk membuat beberapa operasi atau perhitungan. Dalam hal ini, akan dihitung panjang dari kode yang diberi nama “adm_area_code_length” dengan kalkulasinya adalah “Return the length of a string A” dari kolom “adm_area_code”.

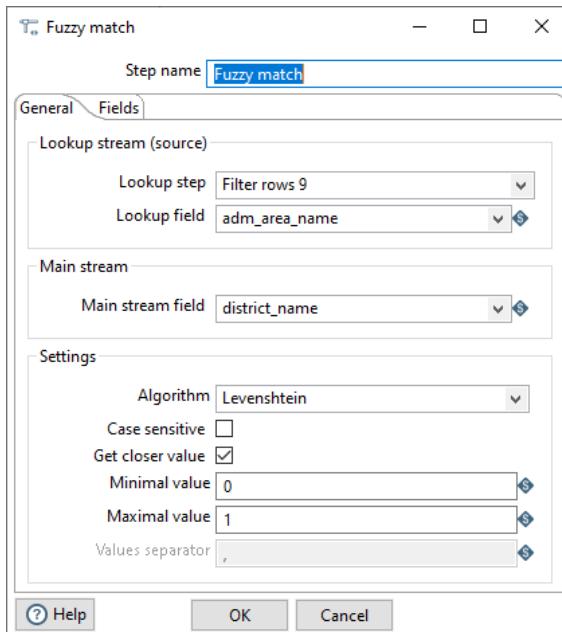
- Step “Filter rows 9”



Gambar 4.54. Konfigurasi Step “Filter rows 9” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam studi kasus ini, terdapat dua filter yaitu nilai dari “adm_area_code_length” lebih dari 5 dan kurang dari sama dengan 8. Hal ini dikarenakan kode area yang memiliki kisaran tersebut adalah kode untuk daerah kecamatan.

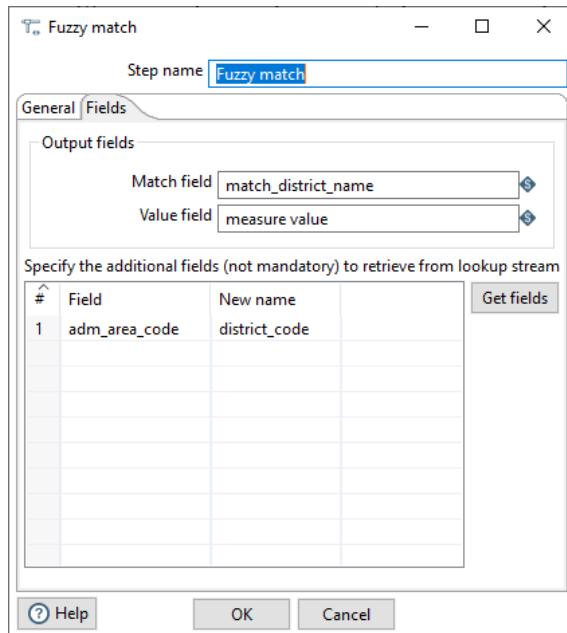
- Step “Fuzzy match”



Gambar 4.55. Konfigurasi Step “Fuzzy match” pada tab “general” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini digunakan untuk mencari kesamaan atau kecocokan yang mendekati (fuzzy) antara dua field atau kolom yang memiliki nilai yang mirip tetapi mungkin tidak identik. Dalam hal ini, step ini digunakan untuk menemukan kemiripan antara nama area administratif dan nama distrik untuk memastikan bahwa data yang disalin atau digabungkan tidak memiliki duplikasi atau ketidaksesuaian meskipun penulisannya sedikit berbeda.

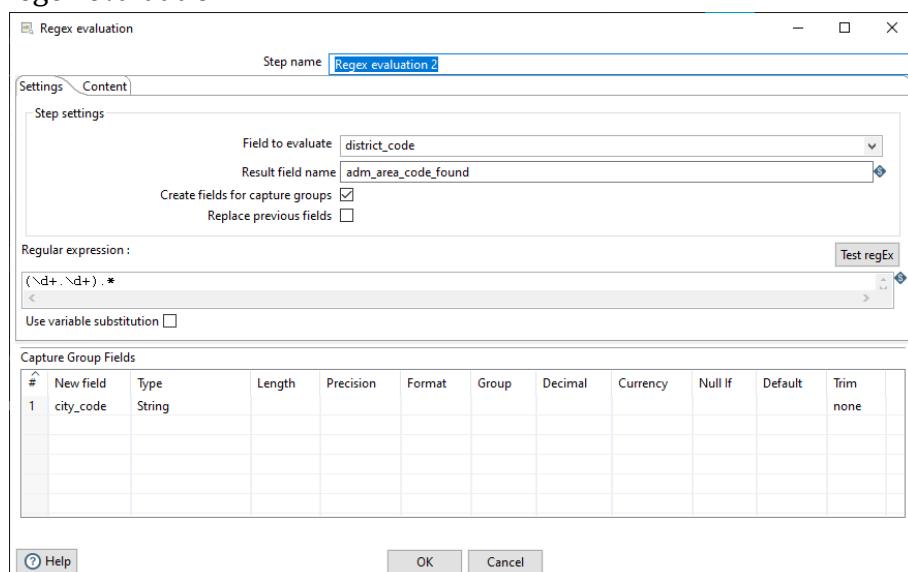
Pada tab “General”, lookup step memiliki nilai “Filter rows 9”, lookup field memiliki nilai “adm_area_name” dan Main stream field adalah “district_name”. Jadi data yang akan dilakukan pengecekan adalah data dari step “Filter rows 9” dimana nilai yang dicocokkan adalah “district_name” dan “adm_area_name”.



Gambar 4.56. Konfigurasi Step “Fuzzy match” pada tab “general” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Pada tab “Fields”, Match Field dengan nilai “match_district_name” adalah nama baru yang akan diberikan untuk hasil pencocokan antara district_name dan adm_area_name setelah proses fuzzy match selesai. Value Field dengan nilai “measure_value” adalah field yang berisi nilai atau data yang relevan yang akan disertakan setelah pencocokan fuzzy antara district_name dan adm_area_name. Additional Field yaitu “adm_area_code” dengan nama baru “district_code” adalah field tambahan yang disertakan dalam hasil setelah pencocokan fuzzy.

- Step “Regex evaluation 2”

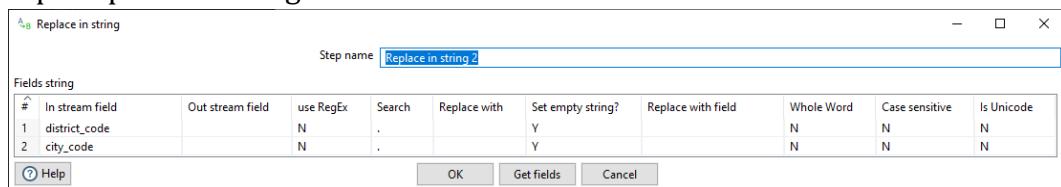


Gambar 4.57. Konfigurasi Step “Regex evaluation 2” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini digunakan untuk mengevaluasi apakah sebuah field cocok dengan pola regex dan mengekstrak bagian-bagian tertentu dari string menggunakan capture groups. Dalam kasus ini, langkah tersebut dilakukan untuk field “district_code” dan hasilnya disimpan di field “adm_area_code_found”. Create Field for Capture Groups akan dicentang agar dibuat field terpisah untuk setiap capture group yang ditemukan dalam regex.

Digunakan regex “(\d+\.\d+).*” untuk mencocokkan angka desimal pertama yang ditemukan dalam sebuah string dan mengambilnya. Setelah itu, bagian .* memungkinkan untuk menangkap sisa teks apapun setelah angka desimal tersebut. Hasilnya akan disimpan dalam kolom baru yaitu city_code. Hal ini digunakan untuk mengambil kode kota Surabaya yang merupakan 4 angka awal dengan titik sebagai penyambung yaitu 35.78.

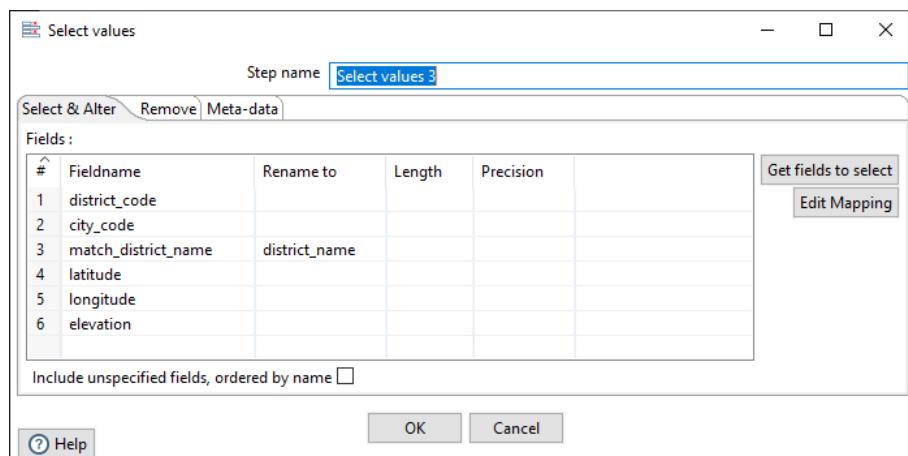
- Step “Replace in string 2”



Gambar 4.58. Konfigurasi Step “Replace in string 2” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

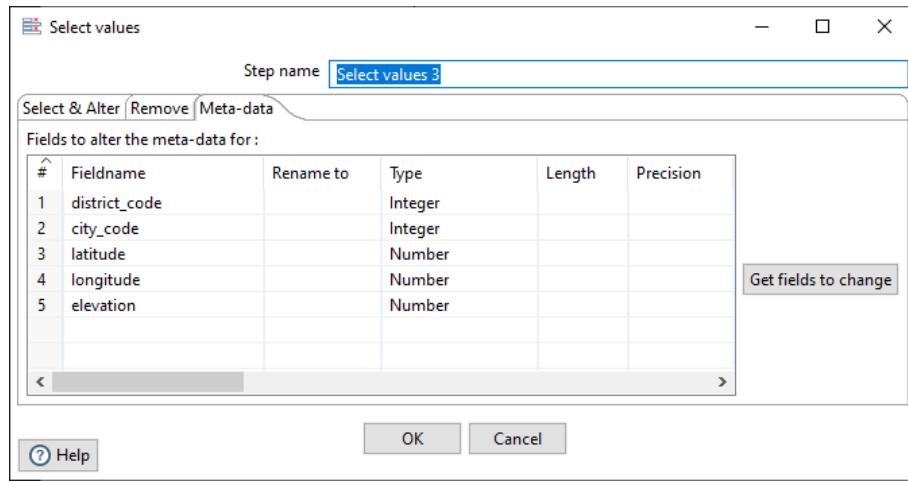
Step ini berfungsi untuk mengganti suatu nilai string ke nilai yang lain. Dalam hal ini, pada field “district_code” dan “city_code” akan menggantikan nilai titik (.) menjadi nilai kosong sehingga kodennya bersambung.

- Step “Select values 3”



Gambar 4.59. Konfigurasi Step “Select values 3” pada tab “Select & Alter” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

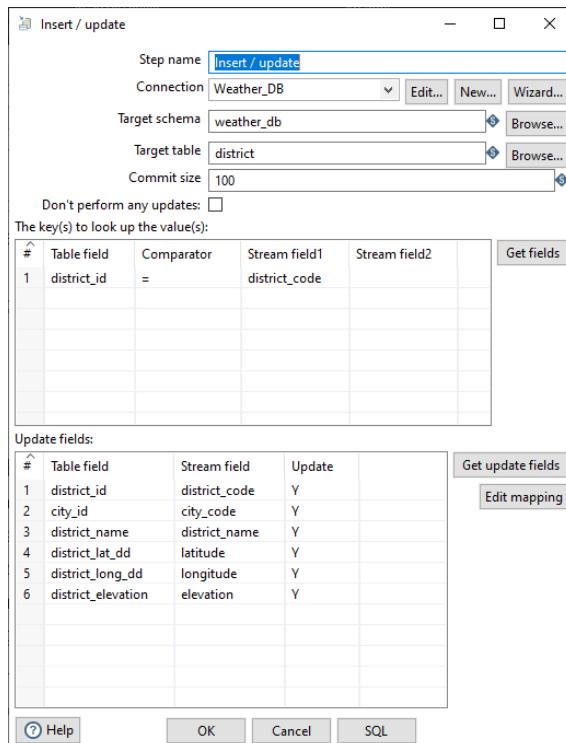
Step ini digunakan untuk memilih variabel atau field yang diinginkan dari semua field. Pada tab “Select & Alter”, didefinisikan 6 field yang akan diambil yaitu “district_code”, “city_code”, “match_district_name” yang diganti nama menjadi “district_name”, “latitude”, “longitude”, dan “elevation”.



Gambar 4.60. Konfigurasi Step “Select values 3” pada tab “Select & Alter” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Pada tab “Meta-data” definisikan tipe – tipe dari tiap field. Field “district_code” dan “city_code” bertipe data integer, sedangkan field “latitude”, “longitude”, dan “elevation” bertipe data Number. Isikan nilai “N” pada “Binary to Normal?”, “Date Format Lenient?”, dan “Lenient Number conversion?”.

- Step “Insert/update”



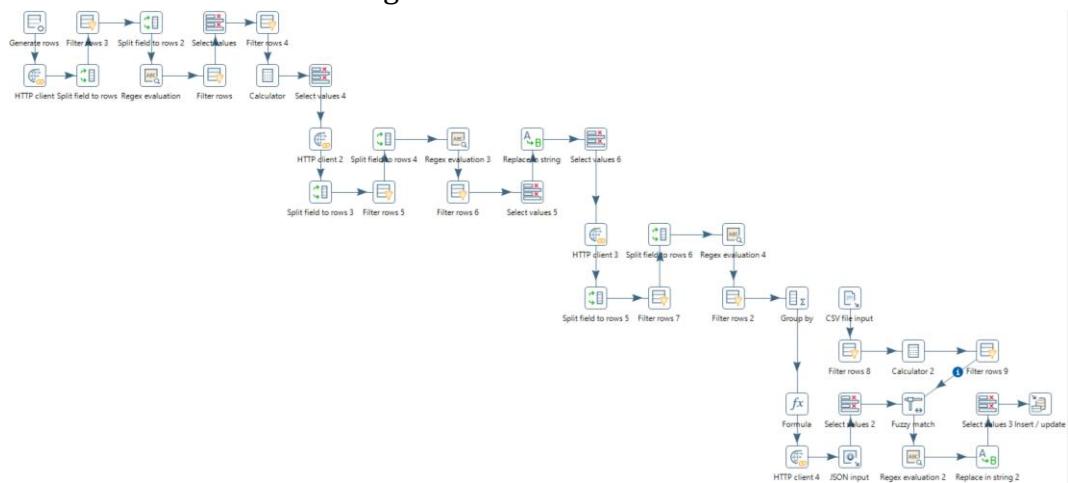
Gambar 4.61. Konfigurasi Step “Insert/update” dalam Ekstraksi Elevasi dan Kode Wilayah Kecamatan

Step ini digunakan untuk menghubungkan data yang ada dengan database dan memperbarui atau menyisipkan data baru ke dalam tabel yang sudah ada, berdasarkan kunci yang ditentukan. Dalam hal ini, database yang digunakan adalah weather_db dengan target table district. Kunci utama untuk

mengidentifikasi apakah baris yang ada sudah ada sebelumnya di dalam tabel adalah district_id dengan district_code. Artinya, jika ada kode yang sama, maka data yang baru akan digunakan untuk update data yang lama. Jika district_code tidak ada, maka data baru akan dimasukkan (insert). Field yang diinputkan adalah “district_id”, “city_id”, “district_name”, “district_lat_dd”, “district_long_dd”, dan “district_elevation” yang disesuaikan dengan nama field dari proses.

- Bentuk Lengkap *Transformation*

Gambar di bawah ini menunjukkan *transformation* secara menyeluruh dalam proses ekstraksi data kecamatan di Surabaya yang terdapat dalam canvas atau work area Pentaho Data Integration:



Gambar 4.62. Keseluruhan Rangkaian Step PDI untuk Ekstraksi Data Kecamatan

- Hasil

Rows of step: Select values 3 (31 rows)						
#	district_code	city_code	district_name	latitude	longitude	elevation
1	357805	3578	Tegalsari	-7.275556	112.733056	6.0
2	357807	3578	Genteng	-7.259167	112.745	7.0
3	357813	3578	Bubutan	-7.248889	112.727778	5.0
4	357811	3578	Simokerto	-7.239444	112.753333	6.0
5	357812	3578	Pabean Cantian	-7.214722	112.731944	5.0

Gambar 4.63. Potongan Data Hasil Ekstraksi Data Kecamatan

Hasil pada proses ekstraksi pada bagian ini adalah kode wilayah kecamatan, kode wilayah kota, nama kecamatan, koordinat *latitude* dan *longitude* kecamatan, serta elevasi kecamatan.

5. Hasil dalam Basis Data OLTP

The screenshot shows a database interface with a query results window. The query is:

```

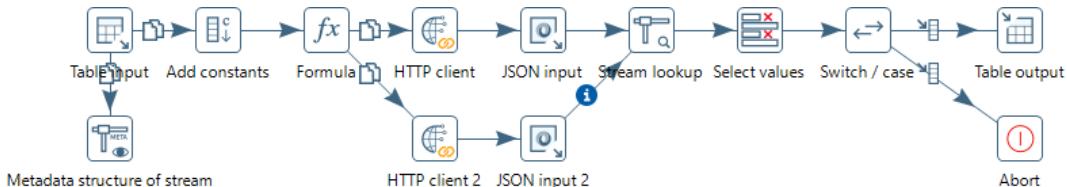
SELECT
    district_id,
    city_id,
    district_name,
    district_lat_dd,
    district_long_dd,
    district_elevation,
    created,
    updated
FROM
    weather_db.district
ORDER BY
    district_id
  
```

The results table has 8 columns and 31 rows. The columns are: district_id, city_id, district_name, district_lat_dd, district_long_dd, district_elevation, created, and updated. The data includes various districts like Karang Pilang, Muncola, Sungut, etc., with their respective coordinates and timestamps.

Gambar 4.64. Hasil Ekstraksi Data Kecamatan dalam Basis Data OLTP

Potongan data dari tabel “district” tampak pada gambar di atas, di mana tabel “district” memiliki 8 kolom dengan 31 baris. Kolom “created” dan “updated” pada tabel ini berguna untuk memantau waktu kapan data dibuat ataupun diperbarui.

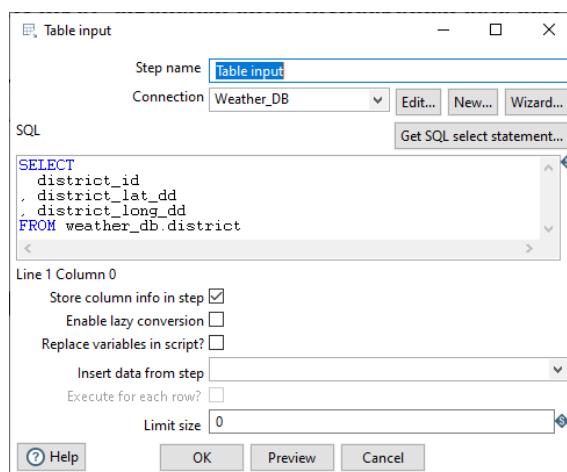
b. Data Historis Cuaca



Gambar 4.65. Rangkaian Step PDI untuk Ekstraksi Data Historis Cuaca

Gambar di atas menunjukkan proses ekstraksi data historis cuaca dan menyimpannya dalam basis data OLTP. Berikut adalah penjelasan dari setiap step yang digunakan:

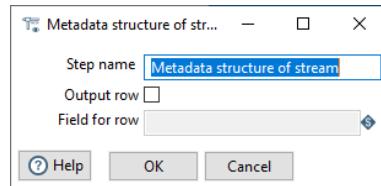
1. Table Input



Gambar 4.66. Konfigurasi Step “Table Input” dalam Ekstraksi Data Historis Cuaca

Step “Table input” digunakan untuk mendapatkan data dari bentuk tabel. Dalam hal ini, tabel berada dalam basis data OLTP dengan nama tabel “district”. Diambil tiga variabel yaitu “district_id”, “district_lat_dd”, dan “district_long_dd”.

2. Metadata Structure of stream



Gambar 4.67. Konfigurasi Step “Metadata Structure of stream” dalam Ekstraksi Data Historis Cuaca

Step “Metadata Structure of Stream” berfungsi untuk memberikan informasi atau struktur *metadata* terkait aliran data yang sedang diproses dalam suatu pipeline atau *workflow*. Step ini dapat menampilkan atau mendeteksi secara otomatis *metadata* dari aliran data yang ada di *pipeline*.

Rows of step: Metadata structure of stream (3 rows)							
#	Position	Fieldname	Comments	Type	Length	Precision	Origin
1	1	district_id	district_id	Integer	9	0	Table input
2	2	district_lat_dd	district_lat_dd	Number	11	8	Table input
3	3	district_long_dd	district_long_dd	Number	11	8	Table input

Gambar 4.68. Output Step "Metadata Structure of stream" dalam Ekstraksi Data Historis Cuaca

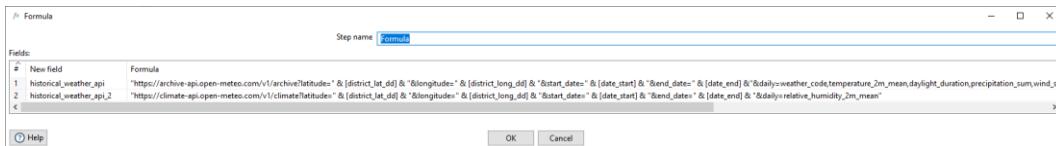
3. Add Constants

Add constants										
Step name [Add constants]										
Fields:										
#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	date_start	String	yyyy-MM-dd						2000-01-01	N
2	date_end	String	yyyy-MM-dd						2024-11-30	N

Gambar 4.69. Konfigurasi Step "Add Constants" dalam Ekstraksi Data Historis Cuaca

Step “Add Constant” digunakan untuk menambahkan nilai konstan (tetap) ke dalam aliran data yang sedang diproses. Terdapat dua kolom baru yang ditambahkan yaitu “date_start” dan “date_end” yang menggunakan tipe data string dengan format yyyy-MM-dd. Nilai untuk “date_start” adalah “2000-01-01” dan nilai untuk “date_end” adalah “2024-11-30”. Hal ini karena data yang ingin di ekstrak berkisar dari tanggal 1 Januari 2000 hingga 30 November 2024.

4. Formula

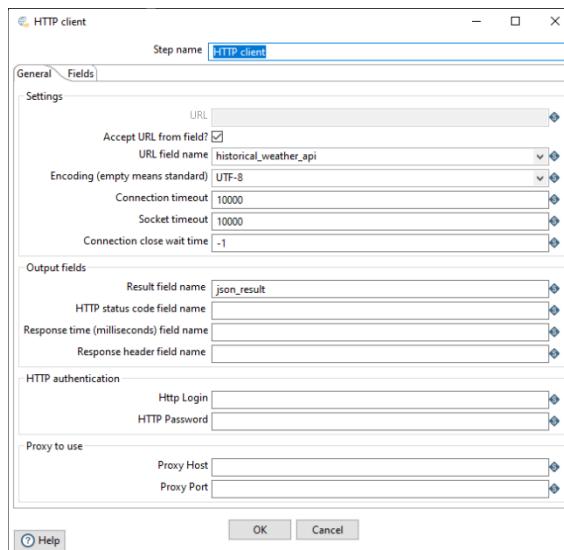


Gambar 4.70. Konfigurasi Step "Formula" dalam Ekstraksi Data Historis Cuaca

Step ini digunakan untuk membuat kolom baru atau memodifikasi kolom yang sudah ada dengan menggunakan formula atau ekspresi tertentu. Dalam hal ini, dibuat dua kolom baru bernama “historical_weather_api” dan “historical_weather_api_2” dengan isi dari formula adalah tautan API yang bisa didapatkan dari open-meteo. Perbedaan antara kedua api adalah api pertama nilai “weather_code”, “temperature_2m_means”, “daylight_duration”, “precipitation_sum”, dan “wind_s”. Sedangkan, api kedua mengambil nilai “relative_humidity_2m_mean”. Variabel “date_start” dan “date_end” diambil dari step sebelumnya.

5. Jalur 1 (HTTP client & JSON input)

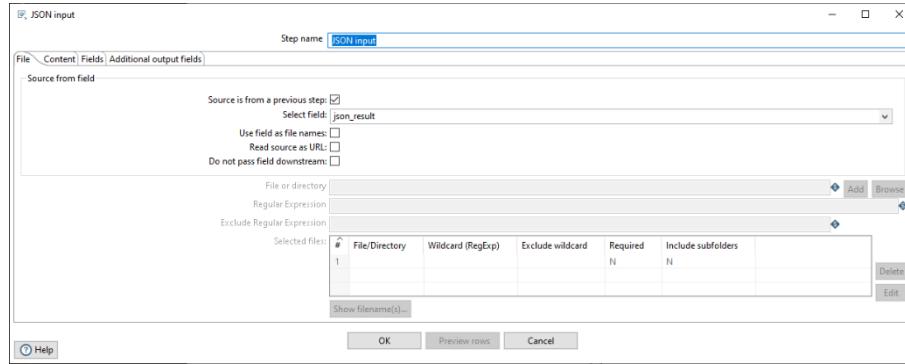
- HTTP client



Gambar 4.71. Konfigurasi Step "HTTP client" dalam Ekstraksi Data Historis Cuaca

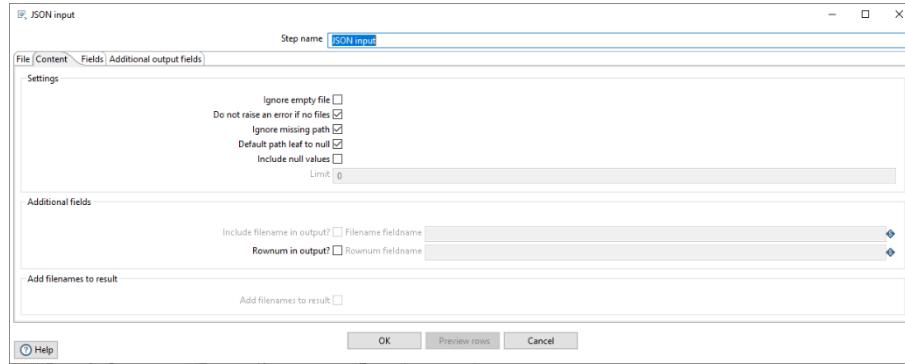
Step “HTTP client” digunakan untuk melakukan *request* ke situs web agar dapat mengekstrak data dari situs web. Dalam hal ini, mengacu pada kolom “historical_weather_api” yang dibuat sebelumnya pada step “Formula”. Hasilnya akan disimpan dalam “json_result”.

- JSON input



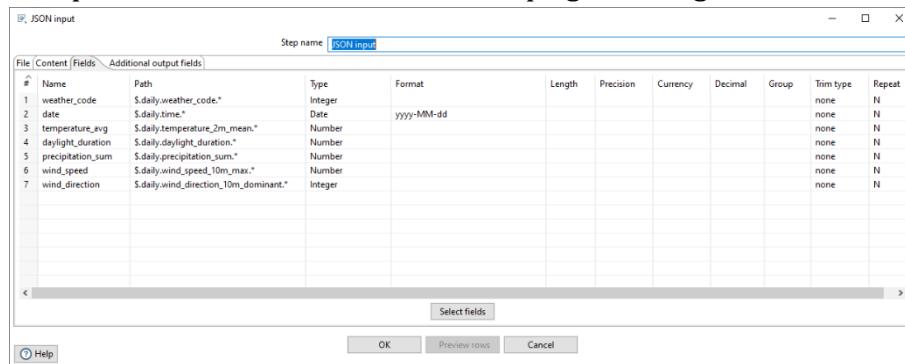
Gambar 4.72. Konfigurasi Step "JSON input" pada tab "File" dalam Ekstraksi Data Historis Cuaca

Step "JSON Input" digunakan untuk mendefinisikan data input berbentuk JSON. Field yang digunakan adalah "json_result" yang didapatkan dari step "HTTP Client" sebelumnya sehingga centang pada bagian "Source is from a previous step".



Gambar 4.73. Konfigurasi Step "JSON input" pada tab "Content" dalam Ekstraksi Data Historis Cuaca

Pada tab "Content", centang pada bagian "Do not raise an error if no files", "Ignore missing path, dan "default path leaf to null" sehingga ketika tidak ada file atau path, tidak perlu memunculkan nilai error tapi, ganti dengan nilai null.



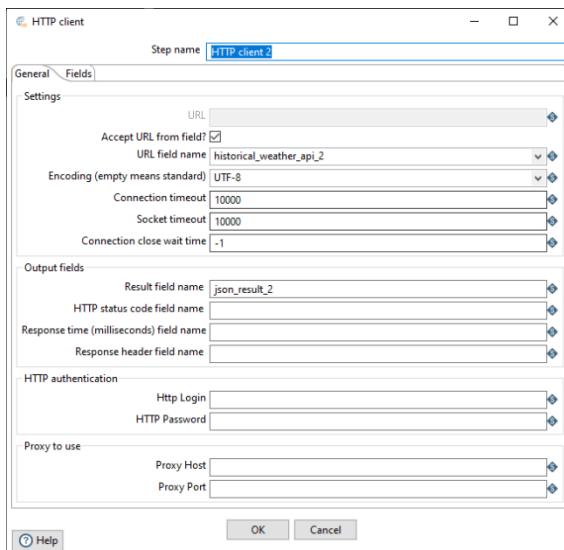
Gambar 4.74. Konfigurasi Step "JSON input" pada tab "Fields" dalam Ekstraksi Data Historis Cuaca

Pada tab "Fields" didefinisikan field yang diambil yaitu "weather_code", "date", "temperature_avg", "daylight_duration", "precipitation_sum", "wind_speed", dan

“wind_direction”. Definisikan juga path, tipe data, dan format untuk tiap field yang diambil. Path yang dituliskan mengacu pada hasil dari ekstraksi data menggunakan api pada step sebelumnya.

6. Jalur 2 (HTTP client 2, JSON input 2)

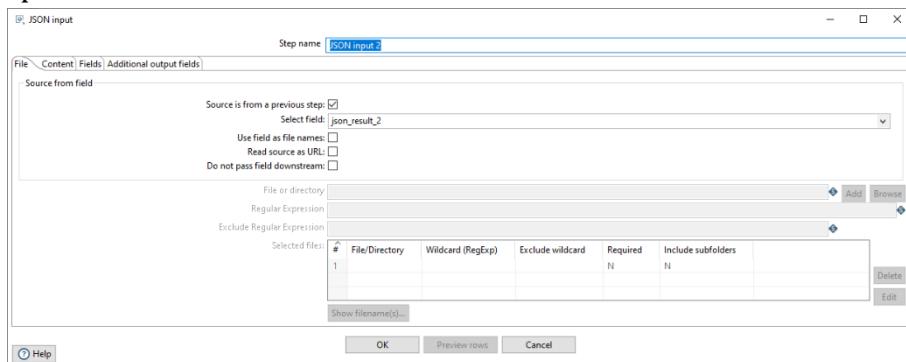
- HTTP client 2



Gambar 4.75. Konfigurasi Step "HTTP client" dalam Ekstraksi Data Historis Cuaca

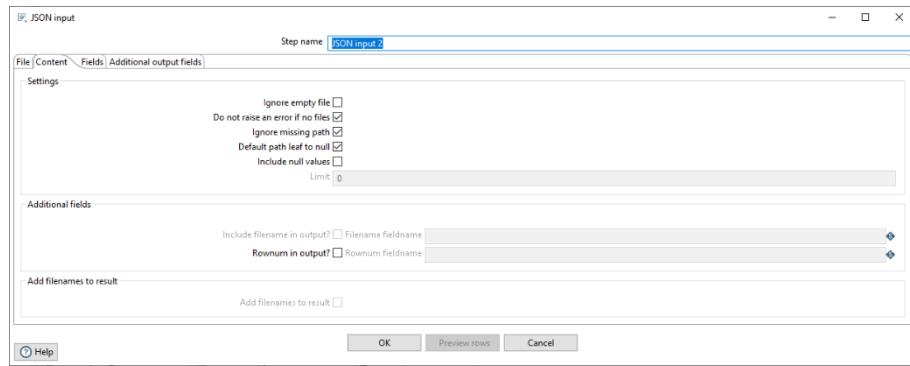
Step “HTTP client 2” digunakan untuk melakukan *request* ke situs web agar dapat mengekstrak data dari situs web. Dalam hal ini, mengacu pada kolom “historical_weather_api_2” yang dibuat sebelumnya pada step “formula”. Hasilnya akan disimpan dalam “json_result_2”.

- JSON Input 2



Gambar 4.76. Konfigurasi Step "JSON input" pada tab "File" dalam Ekstraksi Data Historis Cuaca

Step “JSON Input 2” digunakan untuk mendefinisikan data input berbentuk json. Field yang digunakan adalah “json_result_2” yang didapatkan dari step “HTTP Client 2” sebelumnya sehingga centang pada bagian “Source is from a previous step”.



Gambar 4.77. Konfigurasi Step "JSON input" pada tab "Content" dalam Ekstraksi Data Historis Cuaca

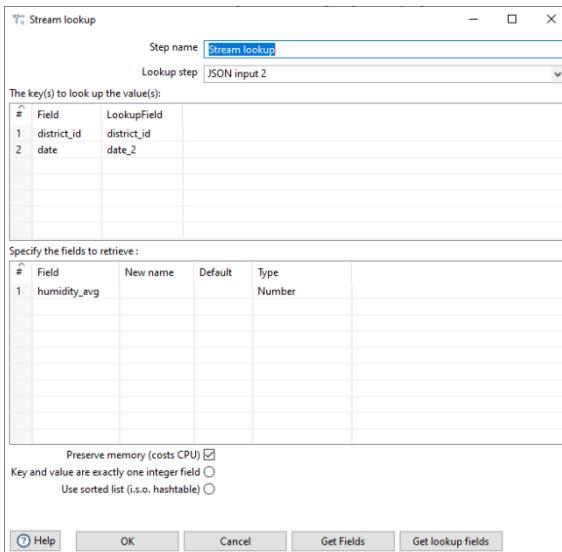
Pada tab "content", centang pada bagian "Do not raise an error if no files", "Ignore missing path, dan "default path leaf to null". Sehingga ketika tidak ada file atau path, tidak perlu memunculkan nilai error tapi, ganti dengan nilai null.

#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	date_2	\$daily.time."	Date	yyyy-MM-dd						none	N
2	humidity_avg	\$daily:relative_humidity_2m_mean."	Number							none	N

Gambar 4.78. Konfigurasi Step "JSON input" pada tab "Fields" dalam Ekstraksi Data Historis Cuaca

Pada tab "Fields" didefinisikan field yang diambil yaitu "date_2" dan "humidity_avg". Definisikan juga path, tipe data, dan format untuk tiap field yang diambil. Path yang dituliskan mengacu pada hasil dari ekstraksi data menggunakan api pada step sebelumnya.

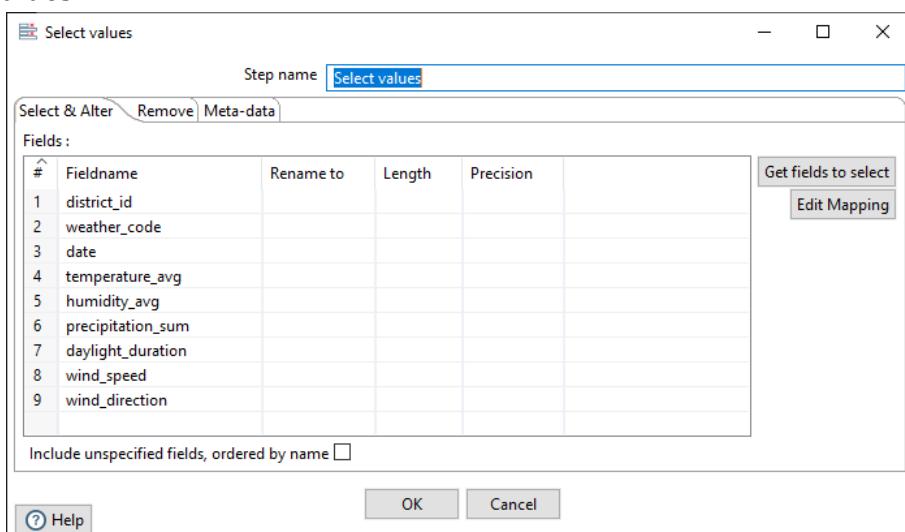
7. Stream Lookup



Gambar 4.79. Konfigurasi Step "Stream Lookup" dalam Ekstraksi Data Historis Cuaca

Stream Lookup digunakan untuk melakukan pencarian atau lookup data dari stream lain (aliran data yang berbeda) dan menambahkannya ke data utama yang sedang diproses. Dalam hal ini, aliran data dari JSON Input 2 akan digunakan sebagai sumber pencarian atau lookup. Key Field adalah kolom-kolom yang digunakan untuk melakukan pencarian (lookup) data yang relevan. Dalam hal ini, field district_id dan date dari aliran utama akan dicocokkan dengan field district_id dan date_2 dari sumber lookup. Variabel "Specify the Field to Retrieve" adalah humidity_avg yang merupakan nilai yang ingin diambil dari stream lookup.

8. Select Values

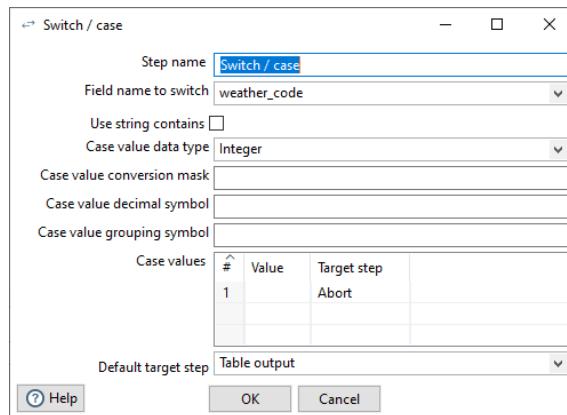


Gambar 4.80. Konfigurasi Step "Select Values" dalam Ekstraksi Data Historis Cuaca

Step "Select Values" digunakan untuk memilih variabel atau field yang diinginkan dari semua field. Terdapat 9 field yang akan diambil yaitu "district_id", "weather_code", "date", "temperature_avg", "humidity_avg", "precipitation_sum",

“daylight_duration”, “wind_speed”, dan “wind_direction”. Field “district_id” diambil dari step “table input”. Field weather_code”, “date”, “temperature_avg”, “precipitation_sum”, “daylight_duration”, “wind_speed”, dan “wind_direction” diambil dari step “JSON_input”. Sedangkan, field “humidity_avg” diambil dari JSON input 2 yang telah dilakukan stream lookup.

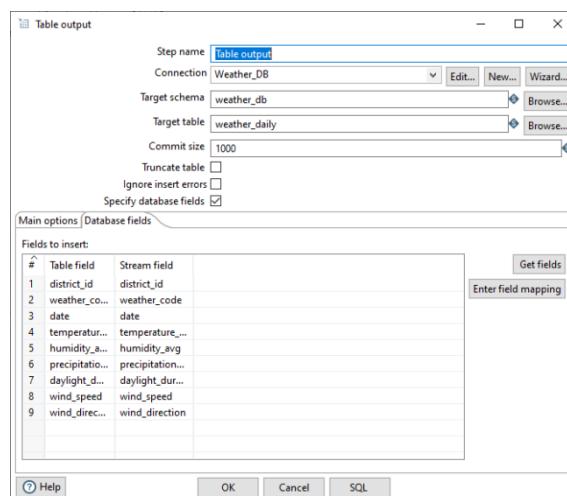
9. Switch/case



Gambar 4.81. Konfigurasi Step "Switch/case" dalam Ekstraksi Data Historis Cuaca

Step Switch/Case digunakan untuk membuat percabangan logika berdasarkan nilai dari suatu field. switch-case dalam mengevaluasi suatu nilai dan melakukan tindakan yang berbeda berdasarkan nilai tersebut. Dalam hal ini, dilakukan pengecekan pada “weather_code” apakah nilai field tersebut integer atau tidak. Jika integer, maka akan masuk ke step “table Output” tetapi, jika bukan integer, maka masuk dalam step “Abort”.

10. Table Output

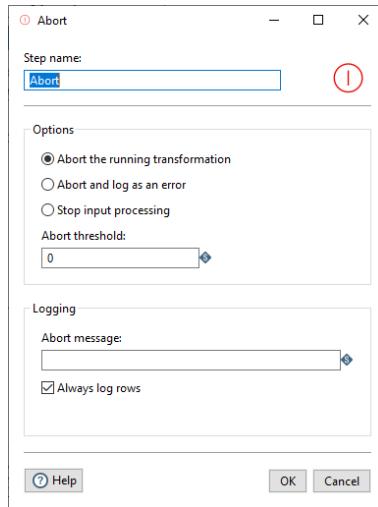


Gambar 4.82. Konfigurasi Step "Table Output" dalam Ekstraksi Data Historis Cuaca

Step “Table Output” digunakan untuk mendefinisikan bentuk tabel sebagai output dari proses dan tempat penyimpanan. Dalam hal ini, hasil tabel tersimpan dalam database weather_db dengan target tabel adalah “weather_daily”. Nama – nama

field yang tersimpan adalah “district_id”, “weather_code”, “date”, “temperature_avg”, “humidity_avg”, “precipitation_sum”, “daylight_duration”, “wind_speed”, dan “wind_direction”.

11. Abort



Gambar 4.83. Konfigurasi Step "Abort" dalam Ekstraksi Data Historis Cuaca

Abort digunakan sebagai penyela atau penghenti aliran data ketika kondisi tertentu dipenuhi. Artinya, jika ada step yang diatur untuk abort, aliran data akan berhenti pada titik tersebut dan tidak akan diteruskan ke langkah berikutnya.

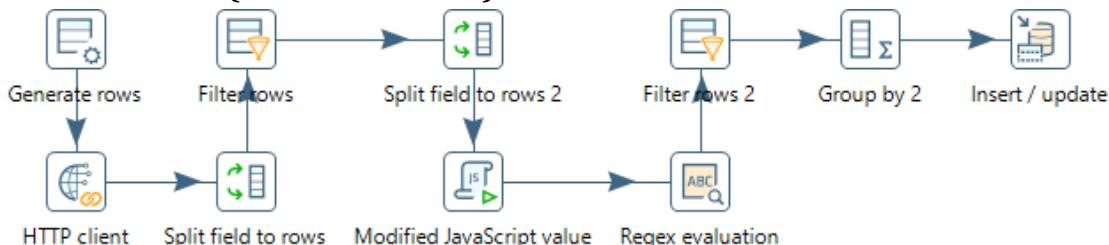
12. Hasil

	weather_id	district_id	weather_code	date	temperature_avg	humidity_avg	precipitation_sum	daylight_duration	wind_speed	wind_direction	created	updated
1	1	357801	45	2010-01-01	26.3	86	2.3	41501_14	9.2		284 2010-12-02 20:00:12	2014-12-02 20:00:12
2	2	357801	45	2010-01-02	25.5	95	12.8	41512_91	11.8		288 2010-12-02 20:00:12	2014-12-02 20:00:12
3	3	357801	45	2010-01-03	25.2	97	18.9	41513_07	9.3		276 2010-12-02 20:00:12	2014-12-02 20:00:12
4	4	357801	55	2010-01-04	26.5	85	2.0	41514_61	6.4		315 2010-12-02 20:00:12	2014-12-02 20:00:12
5	5	357801	45	2010-01-05	24.1	93	12.5	41515_37	14.3		244 2010-12-02 20:00:12	2014-12-02 20:00:12
6	6	357801	45	2010-01-06	25.8	79		41516_95	8.4		249 2010-12-02 20:00:12	2014-12-02 20:00:12
7	7	357801	45	2010-01-07	25.8	75	20.2	41517_79	10.6		293 2010-12-02 20:00:12	2014-12-02 20:00:12
8	8	357801	45	2010-01-08	25.1	75	18.2	41518_09	11.4		382 2010-12-02 20:00:12	2014-12-02 20:00:12
9	9	357801	55	2010-01-09	24.9	74	6.2	41519_30	9.3		269 2010-12-02 20:00:12	2014-12-02 20:00:12
10	10	357801	45	2010-01-10	24.4	73	18.8	41520_47	7.3		373 2010-12-02 20:00:12	2014-12-02 20:00:12
11	11	357801	45	2010-01-11	25.2	69	16.9	41521_81	18.3		299 2010-12-02 20:00:12	2014-12-02 20:00:12
12	12	357801	45	2010-01-12	25.3	70	24.7	41522_29	13.2		299 2010-12-02 20:00:12	2014-12-02 20:00:12
13	13	357801	51	2010-01-13	26.1	64	0.1	41523_12	13.2		284 2010-12-02 20:00:12	2014-12-02 20:00:12
14	14	357801	45	2010-01-14	24.0	77		41524_56	13.2		289 2010-12-02 20:00:12	2014-12-02 20:00:12
15	15	357801	53	2010-01-15	24.5	97		41525_15	6.9		391 2010-12-02 20:00:12	2014-12-02 20:00:12
16	16	357801	45	2010-01-16	24.4	91	20.0	41526_29	7.7		250 2010-12-02 20:00:12	2014-12-02 20:00:12
17	17	357801	45	2010-01-17	25.5	93	11.3	41527_45	13.3		275 2010-12-02 20:00:12	2014-12-02 20:00:12
18	18	357801	45	2010-01-18	25.4	90	9.1	41528_32	12.3		330 2010-12-02 20:00:12	2014-12-02 20:00:12
19	19	357801	45	2010-01-19	25.1	88	22.1	41529_84	9.1		377 2010-12-02 20:00:12	2014-12-02 20:00:12
20	20	357801	45	2010-01-20	24.8	88	13.1	41530_86	18.2		278 2010-12-02 20:00:12	2014-12-02 20:00:12
21	21	357801	45	2010-01-21	25.1	87	7.8	41531_53	20.4		283 2010-12-02 20:00:12	2014-12-02 20:00:12
22	22	357801	45	2010-01-22	25.4	89	9.9	41532_29	24.3		284 2010-12-02 20:00:12	2014-12-02 20:00:12
23	23	357801	45	2010-01-23	25.7	91	9.4	41533_75	24.4		291 2010-12-02 20:00:12	2014-12-02 20:00:12
24	24	357801	45	2010-01-24	25.3	88	10.2	41534_79	19.4		290 2010-12-02 20:00:12	2014-12-02 20:00:12
25	25	357801	63	2010-01-25	25.5	89	25.1	41535_35	19.4		294 2010-12-02 20:00:12	2014-12-02 20:00:12
26	26	357801	45	2010-01-26	24.9	88	15.5	41536_40	19.6		283 2010-12-02 20:00:12	2014-12-02 20:00:12
27	27	357801	45	2010-01-27	25.1	91	13.3	41537_23	19.3		285 2010-12-02 20:00:12	2014-12-02 20:00:12
28	28	357801	45	2010-01-28	24.8	90	14.8	41538_44	19.4		297 2010-12-02 20:00:12	2014-12-02 20:00:12
29	29	357801	45	2010-01-29	25.8	89	18.0	41539_72	11.4		270 2010-12-02 20:00:12	2014-12-02 20:00:12
30	30	357801	45	2010-01-30	25.3	91	20.1	41540_49	16.5		282 2010-12-02 20:00:12	2014-12-02 20:00:12
31	31	357801	45	2010-01-31	24.7	90	12.6	41541_99	14.4		281 2010-12-02 20:00:12	2014-12-02 20:00:12
32	32	357801	45	2010-02-01	24.8	90	15.7	41542_22	13.7		276 2010-12-02 20:00:12	2014-12-02 20:00:12
33	33	357801	45	2010-02-02	24.8	91	10.5	41543_92	14.2		282 2010-12-02 20:00:12	2014-12-02 20:00:12
34	34	357801	45	2010-02-03	25.0	91	22.5	41544_93	14.9		287 2010-12-02 20:00:12	2014-12-02 20:00:12
35	35	357801	53	2010-02-04	24.9	91	5.2	41545_63	29.5		274 2010-12-02 20:00:12	2014-12-02 20:00:12
36	36	357801	2	2010-02-05	25.9	90	0.0	41546_05	16.2		274 2010-12-02 20:00:12	2014-12-02 20:00:12
37	37	357801	45	2010-02-06	25.7	87	10.1	41547_56	13.4		285 2010-12-02 20:00:12	2014-12-02 20:00:12
38	38	357801	63	2010-02-07	25.2	89	21.8	41548_50	17.0		288 2010-12-02 20:00:12	2014-12-02 20:00:12
39	39	357801	45	2010-02-08	24.4	89	20.2	41549_42	14.9		286 2010-12-02 20:00:12	2014-12-02 20:00:12
40	40	357801	45	2010-02-09	24.5	87	-7.7	41550_42	26.9		287 2010-12-02 20:00:12	2014-12-02 20:00:12
41	41	357801	45	2010-02-10	25.4	88	-7.7	41550_88	19.1		292 2010-12-02 20:00:12	2014-12-02 20:00:12

Gambar 4.84. Potongan Hasil Ekstraksi Data Historis Cuaca dalam Basis Data OLTP

Potongan data dari tabel “weather_daily” tampak pada gambar diatas dimana tabel weather_daily memiliki 12 kolom dengan 282.131 baris. Kolom “created” dan “updated” dibuat untuk memantau waktu kapan data dibuat atau diperbarui.

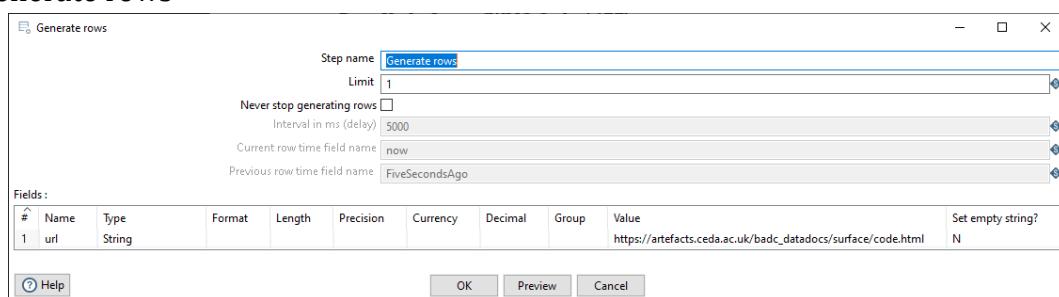
c. Data Kode Cuaca (WMO Code 4677)



Gambar 4.85. Rangkaian Step PDI untuk Ekstraksi Data Kode Cuaca

Gambar di atas menunjukkan proses ekstraksi data deskripsi kode cuaca (WMO Code 4677) dan menyimpannya dalam basis data OLTP. Berikut adalah penjelasan dari setiap step yang digunakan:

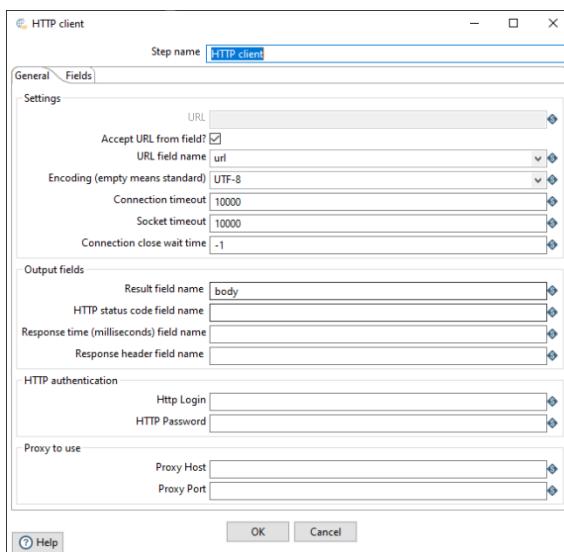
1. Generate rows



Gambar 4.86. Konfigurasi Step "Generate rows" dalam Ekstraksi Data Kode Cuaca

Step "Generate rows" digunakan untuk membuat baris data. Dalam studi kasus ini, akan dibuat data dengan satu kolom bernama "url" dengan tipe string yang memiliki nilai "https://artefacts.ceda.ac.uk/badc_datadocs/surface/code.html" yang merupakan tautan situs web yang menyediakan kode dan deskripsi dari WMO Code 4677.

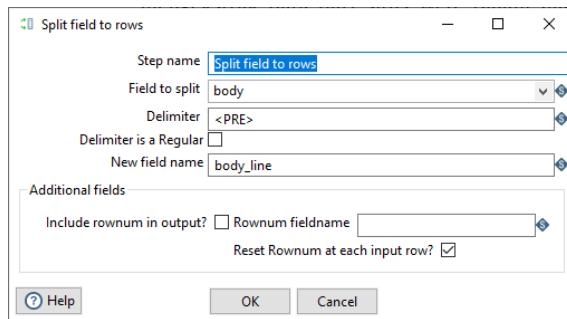
2. HTTP Client



Gambar 4.87. Konfigurasi Step "HTTP client" dalam Ekstraksi Data Kode Cuaca

Step “HTTP client” digunakan untuk melakukan *request* ke situs web agar dapat mengekstrak data dari situs web. Dalam hal ini, mengacu pada ekstraksi dari “https://artefacts.ceda.ac.uk/badc_datadocs/surface/code.html” atau kolom “url” yang telah dibuat pada step “Generate rows”. Hasil ekstraksi data akan tersimpan dalam field bernama “body”.

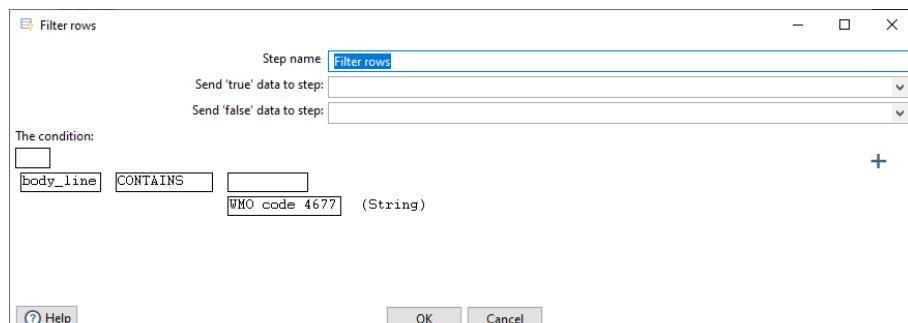
3. Split field to rows



Gambar 4.88. Konfigurasi Step "Split field to rows" dalam Ekstraksi Data Kode Cuaca

Step “Split field to rows” berfungsi untuk membagi sebuah field (dalam hal ini bernama “body”) menjadi beberapa baris, berdasarkan delimiter yang telah ditentukan yaitu “<PRE>” dan menyimpannya dalam field baru bernama “body_line”.

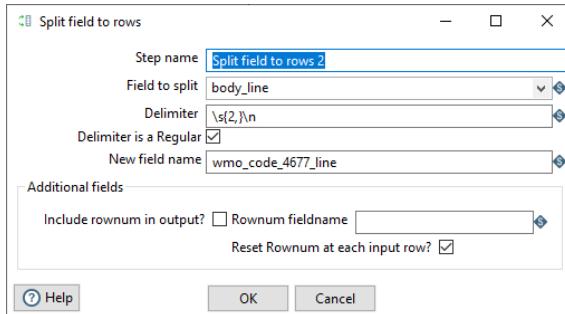
4. Filter rows



Gambar 4.89. Konfigurasi Step "Filter rows" dalam Ekstraksi Data Kode Cuaca

Step “Filter rows” digunakan untuk menyaring atau memfilter baris data berdasarkan kondisi atau aturan tertentu. Dalam studi kasus ini, step ini akan memfilter baris yang berisi teks “WMO code 4677” dalam “body_line”. Baris yang mengandung string tersebut yang akan lolos dari filter dan diteruskan ke langkah pemrosesan berikutnya, sementara baris lain yang tidak mengandung “WMO code 4677” akan dibuang. Kode 4677 sendiri menggambarkan deskripsi cuaca saat ini.

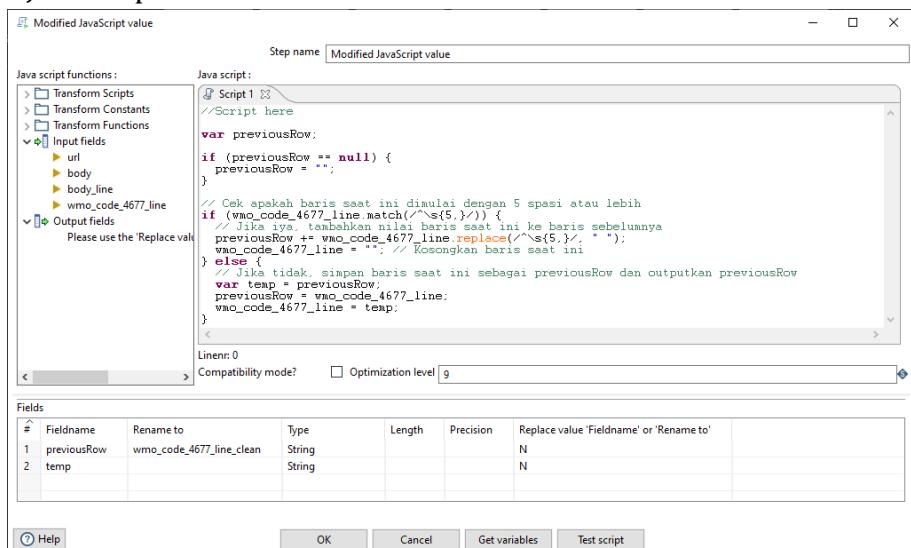
5. Split field to rows 2



Gambar 4.90. Konfigurasi Step "Split field to rows 2" dalam Ekstraksi Data Kode Cuaca

Step "Split field to rows 2" berfungsi untuk membagi sebuah field (dalam hal ini bernama "body_line;) menjadi beberapa baris, berdasarkan delimiter yang telah ditentukan yaitu "\s{2}\n" dan menyimpannya dalam field baru bernama "wmo_code_4677_line. Dalam regular expression, "\s{2}\n" berarti: "\s" adalah simbol untuk whitespace atau karakter spasi, "{2," menunjukkan bahwa karakter whitespace harus muncul minimal 2 kali berturut-turut, dan "\n" adalah simbol untuk karakter newline atau baris baru.

6. Modified JavaScript value



Gambar 4.91. Konfigurasi Step "Modified JavaScript value" dalam Ekstraksi Data Kode Cuaca

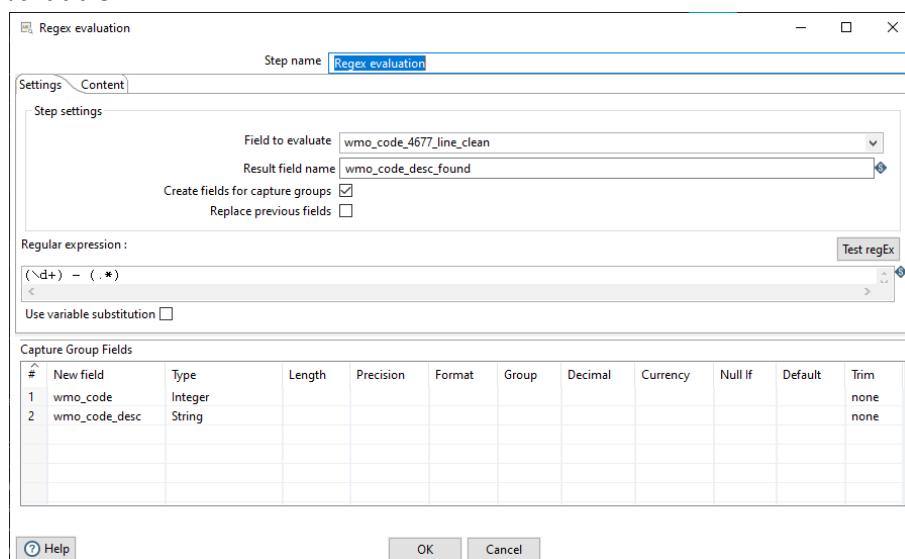
Step "Modified JavaScript Value" digunakan untuk memodifikasi nilai dalam field menggunakan kode JavaScript. Pada kasus ini, skrip di atas melakukan penggabungan baris atau penyusunan ulang nilai dari wmo_code_4677_line dengan tujuan agar baris-baris yang berawal dengan banyak spasi digabungkan dengan baris sebelumnya. Variabel "previousRow" digunakan untuk menyimpan baris sebelumnya. Jika previousRow kosong, maka diset sebagai string kosong (""). Kemudian, dilakukan pengecekan untuk melihat apakah baris saat ini (wmo_code_4677_line) dimulai dengan 5 atau lebih spasi. Jika kondisi ini benar, baris tersebut dianggap sebagai bagian dari baris sebelumnya. Bagian awal dari 5

spasi atau lebih diganti dengan satu spasi (menggunakan replace(/^\s{5,}/, " ")) dan kemudian digabungkan dengan previousRow.

Jika baris tidak dimulai dengan 5 spasi atau lebih, artinya baris tersebut adalah baris baru yang perlu diproses. Baris sebelumnya (previousRow) disimpan sementara ke variabel temp. Baris saat ini (wmo_code_4677_line) diset menjadi previousRow untuk diolah pada langkah berikutnya.

Akan dibangun data dengan dua kolom yaitu variabel “previousRow” menjadi kolom “wmo_code_4677_line_clean” untuk menyimpan baris sebelumnya untuk menggabungkan dengan baris saat ini jika dimulai dengan spasi. Serta, “temp” sebagai tempat sementara untuk menyimpan baris sebelumnya sebelum melakukan perpindahan nilai.

7. Regex evaluation

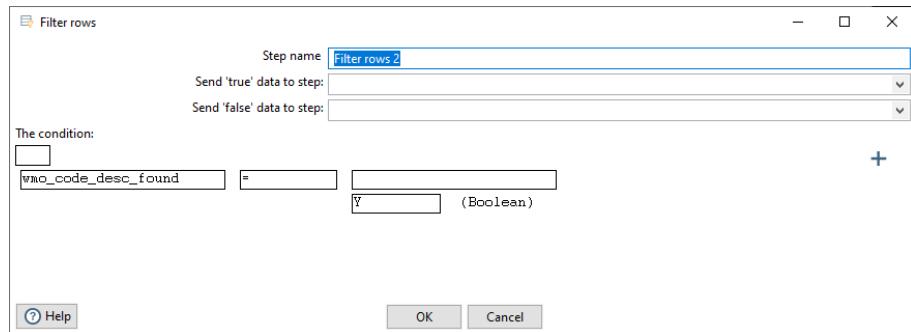


Gambar 4.92. Konfigurasi Step "Regex evaluation" dalam Ekstraksi Data Kode Cuaca

Step Regex evaluation digunakan untuk mengevaluasi apakah sebuah field cocok dengan pola regex dan mengekstrak bagian-bagian tertentu dari string menggunakan capture groups. Dalam kasus ini, langkah tersebut dilakukan untuk field wmo_code_4677_line_clean dan hasilnya disimpan di field wmo_code_desc_found. Create Field for Capture Groups akan dicentang agar dibuat field terpisah untuk setiap capture group yang ditemukan dalam regex.

Digunakan pola regex “(\d+) - (.*)” artinya “\d+” menangkap satu atau lebih digit angka (0-9), “-” adalah pola sambungnya, dan “*” menangkap nol atau lebih karakter apa pun hingga akhir baris. Hasil dari grup pertama yaitu “\d+” disimpan dalam wmo_code dan hasil dari grup kedua yaitu “*” disimpan dalam wmo_code_desc.

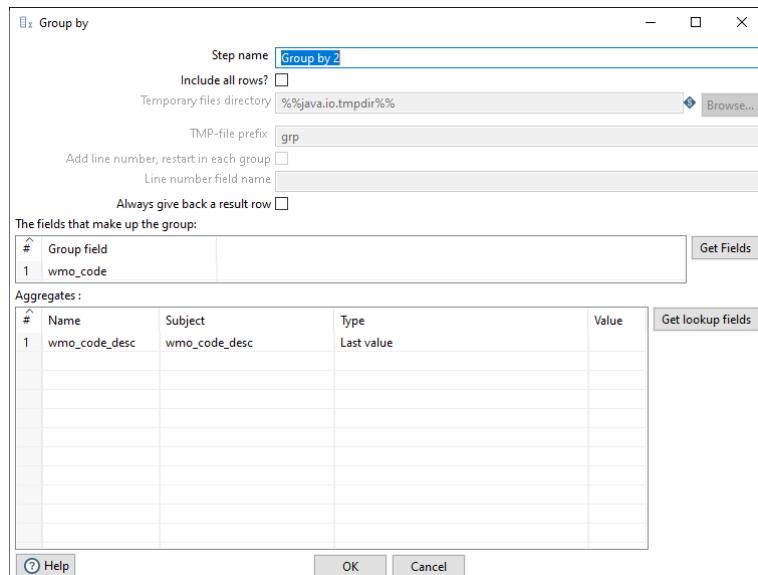
8. Filter rows 2



Gambar 4.93. Konfigurasi Step "Filter rows 2" dalam Ekstraksi Data Kode Cuaca

Pada step "Filter Rows 2" dengan kondisi "wmo_code_desc_found = Y", ini berarti akan menyaring atau memfilter hanya baris-baris yang telah berhasil menemukan atau mencocokkan deskripsi kode WMO (cuaca atau data lain) yang valid dalam kolom wmo_code_desc_found. Field wmo_code_desc_found adalah hasil dari step "Regex evaluation" sebelumnya. Kondisi diisi dengan nilai Y (atau Yes) dimana jika data wmo_code_desc_found sesuai dengan pola regex yang diberikan `((\d+) - (.*))` dan menemukan wmo_code serta wmo_code_desc dalam baris tersebut, maka data dipertahankan.

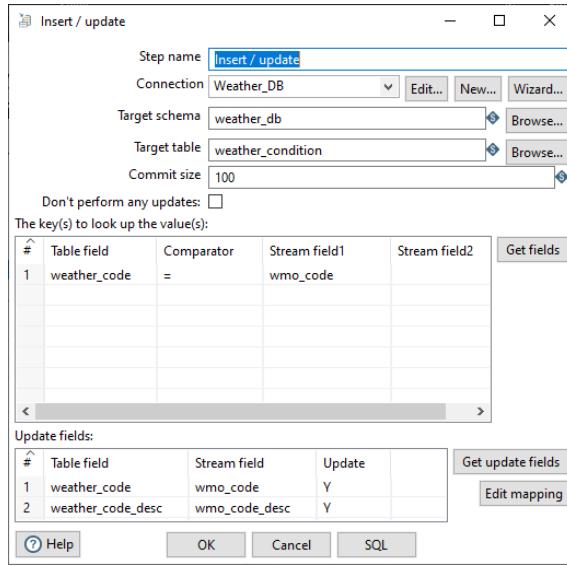
9. Group by 2



Gambar 4.94. Konfigurasi Step "Group by 2" dalam Ekstraksi Data Kode Cuaca

Step "Group By" dapat digunakan untuk mengelompokkan data berdasarkan suatu field tertentu, dan kemudian melakukan agregasi pada field lainnya. Dalam kasus ini, step Group By digunakan untuk mengelompokkan data berdasarkan wmo_code dan mendapatkan nilai terakhir (last value) dari wmo_code_desc. Jadi, semua baris dengan kode WMO yang sama akan dikelompokkan ke dalam satu grup. Kemudian, dilakukan agregasi untuk wmo_code_desc dengan mengambil nilai terakhir dari wmo_code_desc dalam setiap grup.

10. Insert/update



Gambar 4.95. Konfigurasi Step "Insert/update" dalam Ekstraksi Data Kode Cuaca

Step “Insert/Update” digunakan untuk menghubungkan data yang ada dengan basis data dan memperbarui atau menyisipkan data baru ke dalam tabel yang sudah ada, berdasarkan kunci yang ditentukan. Dalam hal ini, basis data yang digunakan adalah weather_db dengan target table weather_condition. Kunci utama untuk mengidentifikasi apakah baris yang ada sudah ada sebelumnya di dalam tabel adalah weather_code (dari tabel weather_condition) dan wmo_code (Kolom yang didefinisikan). Artinya, jika ada kode yang sama, maka data yang baru akan digunakan untuk update data yang lama. Jika wmo_code tidak ada, maka data baru akan dimasukkan (insert).

11. Hasil

#	Table field	Stream field	Update
1	weather_code	wmo_code	Y
2	weather_code_desc	wmo_code_desc	Y

Gambar 4.96. Potongan Hasil Ekstraksi Data Kode Cuaca dalam Basis Data OLTP

Potongan data dari tabel weather_condition tampak pada gambar di atas dimana tabel weather_condition memiliki 4 kolom dengan 100 baris. Kolom “created” dan “updated” dibuat untuk memantau waktu kapan data dibuat atau diperbarui.

d. Data Kota dan Provinsi

Proses ekstraksi data pada tabel “city” dan “province” memiliki langkah-langkah yang serupa seperti proses ekstraksi data kecamatan. Oleh karena itu, diperoleh data kota dan provinsi dari tabel “city” dan “province” seperti berikut ini:

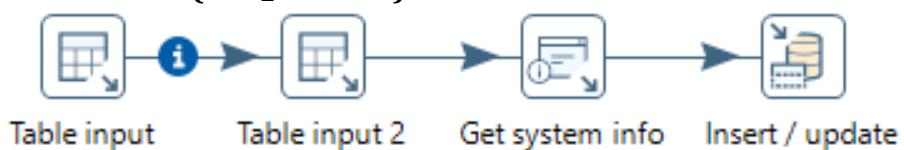
The screenshot shows two separate database query results side-by-side. Both queries are run against a schema named 'main'. The top query, titled 'city', retrieves data for Surabaya with ID 3578. The bottom query, titled 'province', retrieves data for Java Timur with ID 35. Both queries include columns for ID, name, latitude, longitude, created timestamp, and updated timestamp. The data is presented in a standard tabular format with rows and columns.

Table	ID	Name	Latitude	Longitude	Created	Updated
city	3578	Surabaya	-7.2750000	112.71866700	2024-12-02 16:01:02	2024-12-02 16:05:01
province	35	Java Timur	-7.7394000	112.50990000	2024-12-02 15:55:02	2024-12-02 15:56:49

Gambar 4.97. Hasil Ekstraksi Data Kota dan Provinsi dalam Basis Data OLTP

4.4. Transformasi Data

a. Tabel Dimensi Lokasi (dim_location)

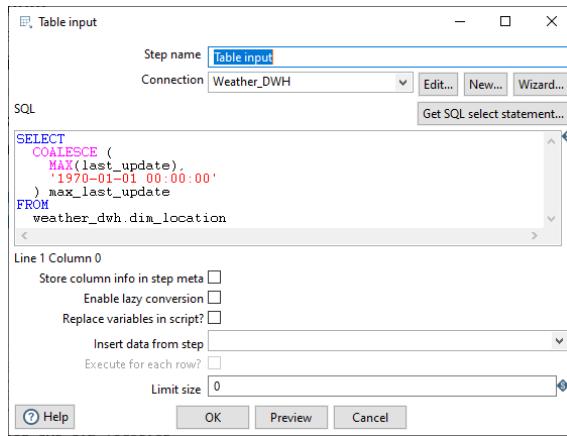


Gambar 4.98. Rangkaian Step PDI untuk Mentransformasikan Data Tabel “dim_location”

Proses pada gambar di atas menggambarkan alur ETL untuk melakukan impor data ke dalam tabel dim_location dalam basis data OLAP. Berikut adalah penjelasan dari setiap step yang digunakan:

1. Table input

Kegunaan dari step ini adalah untuk mengambil timestamp dari pembaruan terakhir (last update) yang tercatat di tabel dim_location pada Data Warehouse (DWH). Fungsi ini berguna untuk memastikan bahwa hanya data terbaru (yang diperbarui setelah last update) yang akan diambil dari database weather_db. Table input memiliki pengaturan sebagai berikut.

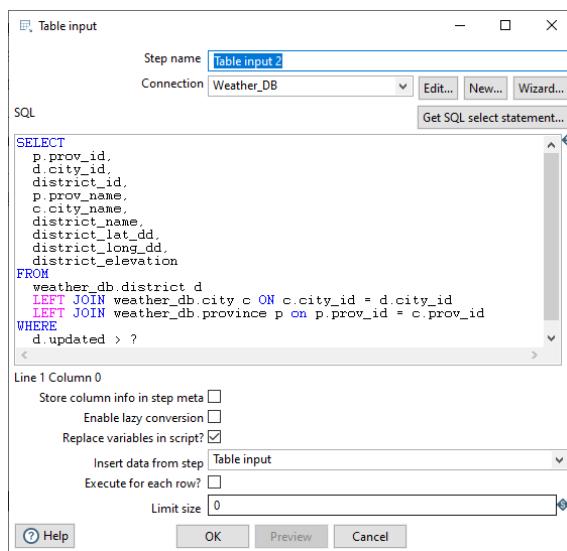


Gambar 4.99. Konfigurasi Step "Table input" dalam Transformasi Data Tabel "dim_location"

Langkah dalam SQL mengambil data max_last_update dari tabel dim_location di basis data OLAP. Kueri ini memastikan jika tidak ada data yang tersedia di dim_location, maka akan menggunakan tanggal default "1970-01-01 00:00:00" namun, jika ada data yang tersedia, digunakan tnggal yang tertera pada "last update".

2. Table input 2

Kegunaan dari step ini adalah untuk mengambil data wilayah terbaru dari database sumber (weather_db) berdasarkan timestamp last update yang diperoleh dari Table Input 1. Dengan kata lain, Table Input 2 hanya akan mengambil data yang di-update setelah waktu yang ditentukan oleh hasil dari Table Input 1. Data yang diambil meliputi informasi provinsi, kota, distrik, serta detail lokasi seperti latitudo, longitudino, dan elevasi.

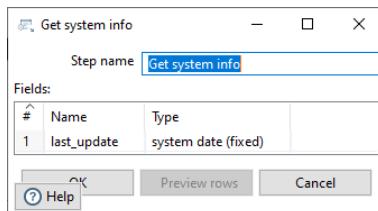


Gambar 4.100. Konfigurasi Step "Table input 2" dalam Transformasi Data Tabel "dim_location"

Pada langkah ini, kueri mengambil data wilayah terbaru dari basis data OLTP (weather_db) berdasarkan last_update yang telah ditentukan sebelumnya dari Table Input 1. Kueri ini menggabungkan beberapa tabel: district, city, dan province

untuk mendapatkan data wilayah secara lengkap. Kolom yang diambil meliputi ID dan nama provinsi, kota, serta distrik, termasuk data latitudo, longitudino, dan elevasi dari distrik tersebut. Filter “d.updated > ?” menggunakan parameter yang diisi dengan max_last_update dari Table Input 1, memastikan bahwa hanya data yang diperbarui setelah waktu tersebut yang akan diambil.

3. Get system info

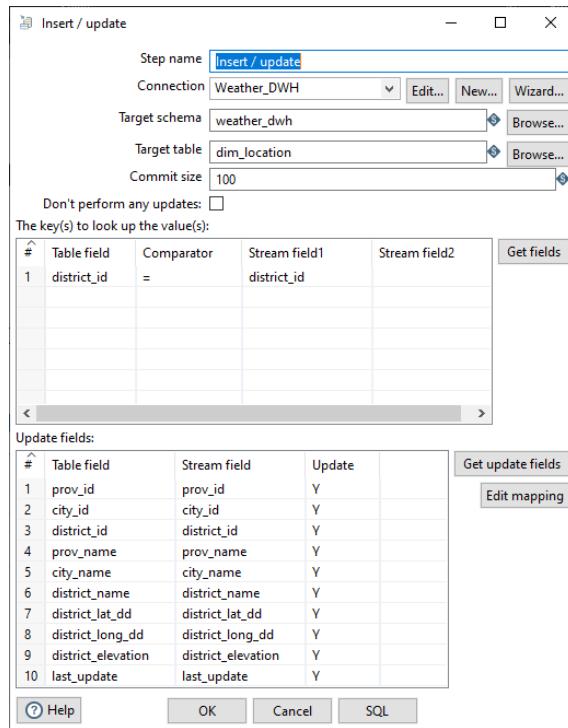


Gambar 4.101. Konfigurasi Step "Get system info" dalam Transformasi Data Tabel "dim_location"

Pada step Get system info digunakan untuk mengambil informasi waktu sistem saat proses ETL berlangsung. Dalam konteks ini, atribut yang dihasilkan adalah "Last Update" dengan tipe System Date (Fixed) yang bertujuan untuk mencatat waktu terbaru ketika proses memasukkan atau memperbarui data ke dalam tabel OLAP dim_location. Pencatatan ini penting karena memungkinkan sistem untuk mengetahui kapan data terakhir diperbarui, memastikan bahwa hanya data yang lebih baru yang akan diproses di masa mendatang.

4. Insert/update

Step ini digunakan untuk memasukkan atau memperbarui data ke dalam tabel target, yaitu dim_location dalam skema weather_dwh. Kunci utama yang digunakan untuk mencocokkan data adalah district_id. Jika data dengan district_id yang sesuai ditemukan, maka proses *update* dilakukan. Jika tidak ditemukan, data baru akan dimasukkan.



Gambar 4.102. Konfigurasi Step "Insert/update" dalam Transformasi Data Tabel "dim_location"

Bagian Update field menunjukkan kolom-kolom yang akan diperbarui dalam tabel target berdasarkan aliran data yang masuk, seperti prov_id, city_id, district_name, district_lat_dd, dan lainnya. Field last_update juga diperbarui dengan waktu terbaru yang diambil dari tahap Get System Info sebelumnya, memastikan setiap data yang di-update memiliki jejak waktu yang tepat.

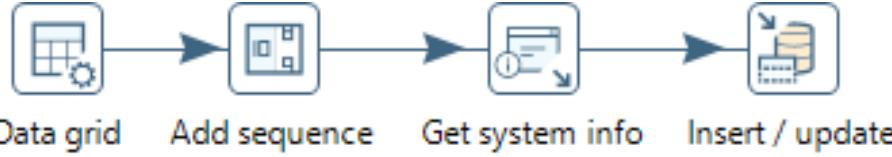
5. Hasil

Potongan data dari dim_location tampak pada gambar dibawah dimana tabel dim_location memiliki 11 kolom dengan 31 baris yang menunjukkan total kecamatan di kota Surabaya.

#	pk	prec_id	city_id	district_id	prev_name	city_name	district_name	district_lv1_id	district_long_id	district_elevation	last_update
1	2	35	3579	357902	Jawa Tengah	Surabaya	Karang Pilang	-7.34311111	122.48413333	10.0	2024-12-12 20:10:03
2	2	35	3579	357902	Jawa Tengah	Surabaya	Wonocojo	-7.32200000	122.73988899	8.9	2024-12-12 20:10:03
3	3	35	3579	357903	Jawa Tengah	Surabaya	Rungkut	-7.32000000	122.74094400	8.0	2024-12-12 20:10:03
4	4	35	3579	357904	Jawa Tengah	Surabaya	Wonokromo	-7.29800000	122.73778800	8.0	2024-12-12 20:10:03
5	5	35	3579	357905	Jawa Tengah	Surabaya	Tegalalang	-7.27050000	122.73306400	6.6	2024-12-12 20:10:03
6	6	35	3579	357906	Jawa Tengah	Surabaya	Ujungpandji	-7.27050000	122.73015700	14.2	2024-12-12 20:10:03
7	7	35	3579	357907	Jawa Tengah	Surabaya	Genteng	-7.21600000	122.74620000	7.4	2024-12-12 20:10:03
8	8	35	3579	357908	Jawa Tengah	Surabaya	Gubeng	-7.27972200	122.75621100	8.4	2024-12-12 20:10:03
9	9	35	3579	357909	Jawa Tengah	Surabaya	Sukolilo	-7.29200000	122.80333300	5.0	2024-12-12 20:10:03
10	10	35	3579	357910	Jawa Tengah	Surabaya	Tambaklari	-7.25050000	122.74865600	5.0	2024-12-12 20:10:03
11	11	35	3579	357911	Jawa Tengah	Surabaya	Siwekeran	-7.23040000	122.79353300	6.8	2024-12-12 20:10:03
12	12	35	3579	357912	Jawa Tengah	Surabaya	Pelungu	-7.21500000	122.73177700	8.3	2024-12-12 20:10:03
13	13	35	3579	357913	Jawa Tengah	Surabaya	Pelutan	-7.24000000	122.72777000	5.4	2024-12-12 20:10:03
14	14	35	3579	357914	Jawa Tengah	Surabaya	Tandes	-7.20300000	122.67138900	9.0	2024-12-12 20:10:03
15	15	35	3579	357915	Jawa Tengah	Surabaya	Krengbangan	-7.22500000	122.72164700	2.0	2024-12-12 20:10:03
16	16	35	3579	357916	Jawa Tengah	Surabaya	Selempr	-7.21300000	122.74853300	2.0	2024-12-12 20:10:03
17	17	35	3579	357917	Jawa Tengah	Surabaya	Kregeran	-7.21300000	122.74794400	5.0	2024-12-12 20:10:03
18	18	35	3579	357918	Jawa Tengah	Surabaya	Laharharjo	-7.20900000	122.64650000	8.9	2024-12-12 20:10:03
19	19	35	3579	357919	Jawa Tengah	Surabaya	Kesso	-7.22300000	122.44722200	1.4	2024-12-12 20:10:03
20	20	35	3579	357920	Jawa Tengah	Surabaya	Klung	-7.31300000	122.68833300	9.0	2024-12-12 20:10:03
21	21	35	3579	357921	Jawa Tengah	Surabaya	Dukuh Pakis	-7.29600000	122.69889900	22.0	2024-12-12 20:10:03
22	22	35	3579	357922	Jawa Tengah	Surabaya	Gayungan	-7.32000000	122.72638800	7.0	2024-12-12 20:10:03
23	23	35	3579	357923	Jawa Tengah	Surabaya	Jamrang	-7.21300000	122.72950000	11.8	2024-12-12 20:10:03
24	24	35	3579	357924	Jawa Tengah	Surabaya	Tenggulungan Mojop	-7.21000000	122.79250000	8.5	2024-12-12 20:10:03
25	25	35	3579	357925	Jawa Tengah	Surabaya	Guning Anyer	-7.33000000	122.79300000	2.4	2024-12-12 20:10:03
26	26	35	3579	357926	Jawa Tengah	Surabaya	Mulorejo	-7.24450000	122.79956700	5.0	2024-12-12 20:10:03
27	27	35	3579	357927	Jawa Tengah	Surabaya	Sukomenggal	-7.26800000	122.69856000	8.0	2024-12-12 20:10:03
28	28	35	3579	357928	Jawa Tengah	Surabaya	Asen Rasa	-7.23700000	122.68700000	4.0	2024-12-12 20:10:03
29	29	35	3579	357929	Jawa Tengah	Surabaya	Bulak	-7.23000000	122.74017600	2.0	2024-12-12 20:10:03
30	30	35	3579	357930	Jawa Tengah	Surabaya	Paket	-7.20972200	122.66010000	5.4	2024-12-12 20:10:03
31	31	35	3579	357931	Jawa Tengah	Surabaya	Semisareng	-7.27600000	122.65956400	20.4	2024-12-12 20:10:03

Gambar 4.103. Hasil Transformasi Data Tabel “dim_location” dalam Basis Data OLAP

b. Tabel Dimensi Intensitas Hujan (dim_rainfall)

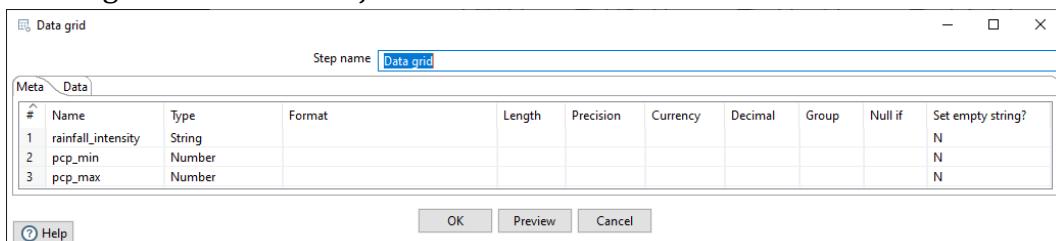


Gambar 4.104. Rangkaian Step PDI untuk Mentransformasikan Data Tabel “dim_rainfall”

Proses pada gambar di atas menggambarkan alur ETL untuk melakukan impor data ke dalam tabel dim_rainfall dalam basis data OLAP. Berikut adalah penjelasan dari setiap step yang digunakan:

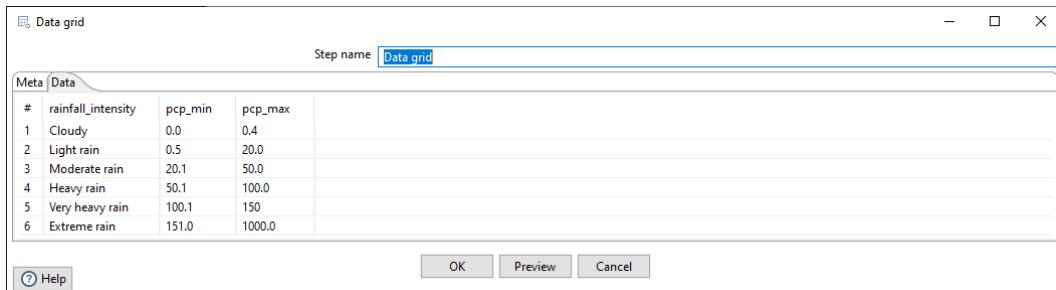
1. Data grid

Step Data grid berfungsi untuk membuat data secara manual dengan memasukkan data mentah ke dalam tabel. Step Data grid akan menjadi titik awal di mana data awal mengenai intensitas hujan dimasukkan ke dalam sistem.



Gambar 4.105. Konfigurasi Step “Data grid” pada tab “Meta” dalam Transformasi Data Tabel “dim_rainfall”

Pada tab Meta digunakan untuk mendeskripsikan metadata setiap kolom. Nama kolom yang didefinisikan adalah rainfall_intensity (intensitas hujan dalam string), pcp_min (curah hujan minimum dalam bentuk angka), dan pcp_max (curah hujan maksimum dalam bentuk angka).

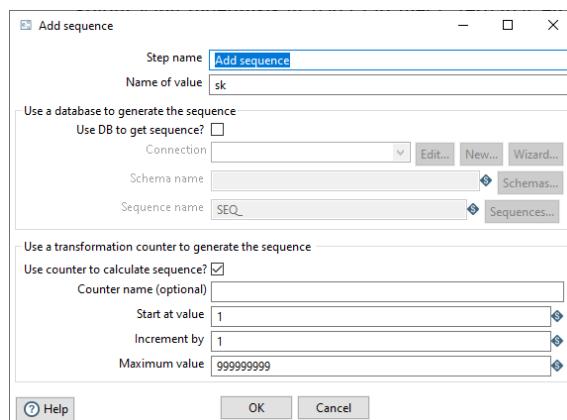


Gambar 4. 106. Konfigurasi Step "Data grid" pada tab "Data" dalam Transformasi Data Tabel "dim_rainfall"

Sedangkan, pada tab data digunakan untuk memasukkan data yang sesuai dengan kolom yang didefinisikan pada tab meta. Terdapat enam data yang dimasukkan yaitu *Cloudy*, *Light rain*, *Moderate rain*, *Heavy rain*, *very heavy rain*, dan *extreme rain* dengan pcp minimal dan maksimum tiap kategori.

2. Add sequence

Step "Add Sequence" berfungsi untuk menambahkan nomor urut ke setiap baris data. Nomor urut dalam dimensi ini akan berfungsi sebagai "sk" yaitu id unik untuk dapat mengidentifikasi setiap baris data.



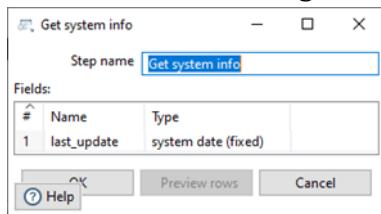
Gambar 4.107. Konfigurasi Step "Add sequence" dalam Transformasi Data Tabel "dim_rainfall"

Gambar di atas merupakan pengaturan untuk step "Add sequence" dengan menunjukkan nomor urut yang ditambahkan akan memiliki kolom bernama sk dengan rentang dari 1 hingga 999.999.999 dengan setiap perpindahan baris, urutan akan ditambahkan sejumlah satu nilai.

3. Get system info

Pada step Get system info digunakan untuk mengambil informasi waktu sistem saat proses ETL berlangsung. Dalam konteks ini, atribut yang dihasilkan adalah "Last Update" dengan tipe System Date (Fixed) yang bertujuan untuk mencatat waktu terbaru ketika proses memasukkan atau memperbarui data ke dalam tabel OLAP dim_rainfall. Pencatatan ini penting karena memungkinkan sistem untuk

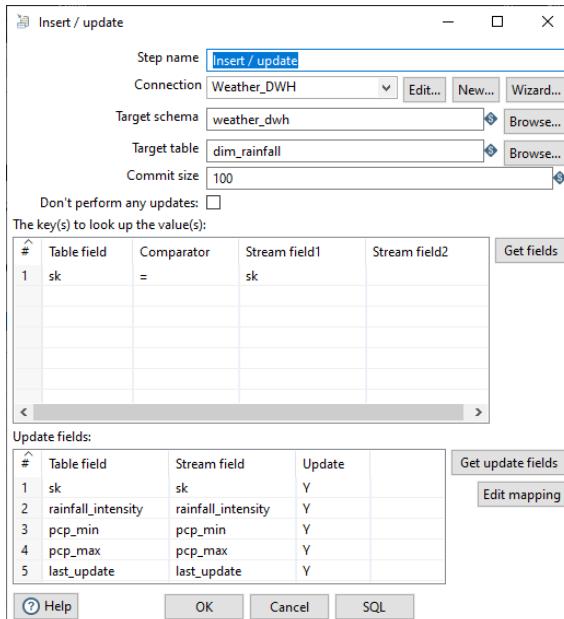
mengetahui kapan data terakhir diperbarui, memastikan bahwa hanya data yang lebih baru yang akan diproses di masa mendatang.



Gambar 4.108. Konfigurasi Step "Get system info" dalam Transformasi Data Tabel "dim_rainfall"

4. Insert/update

Step ini digunakan untuk memasukkan atau memperbarui data ke dalam tabel target, yaitu dim_rainfall dalam skema weather_dwh. Kunci utama yang digunakan untuk mencocokkan data adalah sk. Jika data dengan sk yang sesuai ditemukan, maka proses update dilakukan. Jika tidak ditemukan, data baru akan dimasukkan.



Gambar 4.109. Konfigurasi Step "Insert/update" dalam Transformasi Data Tabel "dim_rainfall"

Bagian Update field menunjukkan kolom-kolom yang akan diperbarui dalam tabel target berdasarkan aliran data yang masuk, seperti sk, rainfall_intensity, pcp_min, dan pcp_max. Field last_update juga diperbarui dengan waktu terbaru yang diambil dari tahap Get System Info sebelumnya, memastikan setiap data yang di-update memiliki jejak waktu yang tepat.

5. Hasil

Potongan data dari dim_rainfall tampak pada gambar dibawah dimana tabel dim_location memiliki 5 kolom dengan 6 baris yang menunjukkan tipe rainfall intensity yang didefinisikan manual di awal.

sk	rainfall_intensity	pcp_min	pcp_max	last_update
1	Cloudy	0.0	0.4	2024-12-13 20:21:04
2	Light rain	0.5	20.0	2024-12-13 20:21:04
3	Moderate rain	20.1	50.0	2024-12-13 20:21:04
4	Heavy rain	50.1	100.0	2024-12-13 20:21:04
5	Very heavy rain	100.1	150.0	2024-12-13 20:21:04
6	Extreme rain	151.0	1000.0	2024-12-13 20:21:04

Gambar 4.110. Hasil Transformasi Data Tabel “dim_rainfall” dalam Basis Data OLAP

c. Tabel Dimensi Waktu (dim_time)

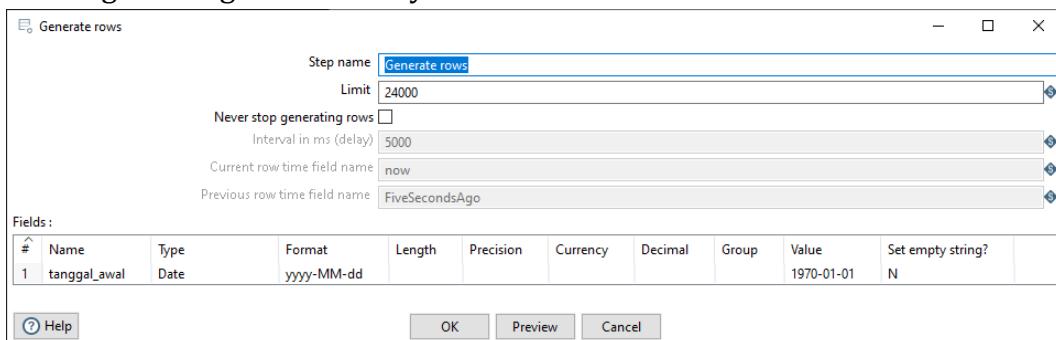


Gambar 4.111. Rangkaian Step PDI untuk Mentransformasikan Data Tabel “dim_time”

Proses pada gambar di atas menggambarkan alur ETL untuk melakukan impor data ke dalam tabel dim_time dalam basis data OLAP. Berikut adalah penjelasan dari setiap step yang digunakan:

1. Generate rows

Step “Generate rows” digunakan untuk membuat baris data dasar dengan tanggal awal yang terdefinisi, yaitu 1970-01-01. Output dari step ini adalah satu kolom (tanggal_awal) dengan nilai tanggal 1970-01-01 yang akan digunakan sebagai input untuk langkah-langkah berikutnya.

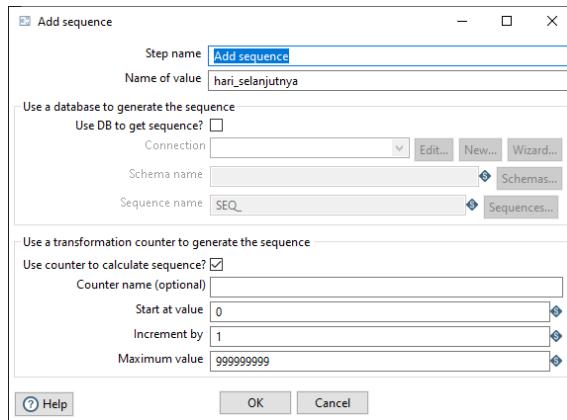


Gambar 4.112. Konfigurasi Step "Generate rows" dalam Transformasi Data Tabel “dim_time”

Nama kolom yang dihasilkan adalah tanggal_awal dengan tipe datanya adalah Date, sesuai kebutuhan untuk atribut waktu. Format tanggal yang ditentukan adalah yyyy-MM-dd. Nilai awal tanggal diisi 1970-01-01 yang menjadi dasar iterasi untuk langkah selanjutnya.

2. Add sequences

Step “Add Sequence” berfungsi untuk menambahkan nomor urut ke setiap baris data yang dalam studi kasus ini adalah untuk menambahkan tanggal – tanggal berikutnya dengan setiap baris baru merepresentasikan hari selanjutnya.

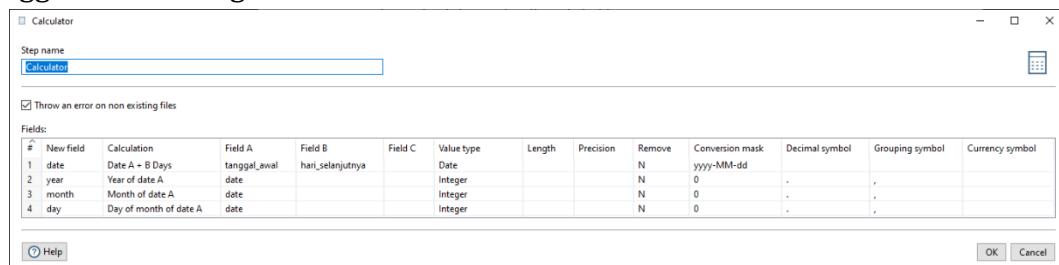


Gambar 4.113. Konfigurasi Step "Add sequences" dalam Transformasi Data Tabel "dim_time"

Gambar di atas merupakan pengaturan untuk step "Add sequence" dengan menunjukkan nomor urut yang ditambahkan akan memiliki kolom bernama hari_selanjutnya dengan rentang dari 0 hingga 999.999.999 dengan setiap perpindahan baris, urutan akan ditambahkan sejumlah satu nilai.

3. Calculator

Step "Calculator" berfungsi untuk melakukan kalkulasi tambahan pada kolom tanggal untuk menghasilkan atribut turunan.



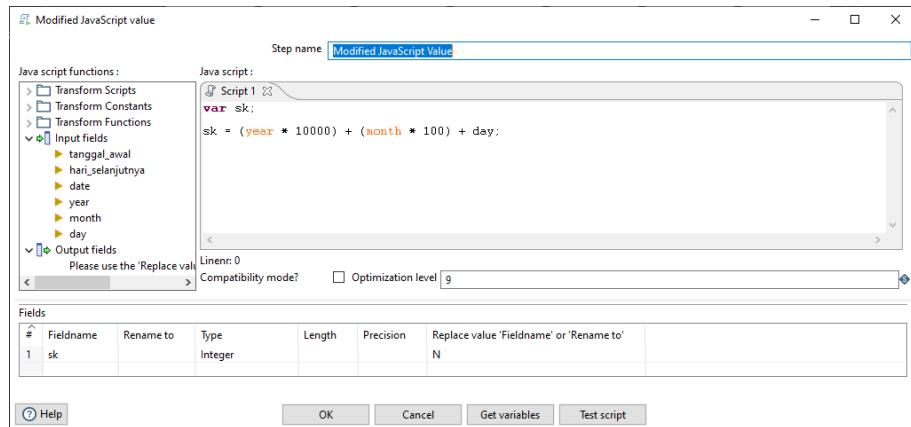
Gambar 4.114. Konfigurasi Step "Calculator" dalam Transformasi Data Tabel "dim_time"

Dalam studi kasus ini, terdapat empat kolom atribut turunan dari tanggal, diantaranya adalah sebagai berikut:

- Date: menggunakan kalkulasi "Date A + B Days" yang artinya menambahkan jumlah hari (Field B) ke tanggal awal (Field A)
- Year: menggunakan kalkulasi "Year of date A" untuk mengekstrak tahun dari kolom "date"
- Month: menggunakan kalkulasi "Month of date A" untuk mengekstrak bulan dari kolom "date"
- Day: menggunakan kalkulasi "Day of month of date A" untuk mengekstrak tahun dari hari atau tanggal dari kolom "date"

4. Modified JavaScript value

Step "Modified JavaScript value" digunakan untuk membuat Surrogate Key (SK) menggunakan skrip JavaScript. SK adalah kunci unik untuk setiap baris dalam tabel dimensi waktu.

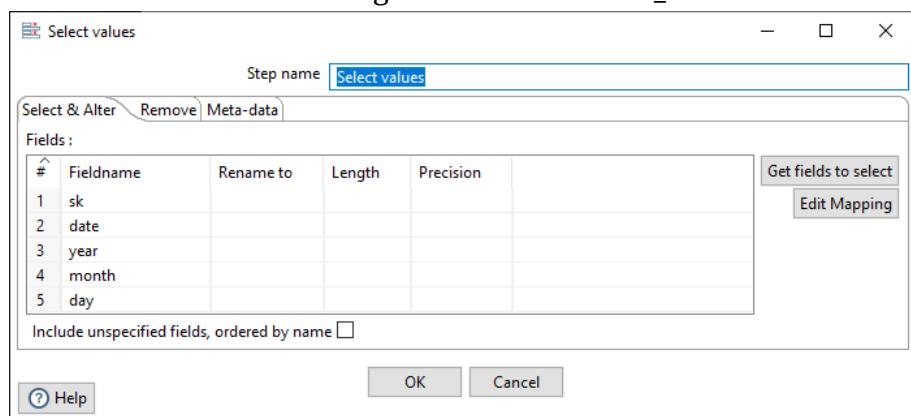


Gambar 4.115. Konfigurasi Step "Modified JavaScript value" dalam Transformasi Data Tabel "dim_time"

JavaScript diatas menunjukkan untuk membuat kolom baru bernama "sk" yang berisi kalkulasi dari $(year * 10000) + (month * 100) + day$. Sebagai contoh, tanggal 2024-11-30 akan memiliki SK dengan nilai 20241130.

5. Select values

Step "Select values" berfungsi untuk memilih kolom yang relevan dari database untuk memastikan data sesuai dengan desain tabel dim_time.

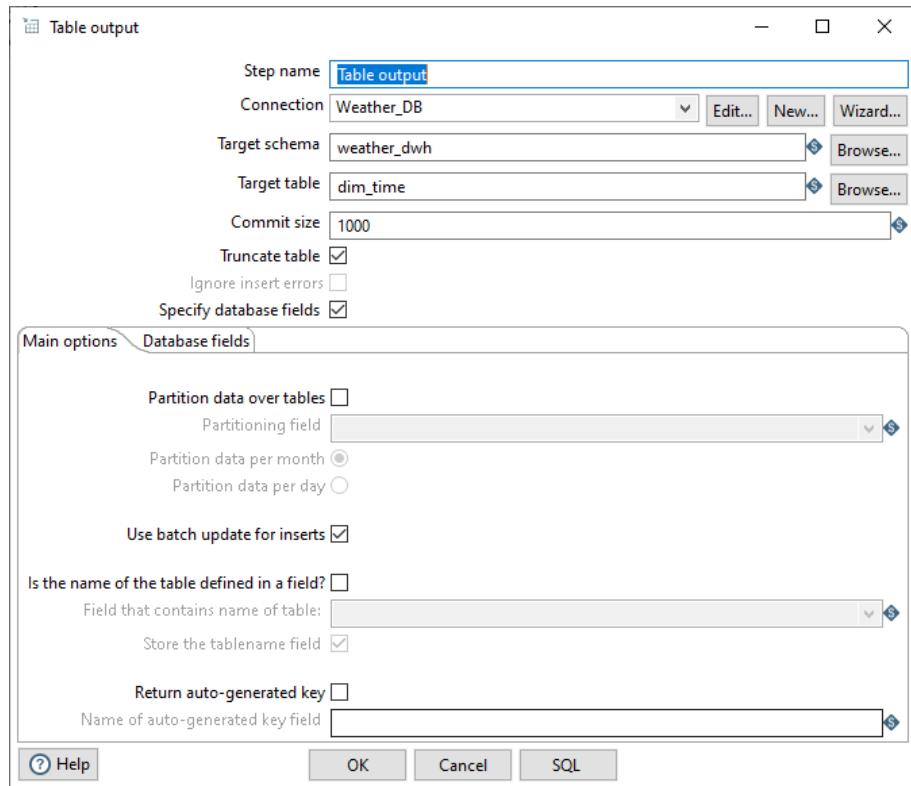


Gambar 4.116. Konfigurasi Step "Select values" dalam Transformasi Data Tabel "dim_time"

Dalam studi kasus ini, kolom yang dipilih adalah kolom "sk", "date", "year", "month", "day". Kolom ini juga telah disesuaikan dengan skema database OLAP yang sudah diinisialisasi sebelumnya.

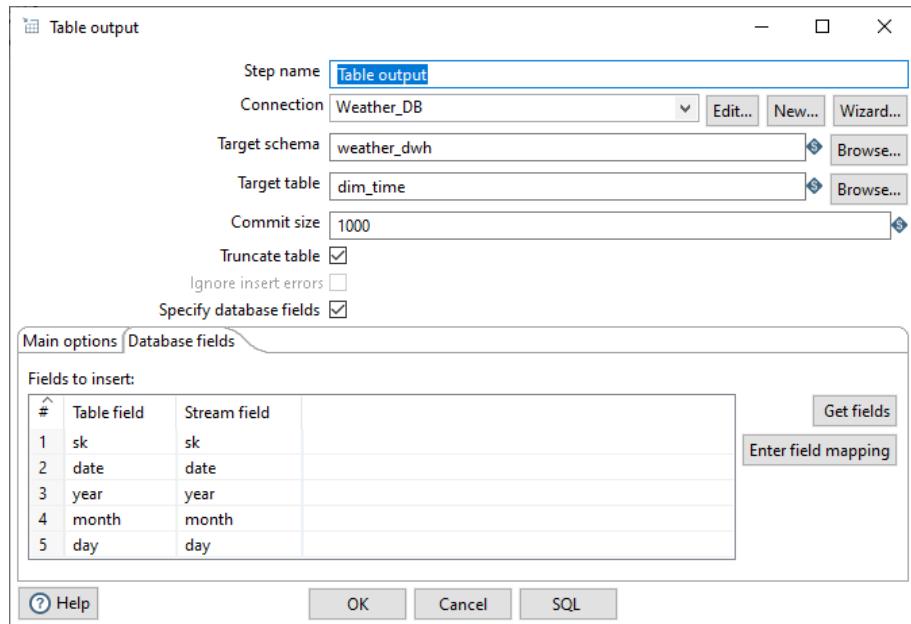
6. Tabel output

Step "Table output" digunakan untuk menyimpan hasil transformasi ke tabel database dim_time.



Gambar 4.117. Konfigurasi Step "Table output" pada tab "Main options" dalam Transformasi Data Tabel "dim_time"

Dalam studi kasus ini, data akan tersimpan dalam tabel dim_time yang memiliki database bernama weather_dwh. Koneksi sudah dibuat sebelumnya dengan nama "Weather_DB". Pada tab "Main options", centang pilihan "Use batch update for inserts" agar mengirimkan banyak baris data ke database dalam satu kali operasi batch (batch insert).



Gambar 4.118. Konfigurasi Step "Table output" pada tab "Database fields" dalam Transformasi Data Tabel "dim_time"

Pada tab "Database field", definisikan atribut yang akan disimpan dalam tabel dim_time. Dalam studi kasus ini, atribut yang dikirimkan adalah "sk", "date", "year", "month", "day".

7. Hasil

Potongan data dari dim_time tampak pada gambar dibawah dimana tabel dim_location memiliki 5 kolom dengan 24000 baris. Terlihat bahwa SK merupakan gabungan dari year, month, dan day sesuai kalkulasi yang ada pada step "Modified JavaScript value".

	sk	date	year	month	day
1	19700101	1970-01-01	1970	1	1
2	19700102	1970-01-02	1970	1	2
3	19700103	1970-01-03	1970	1	3
4	19700104	1970-01-04	1970	1	4
5	19700105	1970-01-05	1970	1	5
6	19700106	1970-01-06	1970	1	6
7	19700107	1970-01-07	1970	1	7
8	19700108	1970-01-08	1970	1	8
9	19700109	1970-01-09	1970	1	9
10	19700110	1970-01-10	1970	1	10
11	19700111	1970-01-11	1970	1	11
12	19700112	1970-01-12	1970	1	12
13	19700113	1970-01-13	1970	1	13
14	19700114	1970-01-14	1970	1	14
15	19700115	1970-01-15	1970	1	15
16	19700116	1970-01-16	1970	1	16
17	19700117	1970-01-17	1970	1	17
18	19700118	1970-01-18	1970	1	18
19	19700119	1970-01-19	1970	1	19
20	19700120	1970-01-20	1970	1	20
21	19700121	1970-01-21	1970	1	21
22	19700122	1970-01-22	1970	1	22
23	19700123	1970-01-23	1970	1	23
24	19700124	1970-01-24	1970	1	24
25	19700125	1970-01-25	1970	1	25
26	19700126	1970-01-26	1970	1	26
27	19700127	1970-01-27	1970	1	27
28	19700128	1970-01-28	1970	1	28
29	19700129	1970-01-29	1970	1	29
30	19700130	1970-01-30	1970	1	30
31	19700131	1970-01-31	1970	1	31

Gambar 4.119. Potongan Hasil Transformasi Data Tabel "dim_time" dalam Basis Data OLAP

d. Tabel Dimensi Kondisi Cuaca (dim_weather_condition)

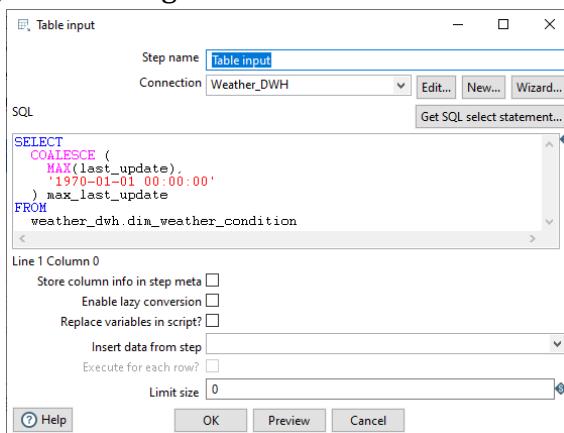


Gambar 4.120. Rangkaian Step PDI untuk Mentransformasikan Data Tabel "dim_weather_condition"

Proses pada gambar di atas menggambarkan alur ETL untuk melakukan impor data ke dalam tabel dim_weather_condition dalam basis data OLAP. Berikut adalah penjelasan dari setiap step yang digunakan:

1. Table input

Fungsi dari Table Input adalah untuk mengambil timestamp dari pembaruan terakhir (last update) yang tercatat di tabel dim_location pada Data Warehouse (DWH). Fungsi ini berguna untuk memastikan bahwa hanya data terbaru (yang diperbarui setelah last update) yang akan diambil dari database weather_db. Table input memiliki pengaturan sebagai berikut.

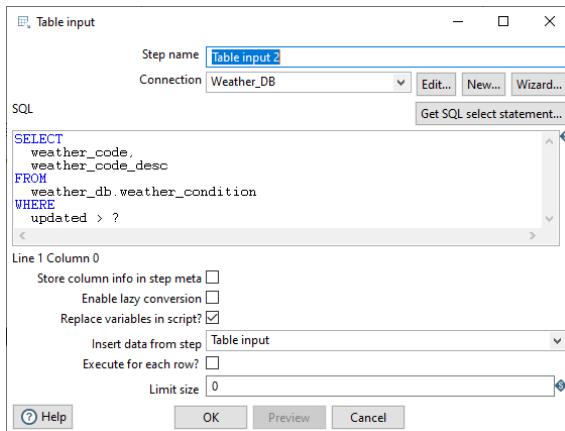


Gambar 4.121. Konfigurasi Step "Table input" dalam Transformasi Data Tabel "dim_weather_condition"

Langkah dalam SQL mengambil data max_last_update dari tabel dim_weather_condition di DWH (Data Warehouse). Kueri ini memastikan jika tidak ada data yang tersedia di dim_location, maka akan menggunakan tanggal default "1970-01-01 00:00:00" namun, jika ada data yang tersedia, digunakan tanggal yang tertera pada "last update".

2. Table input 2

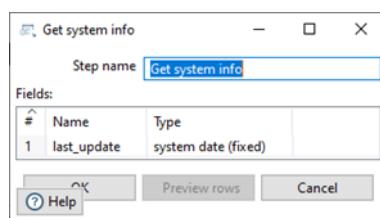
Fungsi dari Table Input 2 adalah untuk mengambil data kode cuaca dan deskripsi kode cuaca yang ada di database weather_db dengan tabel bernama weather_condition. Digunakan "WHERE updated > ?" yang memiliki arti untuk melakukan filter dengan hanya mengambil baris dimana kolom updated lebih besar dari parameter (variabel) yang diberikan. Tanda "?" memiliki makna menunjukkan tempat variabel akan diganti selama eksekusi.



Gambar 4.122. Konfigurasi Step "Table input 2" dalam Transformasi Data Tabel "dim_weather_condition"

Centang pilihan "Replace variabels in script?", maka tanda "?" diganti menjadi nilai variabel yang dikirimkan oleh Table input 1 atau dalam kasus ini adalah max_last_updated.

3. Get system info

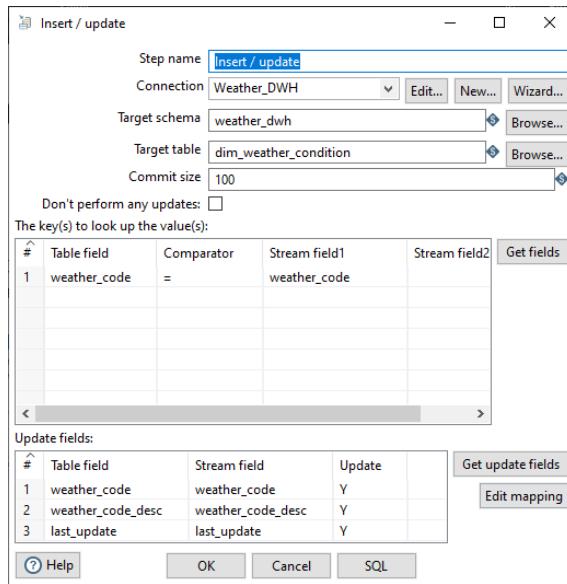


Gambar 4.123. Konfigurasi Step "Get system info" dalam Transformasi Data Tabel "dim_weather_condition"

Pada step Get System Info digunakan untuk mengambil informasi waktu sistem saat proses ETL berlangsung. Dalam konteks ini, atribut yang dihasilkan adalah "Last Update" dengan tipe System Date (Fixed) yang bertujuan untuk mencatat waktu terbaru ketika proses memasukkan atau memperbarui data ke dalam tabel OLAP dim_weather_condition.

4. Insert/update

Pada tahap Insert/Update, step ini digunakan untuk memasukkan atau memperbarui data ke dalam tabel target, yaitu dim_weather_condition dalam skema weather_dwh. Kunci utama yang digunakan untuk mencocokkan data adalah weather_code. Jika data dengan weather_code yang sesuai ditemukan, maka proses update dilakukan. Jika tidak ditemukan, data baru akan dimasukkan.



Gambar 4.124. Konfigurasi Step "Insert/update" dalam Transformasi Data Tabel "dim_weather_condition"

Bagian Update field menunjukkan kolom-kolom yang akan diperbarui dalam tabel target berdasarkan aliran data yang masuk, seperti weather_code dan weather_code_desc. Field last_update juga diperbarui dengan waktu terbaru yang diambil dari tahap Get System Info sebelumnya, memastikan setiap data yang di-update memiliki jejak waktu yang tepat.

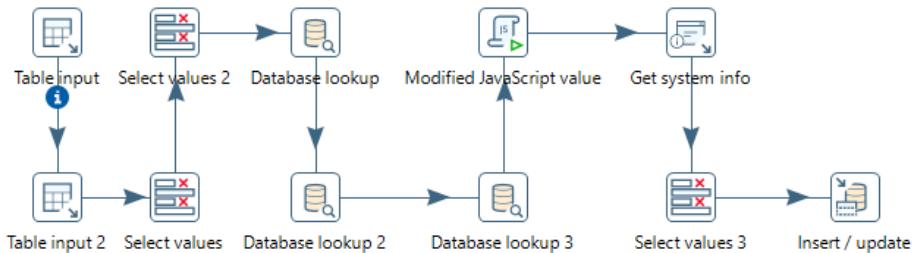
5. Hasil

Potongan data dari dim_weather_condition tampak pada gambar dibawah dimana tabel dim_location memiliki 4 kolom dengan 100 baris yang menunjukkan deskripsi cuaca untuk tiap kode cuaca.

weather_code	weather_code_desc	last_update
0	Cloud development not observed or not observable	2024-12-13 20:29:41
1	Cloud generally dissolving or becoming less developed	2024-12-13 20:29:41
2	State of sky on the whole unchanged	2024-12-13 20:29:41
3	Clouds generally forming or developing	2024-12-13 20:29:41
4	Visibility reduced by smoke, e.g. velvet or forest fires, industrial smoke	2024-12-13 20:29:41
5	Haze	2024-12-13 20:29:41
6	Widespread dust in suspension in the air, not raised by wind at or near the station at the time of observation	2024-12-13 20:29:41
7	Dust or sand raised by wind at or near the station at the time of observation	2024-12-13 20:29:41
8	Belt-developed dust or sand whirl(s) seen at or near the station during the	2024-12-13 20:29:41
9	Duststorm or sandstorm within sight at the time of observation, or at the	2024-12-13 20:29:41
10	Mist	2024-12-13 20:29:41
11	Patches of shallow fog or ice fog at the station, whether on land or sea n.	2024-12-13 20:29:41
12	More or less continuous shallow fog or ice fog at the station, whether on	2024-12-13 20:29:41
13	Lightning visible, no thunder heard	2024-12-13 20:29:41
14	Precipitation within sight, not reaching the ground or the surface of the	2024-12-13 20:29:41
15	Precipitation within sight, reaching the ground or the surface of the sea..	2024-12-13 20:29:41
16	Precipitation within sight, reaching the ground or the surface of the sea..	2024-12-13 20:29:41
17	Thunderstorm, but no precipitation at the time of observation	2024-12-13 20:29:41
18	Squalls at or within sight of the station during the preceding hour or at	2024-12-13 20:29:41
19	Funnel clouds at or within sight of the station during the preceding hour	2024-12-13 20:29:41
20	Brizzle (not freezing) or snow grains, not falling as showers, during the	2024-12-13 20:29:41
21	Rain (not freezing), not falling as showers, during the preceding hour but	2024-12-13 20:29:41
22	Snow, not falling as showers, during the preceding hour but not at the time	2024-12-13 20:29:41
23	Rain and snow or ice pellets, not falling as showers; during the preceding	2024-12-13 20:29:41
24	Freezing drizzle or freezing rain, during the preceding hour but not at the	2024-12-13 20:29:41
25	Showers(s) of rain during the preceding hour but not at the time of observa	2024-12-13 20:29:41
26	Showers(s) of snow, or of rain and snow during the preceding hour but not a	2024-12-13 20:29:41
27	Showers(s) of hail, or of rain and hail during the preceding hour but not a	2024-12-13 20:29:41
28	Fog or ice fog during the preceding hour but not at the time of observation	2024-12-13 20:29:41
29	Thunderstorm (with or without precipitation) during the preceding	2024-12-13 20:29:41
30	Slight or moderate duststorm or sandstorm - has decreased during	2024-12-13 20:29:41
31		

Gambar 4.125. Potongan Hasil Transformasi Data Tabel "dim_weather_condition" dalam Basis Data OLAP

e. Tabel Fakta (fact_weather)

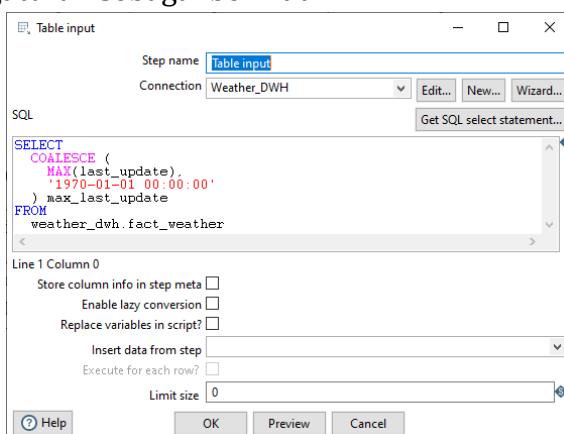


Gambar 4.126. Rangkaian Step PDI untuk Mentransformasikan Data Tabel "fact_weather"

Proses pada gambar di atas menggambarkan alur ETL (Extract, Transform, Load) untuk melakukan impor data ke dalam tabel OLAP fact_weather. Berikut adalah penjelasan dari setiap step yang digunakan:

1. Table Input

Fungsi dari Table Input adalah untuk mengambil timestamp dari pembaruan terakhir (last update) yang tercatat di tabel fact_weather pada Data Warehouse (DWH). Fungsi ini berguna untuk memastikan bahwa hanya data terbaru (yang diperbarui setelah last update) yang akan diambil dari database weather_db. Table input memiliki pengaturan sebagai berikut.



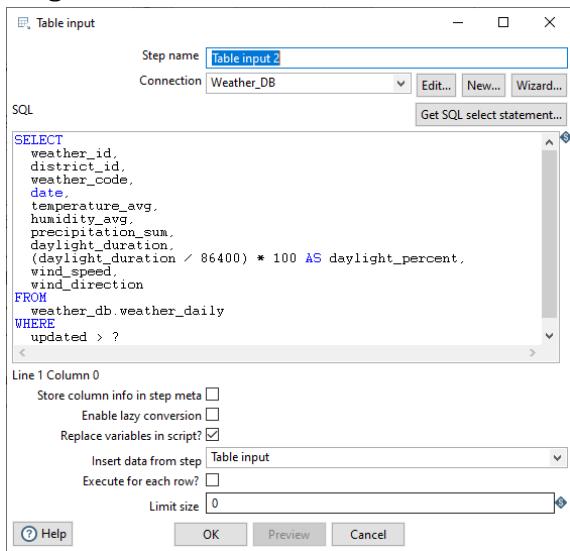
Gambar 4.127. Konfigurasi Step "Table input" dalam Transformasi Data Tabel "fact_weather"

Langkah dalam SQL mengambil data max_last_update dari tabel fact_weather di DWH (Data Warehouse). Kueri ini memastikan jika tidak ada data yang tersedia di dim_location, maka akan menggunakan tanggal default "1970-01-01 00:00:00" namun, jika ada data yang tersedia, digunakan tanggal yang tertera pada "last update".

2. Table Input 2

Fungsi dari Table Input 2 adalah untuk mengambil data cuaca seperti "weather_id", "district_id", "weather_code", "date", "temperature_avg", "humidity_avg", "precipitation_sum", "daylight_duration", "daylight_percent", "wind_speed", dan "wind_direction" yang ada di database weather_db dengan tabel bernama weather_daily. Digunakan "WHERE updated > ?" yang memiliki arti untuk melakukan filter dengan hanya mengambil baris dimana kolom updated lebih besar

dari parameter (variabel) yang diberikan. Tanda “?” memiliki makna menunjukkan tempat variabel akan diganti selama eksekusi.

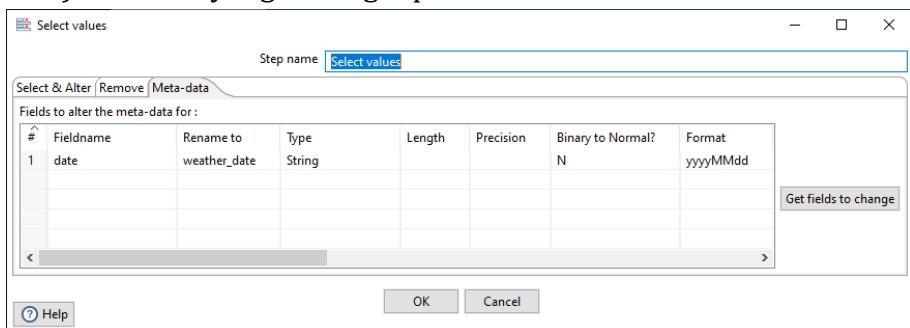


Gambar 4.128. Konfigurasi Step "Table input 2" dalam Transformasi Data Tabel "fact_weather"

Centang pilihan “Replace variabels in script?”, maka tanda “?” diganti menjadi nilai variabel yang dikirimkan oleh Table input 1 atau dalam kasus ini adalah max_last_updated.

3. Select Values

langkah Select Values digunakan untuk memilih, mengubah, atau memanipulasi field (kolom) dari data yang sedang diproses.



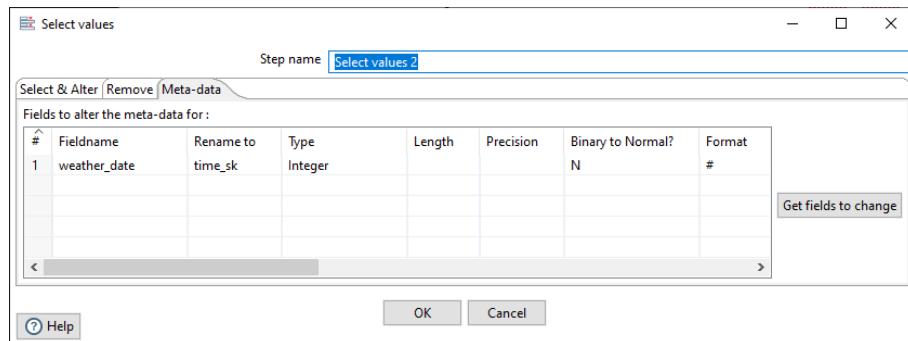
Gambar 4.129. Konfigurasi Step "Select values" dalam Transformasi Data Tabel "fact_weather"

Dalam kasus ini, pada tab “Meta-data”, nama atribut atau field “date” akan diubah menjadi weather_date. Field “date” yang awalnya memiliki tipe data date akan ditampilkan data dalam tipe data string dengan format yyyyMMdd. Ditambahkan nilai “N” pada “Binary to Normal?”, “Date Format lenient?”, dan “Lenient number conversion?”.

4. Select Values 2

Ditambahkan step “Select values 2” untuk memanipulasi field yang dibuat pada step “Select Values”. Dalam kasus ini, pada tab “Meta-data”, nama atribut atau field “weather_date” akan diubah menjadi “time_sk”. Field “weather_date” yang awalnya

memiliki tipe data string akan diubah menjadi tipe data integer. Ditambahkan nilai “N” pada “Binary to Normal?”, “Date Format lenient?”, dan “Lenient number conversion?”.

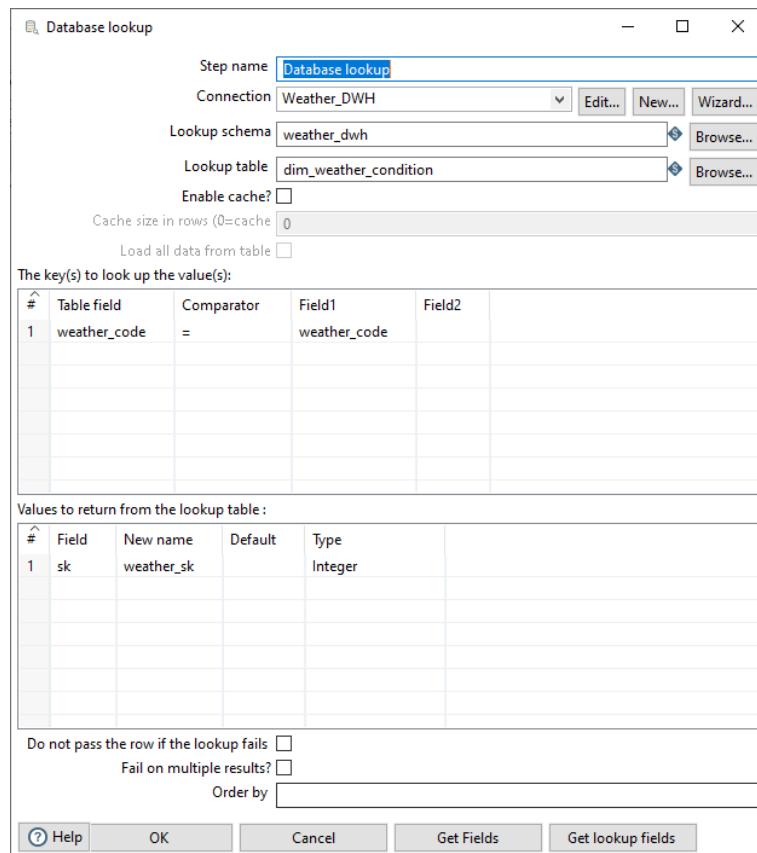


Gambar 4.130. Konfigurasi Step "Select values 2" dalam Transformasi Data Tabel "fact_weather"

Dikarenakan tipe data date tidak dapat langsung diubah menjadi tipe data integer, maka step “Select Values” diperlukan agar mengubah tipe data menjadi string terlebih dahulu.

5. Database Lookup

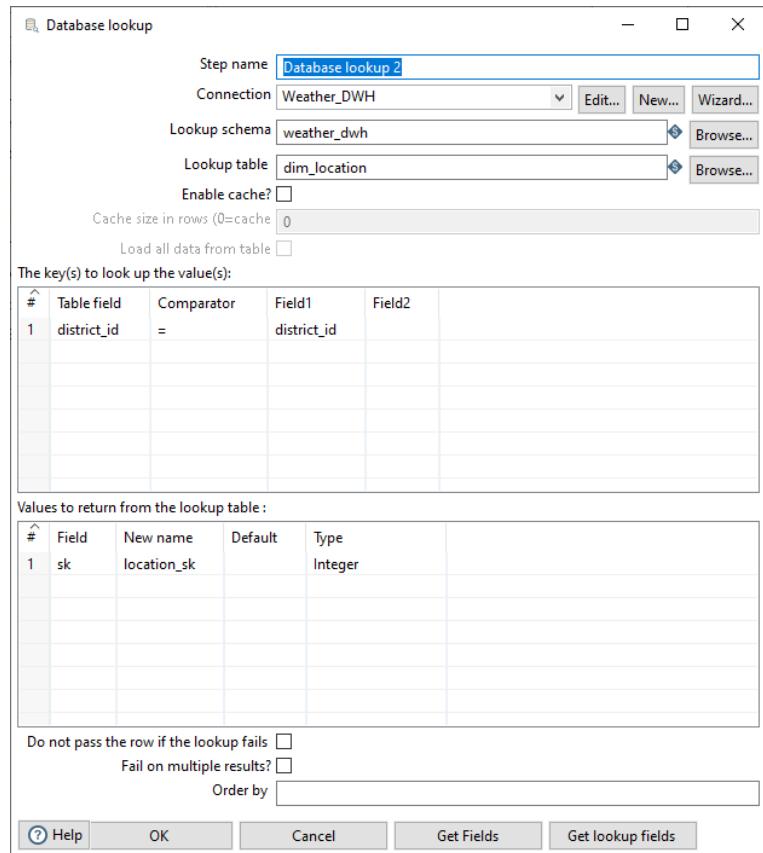
Langkah “Database Lookup” berfungsi untuk mencari nilai weather_code pada tabel dim_weather_condition dan mencocokannya dengan nilai weather_code yang ada di tabel sekarang yang diambil dari weather_daily. Jika ditemukan, nilai sk dari tabel dim_weather_condition akan diambil dan diberi nama baru weather_sk. Jika tidak ditemukan, baris data akan tetap diteruskan tanpa nilai default.



Gambar 4.131. Konfigurasi Step "Database lookup" dalam Transformasi Data Tabel "fact_weather"

6. Database Lookup 2

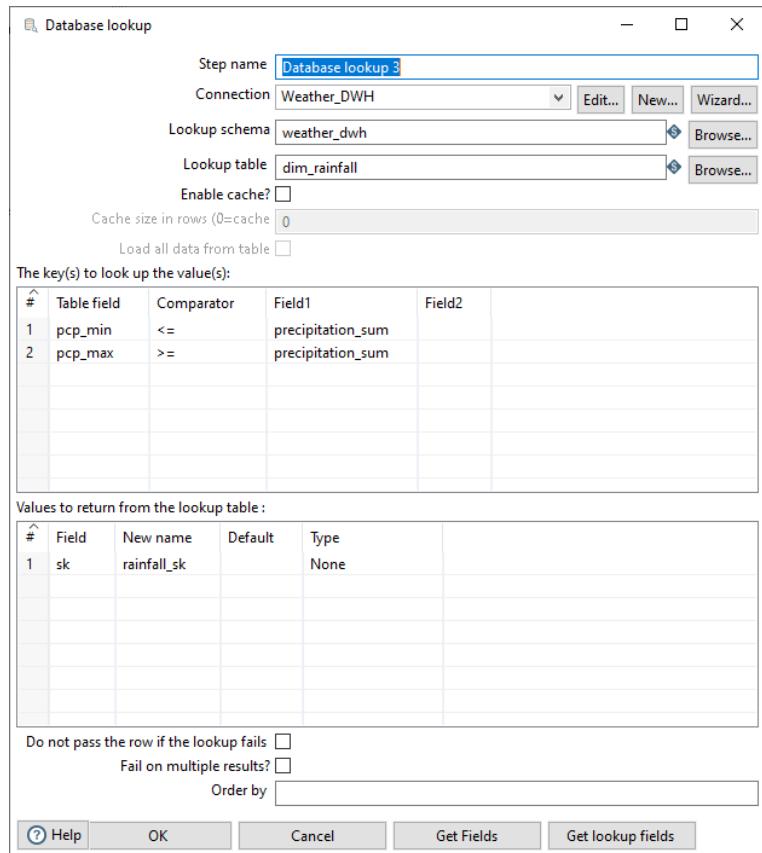
Langkah “Database Lookup 2” berfungsi untuk mencari nilai district_id pada tabel dim_location dan mencocokannya dengan nilai district_id yang ada di tabel sekarang yang diambil dari weather_daily. Jika ditemukan, nilai sk dari tabel dim_location akan diambil dan diberi nama baru location_sk. Jika tidak ditemukan, baris data akan tetap diteruskan tanpa nilai default.



Gambar 4.132. Konfigurasi Step "Database lookup 2" dalam Transformasi Data Tabel "fact_weather"

7. Database Lookup 3

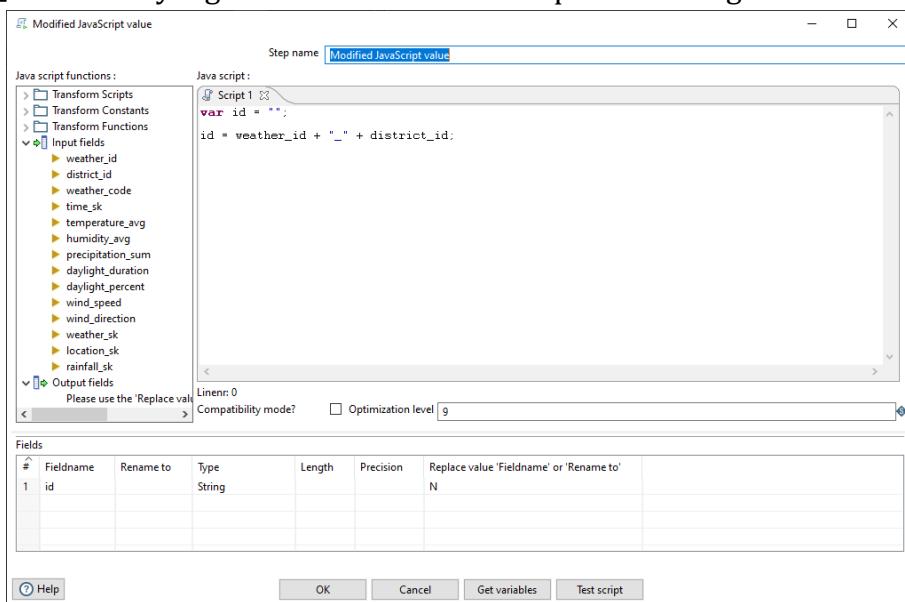
Langkah “Database Lookup 3” berguna untuk mencari rentang curah hujan dari tabel referensi (dim_rainfall) berdasarkan nilai precipitation_sum. Nilai yang ditemukan digunakan untuk menambahkan rainfall_sk yang sesuai ke dataset. Jika data tidak cocok, baris tetap diteruskan tanpa hasil lookup.



Gambar 4.133. Konfigurasi Step "Database lookup 3" dalam Transformasi Data Tabel "fact_weather"

8. Modified JavaScript value

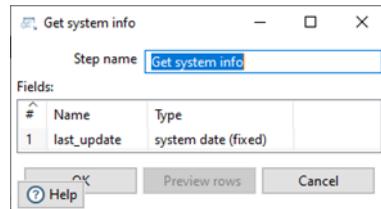
Step "Modified JavaScript value" berfungsi untuk menambahkan field id yang berasal dari weather_id ditambahkan underscore (_) dan ditambahkan dengan district_id. Field id yang ditambahkan akan bertipe data string.



Gambar 4.134. Konfigurasi Step "Modified JavaScript value" dalam Transformasi Data Tabel "fact_weather"

9. Get system info

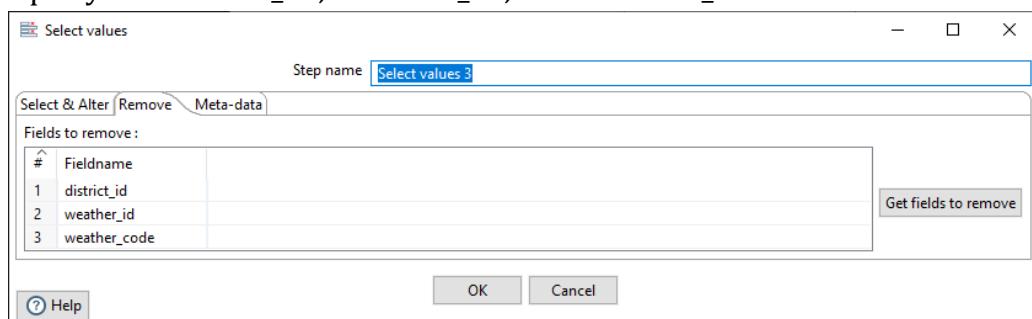
Pada step Get system info digunakan untuk mengambil informasi waktu sistem saat proses ETL berlangsung. Dalam konteks ini, atribut yang dihasilkan adalah "Last Update" dengan tipe System Date (Fixed) yang bertujuan untuk mencatat waktu terbaru ketika proses memasukkan atau memperbarui data ke dalam tabel OLAP fact_weather.



Gambar 4.135. Konfigurasi Step "Get System info" dalam Transformasi Data Tabel "fact_weather"

10. Select Values 3

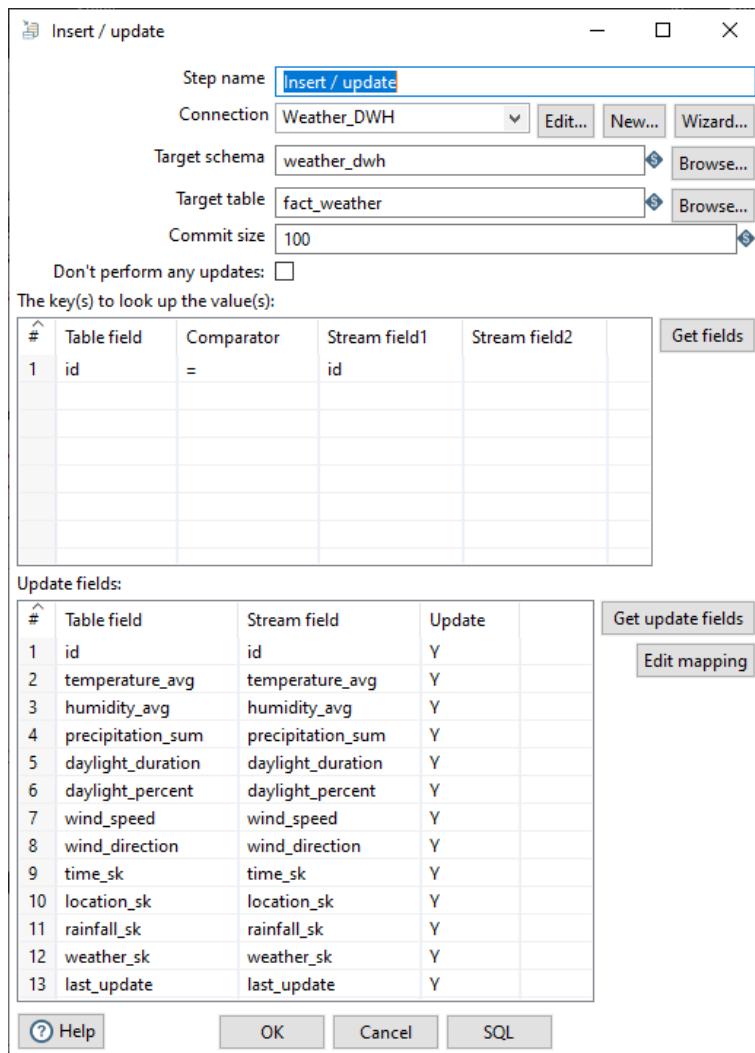
Step "Select Values 3" berfungsi untuk menghapus field yang tidak dibutuhkan dan tidak termasuk dalam fact_weather. Dalam kasus ini, akan ada tiga atribut yang dihapus yaitu "district_id", "weather_id", dan "weather_code".



Gambar 4.136. Konfigurasi Step "Select Values 3" dalam Transformasi Data Tabel "fact_weather"

11. Insert/update

Step "Insert/Update" digunakan untuk memasukkan atau memperbarui data ke dalam tabel target, yaitu fact_weather dalam skema weather_dwh. Kunci utama yang digunakan untuk mencocokkan data adalah id. Jika data dengan id yang sesuai ditemukan, maka proses update dilakukan. Jika tidak ditemukan, data baru akan dimasukkan.



Gambar 4.137. Konfigurasi Step "Insert/update" dalam Transformasi Data Tabel "fact_weather"

Bagian Update fields menunjukkan kolom-kolom yang akan diperbarui dalam tabel target yaitu "id", "temperature_avg", "humidity_avg", "precipitation_sum", "daylight_duration", "daylight_percent", "wind_speed", "wind_direction", "time_sk", "location_sk", "rainfall_sk", "rainfall_sk", dan "weather_sk". Field last_update juga diperbarui dengan waktu terbaru yang diambil dari tahap Get System Info sebelumnya, memastikan setiap data yang di-update memiliki jejak waktu yang tepat.

12. Hasil

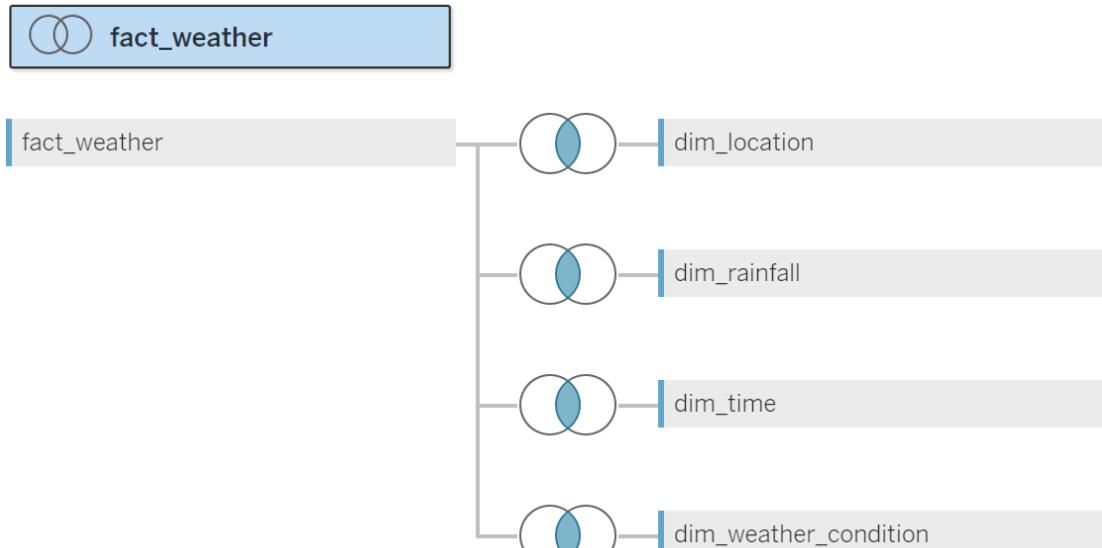
Potongan data dari fact_weather tampak pada gambar dibawah dimana tabel dim_location memiliki 13 kolom dengan 282131 baris yang menunjukkan data cuaca untuk tiap hari dari tanggal 1 Januari 2000 – 30 November 2024.

	date	temperature_avg	humidity_avg	precipitation_mm	daylight_duration	daylight_percent	wind_speed	wind_direction	time_utc	location_pk	rainfall_mm	weather_pk	last_update
1	1.2.2012	26.3	61	0.3	42349.25	52.1	8.2	286	20000101	1	2	42 2012-12-27 20:00:00	
2	10.3.2012	25.4	73	18.9	40972.56	52.2	7.6	271	20000110	1	2	44 2012-12-27 20:00:00	
3	10000_375700	25.7	65	10.9	42338.79	49.9	10.8	299	20000109	1	2	44 2012-12-27 20:00:00	
4	10000_375700	25.8	55	0.1	42339.02	50.5	6.7	128	20000104	3	1	52 2012-12-27 20:00:00	
5	10000_375700	25.8	50	4.8	42339.02	49.7	7.4	260	20000107	29	2	44 2012-12-27 20:00:00	
6	10000_375700	25.6	50	0.8	42346.72	50.1	11.8	95	20000104	29	1	52 2012-12-27 20:00:00	
7	10000_375700	25.6	70	0.0	42346.82	50.5	15.5	122	20000125	28	1	4 2012-12-27 20:00:00	
8	10000_375700	25.8	67	0.2	42348.91	50.8	11.7	131	20000126	28	1	52 2012-12-27 20:00:00	
9	10000_375700	25.9	64	0.3	42372.98	50.4	8.3	150	20000127	28	1	52 2012-12-27 20:00:00	
10	10000_375700	25.8	65	0.9	42377.41	50.4	10.9	131	20000128	29	1	4 2012-12-27 20:00:00	
11	10000_375700	25.4	64	0.9	42376.99	50.4	10.5	134	20000129	28	1	52 2012-12-27 20:00:00	
12	10000_375700	25.3	68	0.9	42376.91	50.7	14.1	133	20000130	28	1	52 2012-12-27 20:00:00	
13	10000_375700	25.4	68	0.9	42388.73	50.7	11.8	180	20000101	28	1	4 2012-12-27 20:00:00	
14	10000_375700	25.4	67	0.9	42387.47	50.7	12.8	99	20000101	28	1	4 2012-12-27 20:00:00	
15	10000_375700	25.4	64	0.5	42386.49	50.8	12.4	209	20000103	28	2	54 2012-12-27 20:00:00	
16	10000_375700	25.4	66	2.2	42373.38	49.5	7.5	240	20000128	28	2	54 2012-12-27 20:00:00	
17	10000_375700	25.7	62	0.1	42379.75	50.8	15.4	137	20000101	28	1	52 2012-12-27 20:00:00	
18	10000_375700	25.5	62	0.1	42349.31	50.8	9.4	98	20000101	28	1	52 2012-12-27 20:00:00	
19	10000_375700	25.4	67	0.5	42349.36	50.8	11.8	90	20000101	28	2	54 2012-12-27 20:00:00	
20	10000_375700	25.6	65	4.9	42343.27	50.9	8.8	93	20000107	28	2	44 2012-12-27 20:00:00	
21	10000_375700	25.6	67	9.9	42379.22	50.9	13.1	40	20000108	28	2	42 2012-12-27 20:00:00	
22	10000_375700	25.5	65	1.2	42399.21	50.9	15.9	131	20000109	28	2	52 2012-12-27 20:00:00	
23	10000_375700	25.2	62	0.4	42412.31	51.0	19.8	110	20000110	28	1	52 2012-12-27 20:00:00	
24	10000_375700	25.2	62	0.9	42407.20	51.0	21.2	137	20000111	28	1	4 2012-12-27 20:00:00	
25	10000_375700	25.3	62	0.2	42375.17	51.0	17.7	100	20000112	28	1	52 2012-12-27 20:00:00	
26	10000_375700	25.2	64	1.4	42409.20	51.0	11.3	59	20000112	28	2	54 2012-12-27 20:00:00	
27	10000_375700	25.5	68	1.5	42374.12	49.4	4.7	250	20000129	28	2	52 2012-12-27 20:00:00	
28	10000_375700	25.4	65	0.6	42338.92	51.1	15.5	79	20000114	28	2	54 2012-12-27 20:00:00	
29	10000_375700	25.4	62	0.2	42342.77	51.1	17.5	123	20000115	28	1	52 2012-12-27 20:00:00	
30	10000_375700	25.4	62	0.9	42364.66	51.1	13.4	61	20000116	28	1	1 2012-12-27 20:00:00	
31	10000_375700	25.7	63	0.5	42340.06	51.1	9.9	101	20000117	28	2	52 2012-12-27 20:00:00	
32	10000_375700	25.3	63	0.2	42323.48	51.2	13.7	94	20000118	28	1	52 2012-12-27 20:00:00	
33	10000_375700	25.2	63	0.5	42374.77	51.2	19.8	146	20000119	28	2	54 2012-12-27 20:00:00	
34	10000_375700	25.1	63	0.2	42320.46	51.2	17.7	105	20000120	28	1	52 2012-12-27 20:00:00	
35	10000_375700	25.4	59	1.4	42382.80	51.3	21.3	109	20000121	28	2	54 2012-12-27 20:00:00	
36	10000_375700	25.1	64	0.4	42335.00	51.3	12.8	107	20000122	28	1	52 2012-12-27 20:00:00	
37	10000_375700	25.7	61	2.5	42327.04	51.3	17.5	46	20000123	28	2	54 2012-12-27 20:00:00	
38	10000_375700	25.8	67	1.4	42384.97	49.4	4.9	267	20000124	28	2	54 2012-12-27 20:00:00	
39	10000_375700	25.8	67	3.4	42330.36	51.3	9.4	295	20000125	28	2	54 2012-12-27 20:00:00	
40	10000_375700	25.4	65	1.6	42372.77	51.4	10.1	13	20000126	28	2	52 2012-12-27 20:00:00	
41	10000_375700	25.7	64	9.4	42398.35	51.4	11.9	200	20000126	28	2	52 2012-12-27 20:00:00	
42	10000_375700	25.2	64	10.9	42327.49	51.4	10.9	101	20000127	28	2	44 2012-12-27 20:00:00	
43	10000_375700	25.3	62	1.4	42429.77	51.4	9.9	85	20000128	28	2	54 2012-12-27 20:00:00	

Gambar 4.138. Potongan Hasil Transformasi Data Tabel “fact_weather” dalam Basis Data OLAP

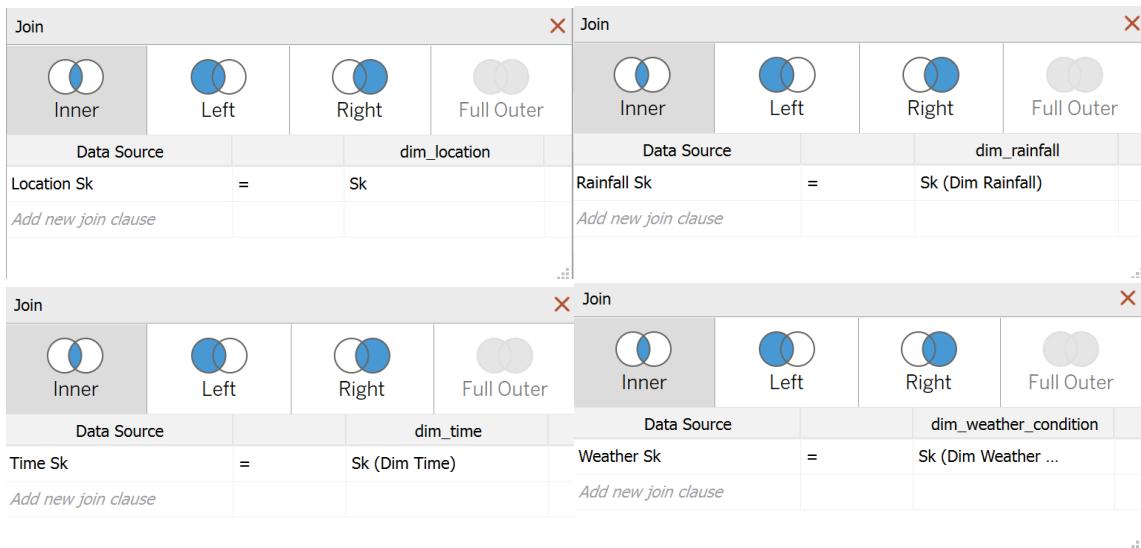
4.5. Visualisasi Dasbor

a. Membuat Data Model



Gambar 4.139. Pembuatan Data Model dalam Tableau Desktop

Data model dalam Tableau berfungsi untuk mengatur dan mengelola hubungan antara berbagai tabel atau sumber data yang digunakan dalam visualisasi. Dalam studi kasus ini, tabel fact_weather memiliki inner join dengan dim_location, dim_rainfall, dim_time, dan dim_weather_condition sehingga bentuk hubungan yang dibentuk dapat dilihat pada gambar di atas.

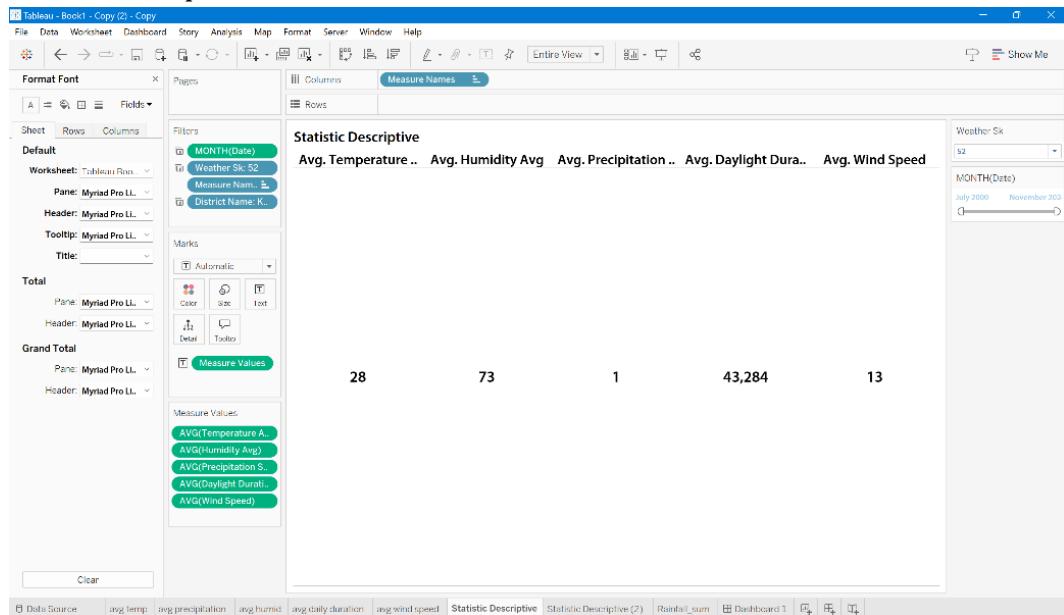


Gambar 4. 140. Bentuk Hubungan Antar Tabel dalam Basis Data OLAP pada Tableau Desktop

Umumnya, Tableau dapat secara automasi mendeteksi hubungan atau relasi antar kedua tabel namun, dapat didefinisikan juga secara manual. Gambar diatas memperlihatkan field yang berhubungan dari tabel fact_weather ke tiap tabel dimensi.

b. Membuat Worksheet

- Statistik Deskriptif

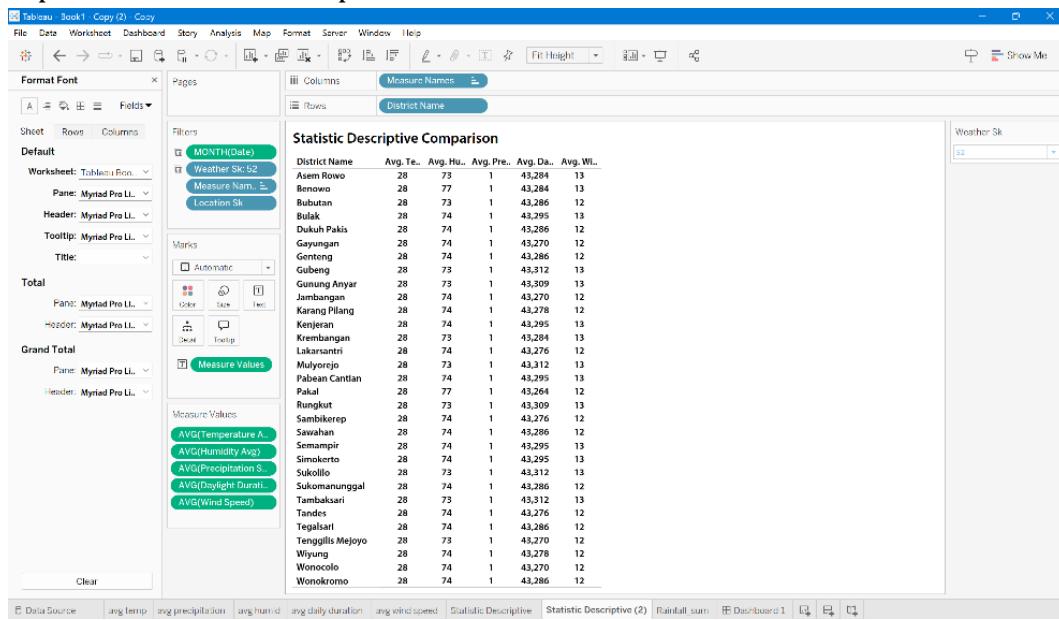


Gambar 4.141. Worksheet Statistik Deskriptif pada Tableau Desktop

Worksheet pertama adalah statistik deskriptif yang berfungsi untuk memberikan ringkasan statistik deskriptif terkait rata – rata temperatur, kelembapan, endapan, daylight duration, dan kecepatan angin yang sesuai dengan hasil filter yang diberikan. Terdapat komponen penting dalam pembuatannya yakni:

- Untuk kolom yang digunakan adalah “Measure Name” termasuk “Avg. Temperature”, “avg. Humidity”, “Avg. Precipitation”, “Avg. Daylight Duration”, dan “Avg. Wind Speed”. Sedangkan rows tidak memiliki dimensi spesifik.
- Jenis visualisasi yang digunakan berbentuk text table atau tabular data yang menggunakan teks untuk menampilkan nilai.
- Pada bagian *marks*, menggunakan “text” untuk “measure value” artinya nilai rata – rata dari tiap variabel ditampilkan dalam bentuk *text*.
- Pada bagian *Filters*, terdapat 3 jenis filter yang *filter* berdasarkan bulan-tahun, *weather sk*, dan *district name*.

- Komparasi Statistik Deskriptif

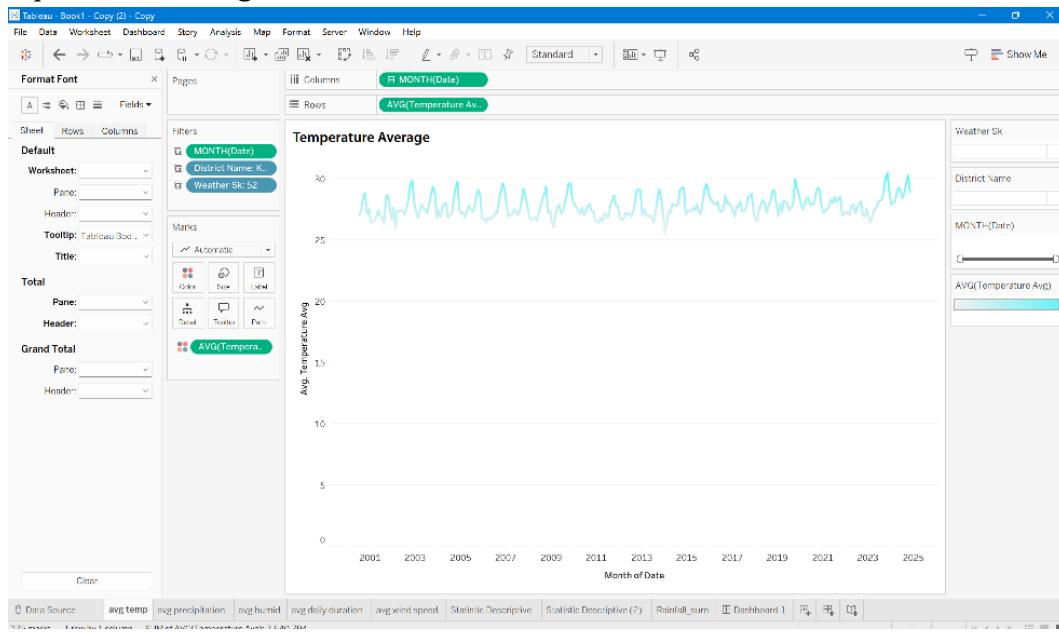


Gambar 4.142. Worksheet Komparasi Statistik Deskriptif pada Tableau Desktop

Worksheet kedua adalah komparasi statistik deskriptif dengan daerah kecamatan lain yang berfungsi untuk membandingkan ringkasan statistik deskriptif terkait rata – rata temperatur, kelembapan, endapan, daylight duration, dan kecepatan angin yang sesuai dengan hasil filter yang diberikan. Terdapat komponen penting dalam pembuatannya yakni:

- Untuk kolom yang digunakan adalah “Measure Name” termasuk “Avg. Temperature”, “avg. Humidity”, “Avg. Precipitation”, “Avg. Daylight Duration”, dan “Avg. Wind Speed”. Sedangkan rows menggunakan “District name” untuk menampilkan nama kecamatan di Surabaya.
- Jenis visualisasi yang digunakan berbentuk text table atau tabular data yang menggunakan teks untuk menampilkan nilai.
- Pada bagian *marks*, menggunakan “text” untuk “measure value” artinya nilai rata – rata dari tiap variabel ditampilkan dalam bentuk *text*.
- Pada bagian *Filters*, terdapat 2 jenis filter yang *filter* berdasarkan bulan-tahun dan *weather sk*. Filter “district names” tidak digunakan dalam *worksheet* ini.

- Temperature Average

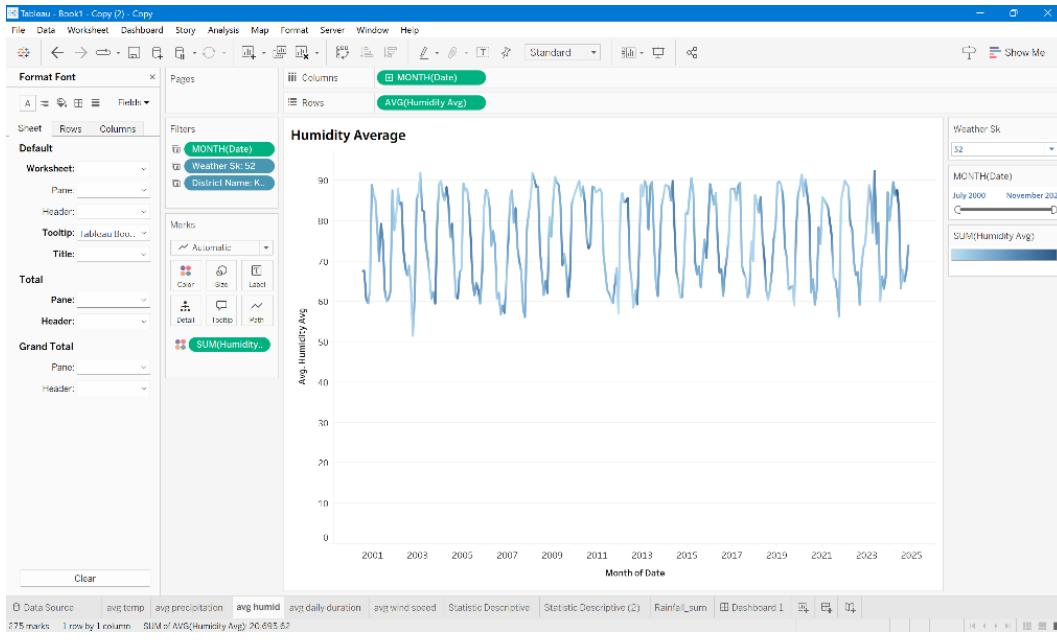


Gambar 4.143. Worksheet Rata-rata Suhu pada Tableau Desktop

Worksheet ketiga adalah visualisasi rata-rata suhu untuk melihat pola atau tren berdasarkan hasil filter yang diberikan. Terdapat komponen penting dalam pembuatannya yakni:

- Untuk kolom yang digunakan adalah “MONTH(DATE)” yang menampilkan distribusi data berdasarkan waktu dalam bulan. Sedangkan, rows yang digunakan adalah “AVG(Temperature Avg)” untuk menampilkan rata – rata suhu.
- Jenis visualisasi yang digunakan berbentuk line chart untuk menampilkan tren rata – rata suhu dari waktu ke waktu.
- Pada bagian *marks*, menggunakan “color” untuk “AVG(Temperature Avg)” artinya warna akan menjadi indikator rentang nilai dalam rata – rata suhu.
- Pada bagian *Filters*, terdapat 3 jenis filter yang *filter* berdasarkan bulan-tahun, *weather sk*, dan *district name*.

- Humidity Average

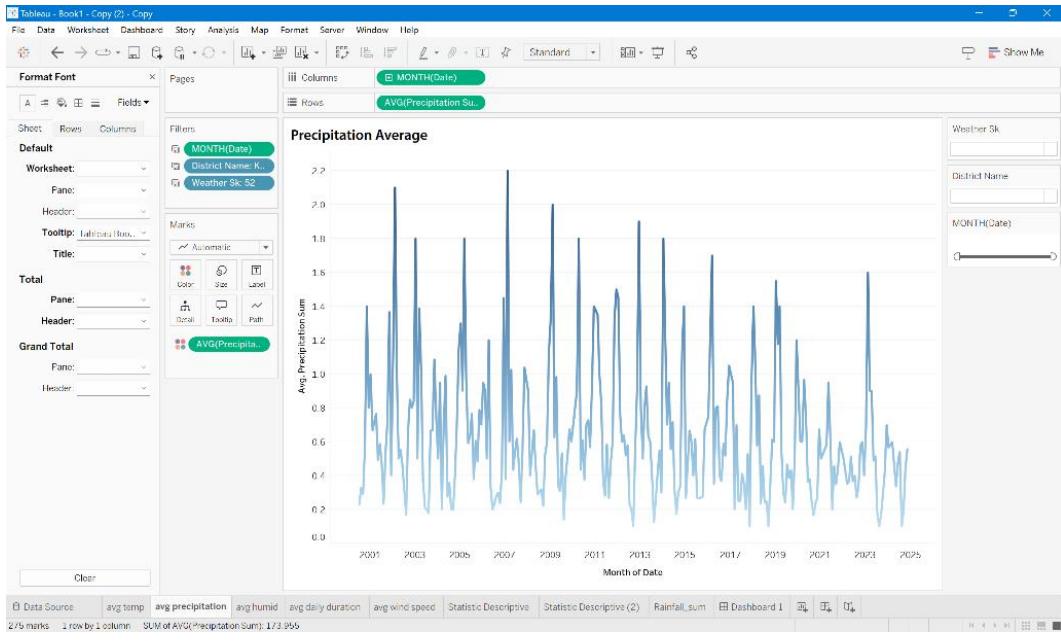


Gambar 4.144. Worksheet Rata-rata Kelembaban pada Tableau Desktop

Worksheet keempat adalah visualisasi rata – rata kelembapan untuk melihat pola atau tren berdasarkan hasil filter yang diberikan. Terdapat komponen penting dalam pembuatannya yakni:

- Untuk kolom yang digunakan adalah “MONTH(DATE)” yang menampilkan distribusi data berdasarkan waktu dalam bulan. Sedangkan, rows yang digunakan adalah “AVG(Humidity Avg)” untuk menampilkan rata – rata kelembapan.
- Jenis visualisasi yang digunakan berbentuk line chart untuk menampilkan tren rata – rata kelembapan dari waktu ke waktu.
- Pada bagian *marks*, menggunakan “color” untuk “AVG(Humidity Avg)” artinya warna akan menjadi indikator rentang nilai dalam rata – rata kelembapan.
- Pada bagian *Filters*, terdapat 3 jenis filter yang *filter* berdasarkan bulan-tahun, *weather sk*, dan *district name*.

- Precipitation Average

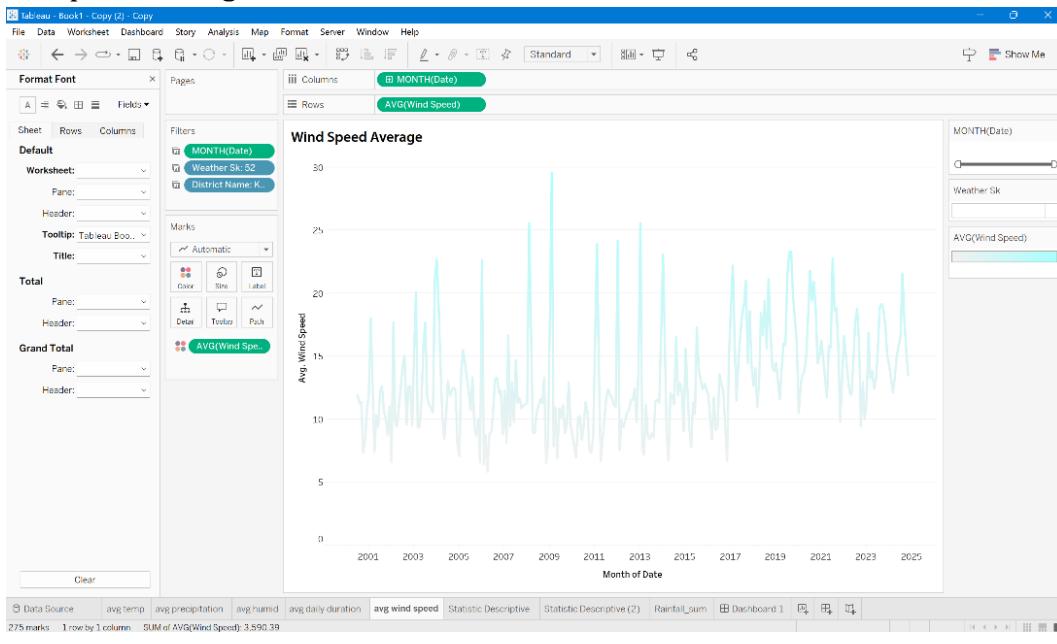


Gambar 4.145. Worksheet Rata-rata Curah Hujan pada Tableau Desktop

Worksheet kelima adalah visualisasi rata – rata endapan untuk melihat pola atau tren berdasarkan hasil filter yang diberikan. Terdapat komponen penting dalam pembuatannya yakni:

- Untuk kolom yang digunakan adalah “MONTH(DATE)” yang menampilkan distribusi data berdasarkan waktu dalam bulan. Sedangkan, rows yang digunakan adalah “AVG(Precipitation Avg)” untuk menampilkan rata – rata endapan.
- Jenis visualisasi yang digunakan berbentuk line chart untuk menampilkan tren rata – rata endapan dari waktu ke waktu.
- Pada bagian *marks*, menggunakan “color” untuk “AVG(Precipitation Avg)” artinya warna akan menjadi indikator rentang nilai dalam rata – rata endapan.
- Pada bagian *Filters*, terdapat 3 jenis filter yang *filter* berdasarkan bulan-tahun, *weather sk*, dan *district name*.

- Wind Speed Average

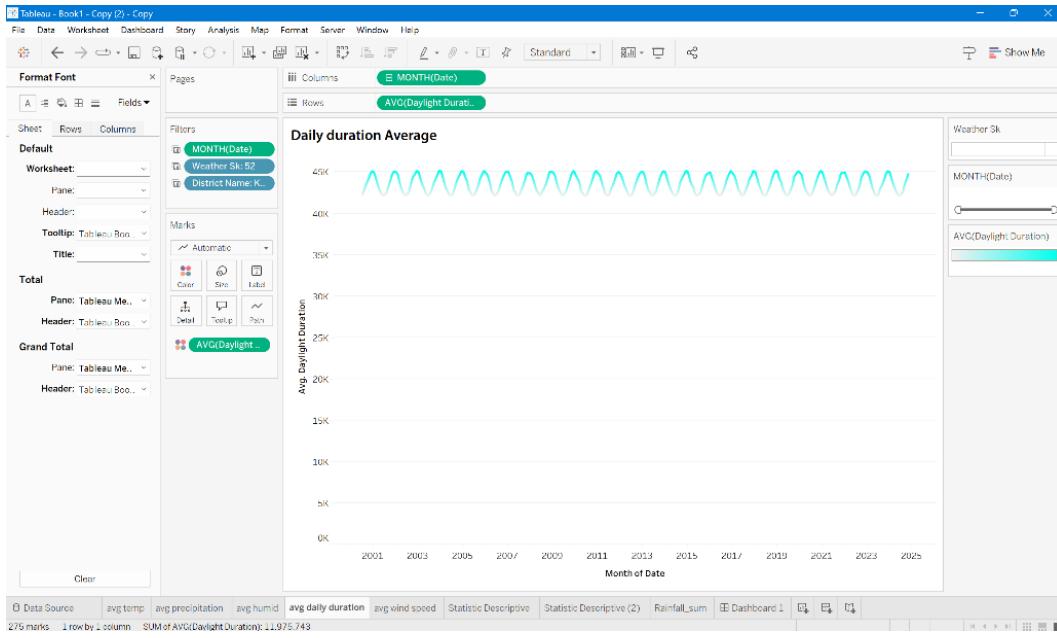


Gambar 4.146. Worksheet Rata-rata Kecepatan Angin pada Tableau Desktop

Worksheet keenam adalah visualisasi rata – rata kecepatan angin untuk melihat pola atau tren berdasarkan hasil filter yang diberikan. Terdapat komponen penting dalam pembuatannya yakni:

- Untuk kolom yang digunakan adalah “MONTH(DATE)” yang menampilkan distribusi data berdasarkan waktu dalam bulan. Sedangkan, rows yang digunakan adalah “AVG(Wind Speed)” untuk menampilkan rata – rata kecepatan angin.
- Jenis visualisasi yang digunakan berbentuk line chart untuk menampilkan tren rata – rata endapan dari waktu ke waktu.
- Pada bagian *marks*, menggunakan “color” untuk “AVG(Wind Speed)” artinya warna akan menjadi indikator rentang nilai dalam rata – rata kecepatan angin.
- Pada bagian *Filters*, terdapat 3 jenis filter yang *filter* berdasarkan bulan-tahun, *weather sk*, dan *district name*.

- Daily Duration Average

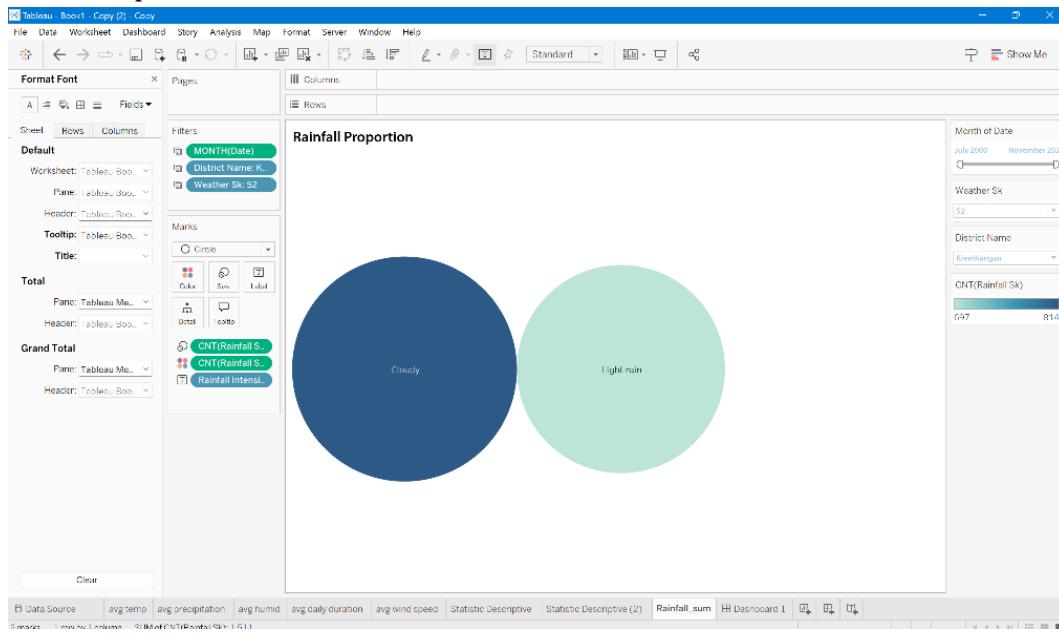


Gambar 4.147. Worksheet Rata-rata Durasi Siang Hari pada Tableau Desktop

Worksheet ketujuh adalah visualisasi rata – rata durasi daylight untuk melihat pola atau tren berdasarkan hasil filter yang diberikan. Terdapat komponen penting dalam pembuatannya yakni:

- Untuk kolom yang digunakan adalah “MONTH(DATE)” yang menampilkan distribusi data berdasarkan waktu dalam bulan. Sedangkan, rows yang digunakan adalah “AVG(Daylight Duration)” untuk menampilkan rata – rata durasi daylight.
- Jenis visualisasi yang digunakan berbentuk line chart untuk menampilkan tren rata – rata endapan dari waktu ke waktu.
- Pada bagian *marks*, menggunakan “color” untuk “AVG(Daylight Duration)” artinya warna akan menjadi indikator rentang nilai dalam rata – rata durasi dayligth.
- Pada bagian *Filters*, terdapat 3 jenis filter yang *filter* berdasarkan bulan-tahun, *weather sk*, dan *district name*.

- Rainfall Proportion



Gambar 4.148. Worksheet Proporsi Intensitas Hujan pada Tableau Desktop

Worksheet kedelapan adalah visualisasi bubble chart untuk menggambarkan distribusi proporsi jenis curah hujan berdasarkan hasil filter yang diberikan. Terdapat komponen penting dalam pembuatannya yakni:

- Untuk kolom dan rows, tidak ada nilai yang dimasukkan.
- Jenis visualisasi yang digunakan berbentuk bubble chart untuk menampilkan distribusi jenis curah hujan dengan lingkaran yang semakin besar menunjukkan jumlah yang lebih tinggi.
- Pada bagian *marks*, menggunakan "size" yang berarti ukuran bubble dipengaruhi nilai "CNT(Rainfall Sk)", "Color" yang berarti tiap bubble memiliki warna berbeda untuk merepresentasikan jenis curah hujan yang berbeda, dan "label" untuk menampilkan nama kataggori atau jenis curah hujan.
- Pada bagian *Filters*, terdapat 3 jenis filter yang *filter* berdasarkan bulan-tahun, *weather sk*, dan *district name*.

c. Membuat Dasbor dan Analisis Data



Gambar 4.149. Dasbor Analisis Data Historis Cuaca Harian pada Tableau Desktop

Dashboard berjudul “Historical Weather Analytics”, berfungsi untuk menganalisis data historis cuaca berdasarkan beberapa indikator cuaca seperti rata-rata suhu, kelembapan, curah hujan, durasi siang, dan kecepatan angin. Dashboard ini mencakup berbagai visualisasi seperti line chart, bubble chart, dan tabel deskriptif untuk memberikan informasi tentang tren dan distribusi data cuaca kecamatan yang ada di Surabaya. Terdapat tiga filter interaktif diterapkan, yaitu Month of Date, Weather Sk, dan District Name yang memungkinkan pengguna untuk menyaring data berdasarkan waktu, jenis cuaca, dan wilayah. Berdasarkan dashboard, didapatkan hasil analisa sebagai berikut:

- Rata-rata suhu udara pada kondisi cuaca berawan berada pada kisaran 28 hingga 29 derajat Celsius, sementara pada kondisi gerimis rata-rata suhu mencapai 28 derajat Celsius. Saat hujan turun, suhu udara menurun menjadi sekitar 26 hingga 27 derajat Celsius. Hal ini menunjukkan adanya penurunan suhu yang konsisten saat terjadi hujan dan sedikit peningkatan saat cuaca berawan. Meskipun demikian, perbedaan suhu antar kondisi cuaca tersebut tidak terlalu signifikan. Selain itu, tingkat suhu yang lebih tinggi umumnya terjadi pada akhir tahun di sekitar bulan September dan Oktober.
- Rata-rata tingkat kelembapan (humidity) pada kondisi cuaca berawan berada di bawah 70%, yaitu dalam kisaran 65% hingga 69%. Pada kondisi gerimis, tingkat kelembapan meningkat menjadi 70% hingga 82%, dan saat hujan, kelembapan kembali naik hingga mencapai lebih dari 82%. Data ini menunjukkan bahwa kelembapan udara cenderung meningkat seiring dengan intensitas curah hujan. Selain itu, tingkat kelembapan yang tinggi umumnya terjadi pada awal tahun di sekitar bulan Januari dan Februari.
- Rata-rata jumlah endapan (precipitation) pada kondisi cuaca berawan memiliki nilai 0 yang menunjukkan tidak ada endapan sama sekali. Pada kondisi gerimis,

jumlah endapan meningkat menjadi kisaran 70 - 82, dan saat hujan, tingkat endapan kembali naik hingga mencapai lebih dari 83. Data ini menunjukkan bahwa jumlah endapan cenderung meningkat seiring dengan intensitas curah hujan. Selain itu, jumlah endapan yang tinggi umumnya terjadi pada awal tahun di sekitar Januari dan Februari.

- Durasi siang (daylight duration) tidak menunjukkan adanya pengaruh signifikan dari kondisi cuaca seperti berawan, gerimis, maupun hujan, karena rentangnya tetap berada di kisaran 41.000 hingga 45.000 detik. Umumnya durasi siang yang tinggi ketika pada akhir tahun terutama bulan November dan Desember, kemudian menurun kembali ketika memasuki awal tahun.
- Kecepatan angin pada kondisi cuaca berawan berkisar antara 15 hingga 16 km/jam. Sementara itu, pada kondisi gerimis maupun hujan, kecepatan angin tidak menunjukkan perbedaan yang signifikan, berada dalam rentang 11 hingga 13 km/jam. Namun, jika dilihat secara keseluruhan analisis, tidak ada pola yang konsisten dalam variasi kecepatan angin berdasarkan kondisi cuaca. Kecepatan angin juga tidak memiliki pola tertentu dari segi rentang waktu, terkadang pada bulan tertentu seperti Februari, September, dan Oktober akan meningkat tetapi, tidak menentu.
- Secara umum, analisis menunjukkan bahwa di awal tahun cuaca cenderung didominasi oleh gerimis atau hujan, sementara di akhir tahun lebih sering berawan. Hal ini terbukti dari peningkatan jumlah curah hujan dan tingkat kelembapan di awal tahun, sedangkan suhu udara dan durasi siang hari cenderung meningkat menjelang akhir tahun.
- Secara keseluruhan, kondisi cuaca di kota Surabaya didominasi oleh “light rain”, diikuti oleh “cloudy”, “moderate rain”, dan “heavy rain”. Data menunjukkan bahwa “light rain” paling sering terjadi di hampir semua kecamatan, dengan sekitar 50% dari total kondisi cuaca masuk dalam kategori ini, menjadikannya jenis cuaca yang paling dominan dibandingkan lainnya.

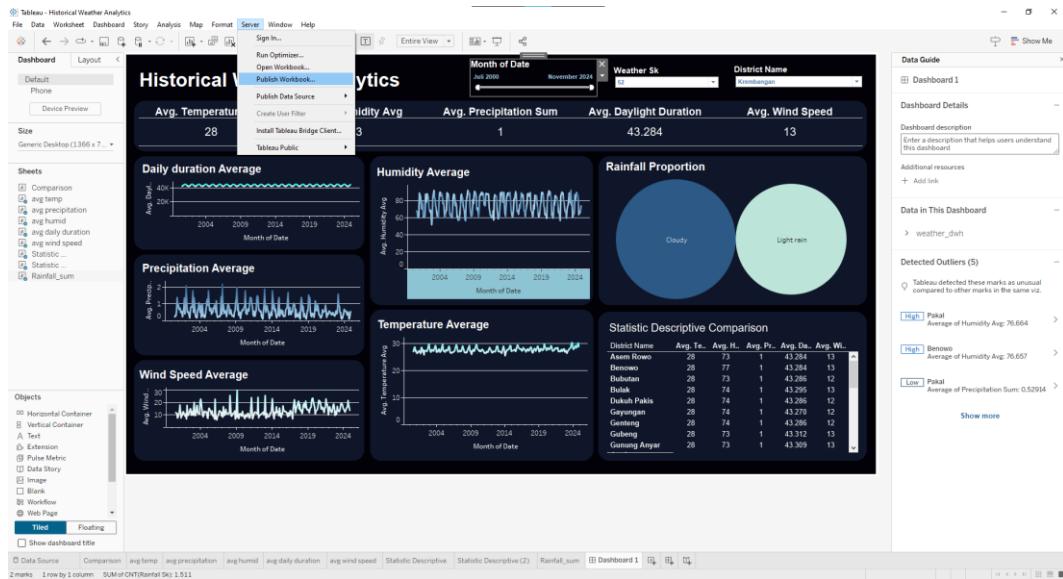
d. Publikasi Dasbor

1. Extract Database

Dashboard public dalam Tableau membutuhkan file berekstensi .hyper yang berfungsi sebagai format penyimpanan data yang digunakan untuk membuat dan mengelola ekstrak data. Dalam studi kasus ini, Data telah dilakukan ekstraksi dan didapatkan file bernama “weather_dwh.hyper”.

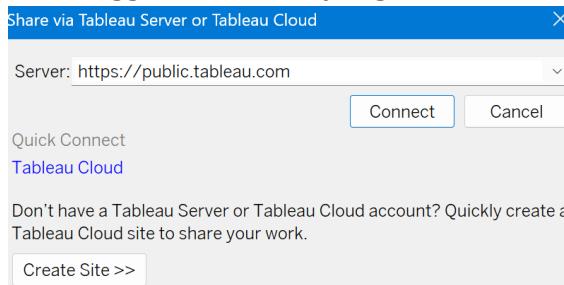
2. Publish Dashboard dalam Tableau Public

- Untuk melakukan publish dashboard dari Tableau Desktop ke Tableau Public. Dapat melalui menu server --> Publish Workbook.



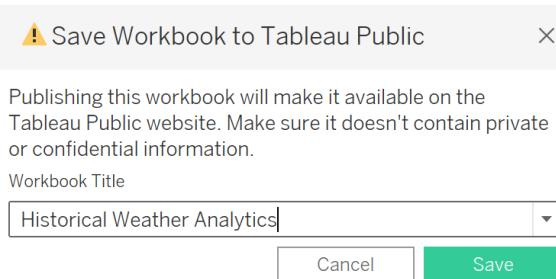
Gambar 4.150. Opsi "Publish Workbook" pada Tableau Dekstop

- Kemudian, akan muncul window baru seperti pada gambar di bawah. Klik tombol "Connect" dan sign-in menggunakan akun yang terdaftar.



Gambar 4.151. Jendela Pemilihan Server dalam opsi "Publish Workbook" pada Tableau Dekstop

- Jika berhasil, maka akan muncul window baru untuk menyimpan Workbook pada Tableau Public. Isi Nama yang diinginkan dan klik "save".



Gambar 4.152. Jendela Inisialisasi Judul Publikasi *Workbook* dalam Tableau Public

- Tunggu beberapa detik, kemudian akan diarahkan secara langsung pada website Tableau Public yang menunjukkan dashboard sudah terpublish. Dashboard dapat diakses melalui tautan berikut:

<https://public.tableau.com/app/profile/angela.lisanthoni/viz/HistoricalWeatherAnalytics/Dashboard1>

BAB V

KESIMPULAN

Penggunaan Pentaho Data Integration (PDI) untuk proses ETL sangat efektif dalam menangani berbagai jenis data. PDI menawarkan keunggulan dalam kemudahan penggunaannya, di mana proses pengaturan alur kerja dapat dilakukan melalui drag-and-drop, serta fleksibilitas dalam menyambungkan ke berbagai sumber data, seperti database MySQL dan file CSV. Hal ini memudahkan dalam melakukan proses penyambungan dan transformasi data sesuai kebutuhan. Selain itu, penggunaan MySQL sebagai basis data sangat cocok untuk data historis cuaca yang digunakan dalam studi kasus ini, terutama karena struktur tabularnya yang mendukung relasi antar tabel dengan mudah. Tableau juga berperan penting dalam visualisasi data, karena kemampuannya dalam menghubungkan berbagai sumber data, termasuk MySQL, dan menyediakan dashboard yang mudah diimplementasikan dan dipublikasikan.

Berdasarkan hasil ETL dan visualisasi data historis cuaca di kecamatan kota Surabaya, beberapa analisis menarik dapat diambil. Suhu udara rata-rata pada kondisi cuaca berawan berada di kisaran 28 hingga 29 derajat Celsius, sedikit lebih tinggi daripada suhu saat gerimis dan hujan yang masing-masing mencapai 28 derajat dan 26 hingga 27 derajat Celsius. Penurunan suhu ini menunjukkan bahwa kondisi hujan berdampak pada penurunan suhu secara konsisten, meskipun perbedaan antar kondisi cuaca tidak terlalu signifikan. Dari segi kelembapan, pada kondisi berawan, kelembapan berada di bawah 70%, tetapi meningkat menjadi 70-82% saat gerimis, dan di atas 82% ketika hujan, menandakan hubungan langsung antara curah hujan dan peningkatan kelembapan. Selain itu, jumlah endapan pada cuaca berawan adalah nol, sementara pada kondisi gerimis dan hujan masing-masing berada pada kisaran 70-82 dan lebih dari 83. Ini memperlihatkan peningkatan signifikan pada intensitas curah hujan yang sejalan dengan jumlah endapan. Durasi siang (daylight duration) tidak menunjukkan pengaruh signifikan dari variasi cuaca, tetapi stabil antara 41.000 hingga 45.000 detik. Di sisi lain, kecepatan angin menunjukkan sedikit variasi, dengan kondisi berawan memiliki kecepatan 15-16 km/jam, sementara saat gerimis dan hujan, kecepatan menurun ke rentang 11-13 km/jam. Namun, pola kecepatan angin yang lebih mencolok terlihat pada akhir tahun, terutama pada bulan September dan Oktober, di mana kecepatan cenderung meningkat, terlepas dari kondisi cuaca harian.

DAFTAR PUSTAKA

- Bouman, R., & Dongen, J. van. (2009). *Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL*. Wiley.
- Casters, M., Bouman, R., & Dongen, J. van. (2010). *Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration*. Wiley.
- Dyer, R. J. T. (2015). *Learning MySQL and MariaDB*. O'Reilly Media.
- Hadi, A. P. (2017). *Penduan Lengkap Query MySQL*.
- Hitachi Vantara. (2024a). *What is new in Pentaho Data Integration and Analytics*.
- Hitachi Vantara, L. (2024b). *Try Pentaho Data Integration and Analytics*.
<https://docs.hitachivantara.com/r/en-us/pentaho-data-integration-and-analytics/9.4.x/mk-95pdia000>
- Maharani, D. (2017). PERANCANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEB PADA SEKOLAH ISLAM MODERN AMANAH. *Jurnal Manajemen Informatika dan Teknik Komputer*, 2(1), 27–32. <https://doi.org/10.31227/osf.io/r9szc>
- Mittal, M., & Raheja, N. G. (2024). *Data Visualization and Storytelling with Tableau*. CRC Press. <https://doi.org/10.1201/9781003429593>
- Oracle. (2024a). *Connectors and APIs Manual*.
<https://dev.mysql.com/doc/connectors/en/>
- Oracle. (2024b). *MySQL Enterprise Edition*.
<https://www.mysql.com/products/enterprise/>
- Pratama, I. P. A. E., & Widhiasih, N. P. N. D. (2020). Perancangan Data Warehouse Untuk Prediksi Penjualan Pada Orba Express Menggunakan Pentaho. *JUSS (Jurnal Sains dan Sistem Informasi)*, 3(2), 43–48. <https://doi.org/10.22437/juss.v3i2.8147>
- Roldán, M. C. (2013). *Pentaho Data Integration Beginner's Guide* (Second Edition). Packt Publishing.
- Saepuloh, D. (2020). Visualisasi Data Covid 19 Provinsi DKI Menggunakan Tableau. *Jurnal Riset Jakarta*, 13(2). <https://doi.org/10.37439/jurnaldrd.v13i2.37>
- Salesforce, Inc. (2024a). *Tableau*. <https://www.tableau.com/products/tableau>
- Salesforce, Inc. (2024b). *Tableau Release Notes*.
https://help.tableau.com/current/tableau/en-us/whatsnew_all.htm
- Solichin, A. (2010). *MySQL5: Dari Pemula Hingga Mahir* (versi 1.0).

LAMPIRAN

Lampiran 1. Video Dokumentasi Proyek

Tautan YouTube Video Dokumentasi Proyek: <https://youtu.be/V-nhFqYtc8I>

Lampiran 2. Berkas (*File*) Proyek (Data, Pentaho Transformation, dan Tableau Workbook)

Tautan Berkas (*File*) Proyek: https://1drv.ms/f/c/e85a400e9475fb16/EjCjk-DWAaBNmxtQ3pfXKiMBnhSl0W_nl9HkmVVO40VWMQ