

**ANALISIS DISTRIBUSI DAYA PEMAKAIAN LISTRIK PADA
AREA PENGATUR DISTRIBUSI (APD) PT. PLN (Persero)
MENGUNAKAN OLAP (*ONLINE ANALYTICAL ROCESSING*)**

**Tesis
Untuk memnuhi sebagian persyaratan
Mencapai derajat Sarjana S-2 Program Studi
Magister Sistem Informasi**



**RIZA SAMSINAR
30000415410026**

**SEKOLAH PASCASARJANA
UNIVERSITAS DIPONEGORO
SEMARANG
2017**

HALAMAN PENGESAHAN

TESIS

ANALISIS DISTRIBUSI DAYA PEMAKAIAN LISTRIK PADA AREA PENGATUR DISTRIBUSI (APD) PT. PLN (Persero) MENGUNAKAN OLAP (ONLINE ANALYTICAL PROCESSING)

Oleh:
Riza Samsinar
30000415410026

Telah diujikan dan dinyatakan lulus ujian tesis pada tanggal 09 Agustus 2017 oleh tim penguji Program Studi Magister Sistem Informasi Sekolah Pascasarjana Universitas Diponegoro.

Semarang, 20 Agustus 2017

Mengetahui,

Penguji I

Prof. Drs. Mustafid, M.Eng, Ph.D
NIP. 195505281980031002

Penguji II

Dr. Suryono, S.Si., M.Si
NIP. 197306301998021001

Pembimbing I

Djatmiko Endro Suseno, M.Si., Ph.D
NIP. 197211211998021001

Pembimbing II

Dr. Catur Edi Widodo, MT
NIP. 196405181992031002

Mengetahui :
Dekan Sekolah Pascasarjana
Universitas Diponegoro



Prof. Dr. Ir. Purwanto, DEA
NIP. 196112281986031004

**Ketua Program Studi
Magister Sistem Informasi**

Dr. Suryono, S.Si., M.Si
NIP. 197306301998021001

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam tesis ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar akademik di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Semarang, Agustus 2017



Riza Samsinar

PERNYATAAN PERSETUJUAN

PUBLIKASI TESIS UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Universitas Diponegoro, saya yang bertanda tangan dibawah ini :

Nama : Riza Samsinar
NIM : 30000415410026
Program Studi : Magister Sistem Informasi
Program : Pascasarjana
Jenis Karya : Tesis

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Diponegoro hak bebas royalti noneksklusif atas karya ilmiah saya yang berjudul :

ANALISIS DISTRIBUSI DAYA PEMAKAIAN LISTRIK PADA AREA PENGATUR DISTRIBUSI (APD) PT. PLN (Persero) MENGUNAKAN OLAP (ONLINE ANALYTICAL PROCESSING)

Beserta perangkat yang ada. Dengan hak bebas royalti noneksklusif ini Magister Sistem Informasi Pascasarjana Universitas Diponegoro berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*) merawat, dan mempublikasikan tesis saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta.

Dibuat di : Semarang

Pada tanggal : Agustus 2017

Yang menyatakan



Riza Samsinar
NIM. 30000415410026

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur kehadirat Allah SWT., atas segala Rahmat, Karunia dan Ridho-Nya, sehingga Tesis dengan judul “**Analisis Distribusi Daya Pemakaian Listrik Pada Area Pengatur Distribusi (Apd) PT. PLN (Persero) Menggunakan OLAP (Online Analytical Processing)**” ini dapat diselesaikan. Tesis ini disusun untuk memenuhi salah satu persyaratan memperoleh gelar Magister Komputer (M.Kom) pada Program Studi Magister Sistem Informasi Universitas Diponegoro.

Pada kesempatan ini penulis menyampaikan terima kasih yang sebesar – besarnya kepada:

1. Prof. Dr. Ir. Purwanto, DEA. selaku Dekan Sekolah Pascasarjana Universitas Diponegoro.
2. Dr. Suryono, S.Si., M.Si. selaku Ketua Program Studi Magister Sistem Informasi Sekolah Pascasarjana Universitas Diponegoro Semarang.
3. Djatmiko Endro Suseno, M.Si, Ph.D selaku Pembimbing I yang telah memberikan waktu, ilmu, saran, semangat dan nasehat yang diberikan selama bimbingan.
4. Dr. Catur Edi Widodo., MT selaku Pembimbing II yang penuh dengan kesabaran memberikan pengarahan dan ilmu yang berguna dalam penulisan tesis ini.
5. Dr. Budiyanto Selaku Dekan Fakultas Teknik Universitas Muhammdiyah Jakarta yang telah memberikan nasehat dan dukungan bagi penulis
6. Segenap Pimpinan dan Staff di PT. PLN (Persero) Disjaya yang telah menyisihkan waktu dan membantu penulis dalam proses pengumpulan dan analisa data penelitian.

Serta pihak-pihak terkait yang telah memberikan kontribusi hingga tesis ini dapat terselesaikan dengan baik. Penulis menyadari masih terdapat kekurangan dalam penyusunan tesis ini, namun demikian penulis berharap tesis ini dapat bermanfaat bagi seluruh pembaca.

Semarang, Agustus 2017
Penulis,

Riza Samsinar

DAFTAR ISI

Halaman

HALAMAN JUDUL.....	i
Halaman Pengesahan	ii
Halaman Persetujuan Publikasi.....	ii
Halaman Pernyataan.....	v
Kata Pengantar	v
Daftar Isi.....	vvi
Daftar Gambar.....	viii
Daftar Tabel	ix
Daftar Lampiran	x
Abstrak	xi
Abstract	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	2
1.3 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	4
2.1 Tinjauan Pustaka	4
2.2 Dasar Teori.....	5
2.2.1 Energi Listrik	5
2.2.2 Sistem Tenaga Listrik.....	6
2.2.3 Keandalan Sistem Tenaga Listrik.....	7
2.2.4 Sistem Distribusi	8
2.2.5 OLAP (<i>Online Analytical Processing</i>)	9
2.2.6 Operasi-operasi OLAP	11
2.2.7 Hubungan <i>Data Warehouse</i> dan OLAP	13
2.2.8 Arsitektur <i>Data Warehouse</i> dan OLAP.....	15
2.2.9 Analisa Multidimensi	16
2.2.10 <i>Reporting</i>	17
2.2.11 Model Pengembangan Perangkat Lunak <i>Waterfall</i>	17
BAB III METODE PENELITIAN	19
3.1 Waktu dan Tempat	19
3.2 Bahan dan Alat Penelitian	19
3.3 Prosedur Penelitian	19
3.4 Kerangka Sistem	23
3.5 Tahapan Pengolahan	23
3.6 Data Flow Diagram.....	29
3.7 Perancangan Basis Data	30
3.8 Perancangan Antarmuka	33

<u>BAB IV HASIL DAN PEMBAHASAN</u>	37
4.1 Hasil Penelitian	37
4.1.1 Pengumpulan Data.....	37
4.1.2 Data <i>Extraction</i>	38
4.1.3 Data <i>Transformation</i>	39
4.1.4 <i>Data Loading</i>	48
4.1.6 Penyimpanan pada <i>Data Warehouse</i>	52
4.1.5 Model Perancangan <i>Start Schema</i> pada OLAP	53
4.2 Pembahasan	54
4.2.1 Halaman Login	54
4.2.2 Halaman Utama Analisa OLAP	55
4.2.3 Hasil Tampilan <i>Start Schema</i>	57
4.2.4 Hasil Analisa Dimensi Area	59
4.2.5 Hasil Analisa Dimensi Gardu Induk	61
4.2.6 Hasil Analisa Dimensi <i>Feeder</i>	64
4.2.7 Hasil Analisa Dimensi Waktu	65
<u>BAB V PENUTUP</u>	68
5.1. Kesimpulan	68
5.2. Saran	68
Daftar Pustaka	70
Lampiran	

DAFTAR GAMBAR

Halaman

Gambar 2.1 Arsitektur <i>data warehouse</i> dengan OLAP	15
Gambar 3.1 Prosedur penelitian	20
Gambar 3.2 Contoh gambar grafik beban <i>feeder</i>	21
Gambar 3.3 Kerangka sistem.....	23
Gambar 3.4 Model <i>star schema</i> untuk <i>data warehouse</i> beban <i>feeder</i> dan trafo gardu induk ditribusi	27
Gambar 3.5 Tahapan pengolahan.....	28
Gambar 3.6 Diagram konteks level 0 sistem analisis beban <i>feeder</i>	29
Gambar 3.7 Tabel data <i>excel</i> total beban <i>feeder</i> dan trafo gardu induk.....	30
Gambar 3.8 Antarmuka login.....	34
Gambar 3.9 Antarmuka halaman utama administrator	34
Gambar 3.10 Antarmuka halaman analisis pengambilan data	35
Gambar 3.11 Antarmuka halaman analisis pemilihan perhitungan	36
Gambar 4.1 <i>Database connection</i> pada <i>data integration</i>	41
Gambar 4.2 Proses data transformasi.....	42
Gambar 4.3 <i>Settings CSV file input</i>	43
Gambar 4.4 Pengaturan <i>dimension lookup/update</i> dimensi area	44
Gambar 4.5 Proses data transformasi waktu	45
Gambar 4.6 Pengaturan pada waktu <i>generate row</i>	46
Gambar 4.7 Pengaturan pada <i>calculator</i> untuk penambahan waktu.....	47
Gambar 4.8 Pengaturan input data tanggal libur.....	48
Gambar 4.9 Pengaturan <i>dimension lookup/update</i> dimensi waktu	49
Gambar 4.10 Proses <i>Data Loading</i>	50
Gambar 4.11 Pengaturan <i>loading</i> proses <i>file csv input</i>	50
Gambar 4.12 Pengaturan <i>loading</i> tabel <i>output</i> fakta <i>feeder</i>	51
Gambar 4.13 Penyimpanan pada <i>data warehouse</i>	52
Gambar 4.14 Perancangan model <i>start schema</i> OLAP.....	53
Gambar 4.15 Halaman login	54
Gambar 4.16 Halaman utama analisa OLAP	55
Gambar 4.17 Tampilan <i>database connection</i>	56
Gambar 4.18 Tampilan <i>schema</i> dan <i>cube</i> dimensi.....	57
Gambar 4.19 Hasil analisa <i>Slice</i>	58
Gambar 4.20 Hasil analisa <i>drill down</i>	58
Gambar 4.21 Hasil Analisa Dimensi Area tampilan <i>Slice</i>	59
Gambar 4.22 Hasil Analisa dimensi area tampilan <i>Drill down</i>	60
Gambar 4.23 Grafik Hasil Analisa Dimensi Area	61
Gambar 4.24 Hasil Analisa Dimensi Gardu Induk	63
Gambar 4.25 Grafik Hasil Analisa Dimensi Gardu Induk.....	63
Gambar 4.26 Hasil analisa dimensi <i>feeder</i>	65
Gambar 4.27 Hasil analisa dimensi waktu	66
Gambar 4.28 Grafik hasil analisa dimensi waktu	66
Gambar 4.29 Hasil analisa dimensi waktu tampilan <i>Drill down</i>	67

DAFTAR TABEL

	Halaman
Tabel 3.1 Struktur tabel db_tanggal.....	31
Tabel 3.2 Struktur tabel db_area	31
Tabel 3.3 Struktur tabel db_penyulang	32
Tabel 3.4 Struktur tabel db_wilayah	32
Tabel 3.5 Struktur tabel db_fakta_beban_penyulang.....	33
Tabel 4.1 Data yang dikumpulkan dan Sumber Data	38
Tabel 4.2 Data <i>Extraction</i>	39
Tabel 4.3 Data <i>Transformation</i>	40

DAFTAR LAMPIRAN

Halaman

Lampiran 1. Surat Penelitian
Lampiran 2. *Execute SQL*

**ANALISIS DISTRIBUSI DAYA PEMAKAIAN LISTRIK PADA
AREA PENGATUR DISTRIBUSI (APD) PT. PLN (Persero)
MENGUNAKAN OLAP (*ONLINE ANALYTICAL PROCESSING*)**

ABSTRAK

Teknik pengolahan *data warehouse* secara online telah digunakan untuk mengelola dan menganalisis kapasitas data yang besar. Metode OLAP untuk mengolah data untuk dapat menganalisa secara online, di proses dengan ETL (*Extraction, Transformation, Loading*), perancang cube dimensi pada *start schema* dan hasil pada *report designer*. Hasil Informasi dari reporting terdiri dari chart distribusi beban berdasarkan pengulangan waktu pada setiap jam setiap hari dan bulan, chart distribusi pada area, chart wilayah gardu induk dan chart pemakaian beban listrik. Hasil proses OLAP menunjukkan perkembangan distribusi beban listrik, serta analisis informasi pada beban pemakaian daya listrik dan menjadi alternatif dalam menyajikan informasi yang berkaitan dengan beban puncak. Analisa dimensi area, Menteng memiliki total beban terbesar dengan presentase 17% dan Serpong memiliki rata-rata terbesar dengan presentase 8%, rata-rata dimensi Area memiliki kisaran presentase sebesar 4% - 6%. Dimensi gardu induk dengan total beban memperoleh presentase dengan kisaran 2% - 4%, total beban terbesar pada gardu induk Duri Kosambi memiliki 4%. Pada rata-rata dimensi gardu induk memiliki kisaran presentase sebesar 2% - 4%. Dimensi waktu dengan total beban dan rata-rata memperoleh presentase dengan kisaran 5% - 6%.

Kata Kunci : *Data Warehouse, Star Schema, OLAP, Distribusi Beban Listrik*

**POWER DISTRIBUTION ANALYSIS FOR ELECTRICAL USAGE IN
AREA DISTRIBUTION SETTINGS PT. PLN (Persero) USING OLAP
(ONLINE ANALYTICAL PROCESSING)**

ABSTRACT

The technique of data warehousing online analytical processing has been used to manage and analysis the great capacity of data. OLAP method for processing data to analyze online, in process with ETL (Extraction, Transformation, Loading), cube designer dimension at start schema and result of report designer. The information in the form of graph is the distribution graph of the repeater load based on the repeating time per hour per month and per month, the total load distribution graph based on the location of the area and the main distribution and distribution of the repeater load based on the load of repetitive power and tarfo power. The results of the OLAP process show the great development of electric power loads on the related parts. Large power information analysis system at the load of repeatability can be an alternative in presenting information related to peak load. Area dimension analysis, Menteng Area has the largest total load with 17% presentase and Serpong has the largest average with 8% percentage, Area dimension average has percentage range of 4% - 6%. The dimension of the main substation with the total load obtained percentage with the range of 2% - 4%, the largest total load on the Duri Kosambi substation has 4%. On the average dimension of the substation has a percentage range of 2% - 4%. Time dimension with total load and average gain percentage with range 5% - 6%.

Key Words : Data Warehouse, OLAP, star Schema, power load Analysis.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan listrik dewasa ini semakin meningkat, sebanding dengan peningkatannya permintaan listrik secara terus menerus yang dapat mengakibatkan sumber pembangkit listrik menipis (Prakash dkk., 2016). Kebutuhan energi listrik begitu besar sehingga kecukupan suplai energi listrik menjadi faktor utama pendukung keberhasilan setiap kegiatan, sebaliknya kehabisan energi listrik menjadi faktor penghambat kegiatan seperti contoh pada bidang industri, penghambatan ini mengakibatkan kegagalan dan kerugian yang sangat besar.

Penginputan data suplai energi listrik PT. PLN (Persero) pada setiap gardu induk untuk distribusi feeder energi listrik setiap hari setiap jam mengakibatkan data yang besar. proses penginputan untuk melihat grafik pada setiap jam nya, untuk melihat dijam berapa beban puncak itu berada pada gardu induk, feeder dan trafo. Data warehouse (gudang data) salah satu wadah untuk Pengumpulan sejumlah data dalam kebijakan pengambilan keputusan, yang berfungsi untuk penyimpanan, analisis dan visualisasi data secara efektif. Proses Extract transform load (ETL) yang berfungsi sebagai pengumpulan data dari beberapa sumber, tidak dapat diperbaharui pada data warehouse secara optimal, sehingga ETL harus didistribusikan dalam data warehouse untuk pendekatan pada data arsip pelaporan dan visualisasi (Boulekrouche dkk., 2015).

Data cube yang dihasilkan oleh data warehouse diproses menggunakan analisis statistika dalam proses data mining. Salah satu metode yang digunakan untuk melakukan analisis adalah OLAP (OnLine Analytical Processing). Hasil penyederhanaan dan peringkasan ini disajikan kepada pengguna yang merupakan dasar pengambilan keputusan bisnis. Dengan demikian manajemen dapat mengambil keputusan berdasarkan fakta-fakta aktual dan tidak hanya mengandalkan intuisi dan pengalaman kuantitatif saja (Hendric, 2006)

Data warehouse merupakan kumpulan data yang bersifat subject-oriented, terintegrasi, time varian, nonvolatile yang membantu manajemen suatu perusahaan

dalam proses pembuatan keputusan. Data warehouse menyediakan suatu tools yang disebut OLAP untuk melakukan analisis data multidimensional secara interaktif yang nantinya akan menjadi fasilitas yang memudahkan untuk melakukan proses data mining (Reddy dkk., 2010).

Penggunaan sistem selain Data warehouse dan OLAP sebagai tools untuk menganalisis dan mengumpulkan data yang ada menjadi informasi yang sangat penting. Penerapan OLAP ini menunjukkan keunggulan pada tingkat validasi informasi, sehingga informasi yang dihasilkan dapat dipercaya. Selain itu, dengan sistem OLAP dapat juga menganalisis data dari beberapa pakar (Wang dkk., 2013).

Permasalahan pada proses ini data hanya di transformasi pada aplikasi penginputan data. Data di ekstraksi ke excel setiap jam, data di kumpulkan pada satu sheet dalam sebulan, yang kemudian di proses dan di analisa untuk melihat grafik setiap bulannya. Dengan adanya permasalahan yang sudah penulis uraikan sebelumnya, maka perlu adanya sistem untuk menganalisis pemakaian daya energi listrik dengan menggunakan OLAP. Sehingga dapat memberikan hasil grafik dari pemakaian daya, kurva beban daya sehingga terlihat puncak beban, menganalisa untuk penghematan, dan dapat memberikan informasi sebagai dasar untuk pengambilan keputusan.

1.2 Tujuan Penelitian

Tujuan dari penelitian adalah merancang Sistem Informasi analisis distribusi daya pemakaian listrik dengan menggunakan metode OLAP pada area pengatur distribusi (APD) PT. PLN (Persero) yang dijadikan sebuah informasi dan sebagai dasar pengambilan keputusan serta rekomendasi bagi pihak yang terkait. Adapun perancangan sistem analisis OLAP dirancang sampai pada pembuatan laporan, kurva berupa grafik dan tabel nilai yang dapat digunakan sebagai dasar pengambilan keputusan.

1.3 Manfaat penelitian

Manfaat penelitian ini adalah :

1. Dapat *monitoring* grafik, capai kurva beban dalam penggunaan distribusi beban untuk daya listrik pada sekitar area Jakarta Raya.
2. Sistem rancang dengan metode OLAP dapat dikembangkan dengan penambahan metode lain untuk dapat memberikan memprediksi besar beban *feeder* yang dibutuhkan untuk jangka pendek, jangka menengah, dan jangka panjang.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Keandalan sistem tenaga listrik pada transmisi dan distribusi tenaga listrik memberikan gambaran probabilitas pada terputusnya jaringan listrik (mati listrik), sedangkan frekuensi terputusnya jaringan listrik dan lamanya terputusnya daya listrik disebabkan karena gangguan jaringan transmisi dan distribusi. Penggunaan teknologi sistem informasi dapat dilakukan untuk menganalisis setiap aspek dari data dalam memahami dan mengidentifikasi operasi bisnis, memonitoring, menganalisis, dan meningkatkan proses bisnis dengan tujuan mendapatkan keuntungan yang kompetitif di pasar (Meliopoulos dkk., 2005).

Penggunaan teknologi sistem informasi dapat dilakukan untuk menganalisis setiap aspek dari data dalam memahami dan mengidentifikasi operasi bisnis, memonitoring, menganalisis, dan meningkatkan proses bisnis dengan tujuan untuk mendapatkan keuntungan yang kompetitif di pasar (Curko dan Varga., 2008). Kebutuhan untuk analisis data multi dimensi sebagai dukungan untuk pengambilan keputusan bisnis yang telah hadir selama beberapa dekade terakhir ini. Teknologi OLTP (*OnLine Transaction Processing*) diterima dengan baik pada pasar pengguna sistem informasi akan tetapi tidak dirancang untuk tugas-tugas dalam menangani BI dan oleh karena itu, teknologi OLAP dikembangkan sebagai solusinya. Penggunaan teknologi OLAP telah menggantikan teknologi dari OLTP adalah dapat mengimplementasikan berupa pelaporan analisis kubus (*Cube Reporting Analysis*) pada data yang diambil secara langsung dalam relasional sebuah perangkat basis data tanpa modifikasinya terlebih dahulu (Westerlund, 2008).

Data warehouse dapat digunakan untuk mendukung OLAP yang secara fungsional dan kinerja tidak jauh berbeda dengan OLTP (*Online Transaction Processing*) yang secara umum didukung oleh *database* operasional. *Data warehouse* menyediakan OLAP sebagai alat yang bersifat multidimensional sehingga dapat memberikan kemudahan untuk melakukan penambangan data (*data mining*). *Data warehouse* dan OLAP merupakan komponen utama dan penting

dalam pendukung keputusan, yang fungsinya juga meningkatkan menjadi *database* utama dalam industri (Reddy dkk., 2010).

Sistem untuk membantu menghasilkan dukungan keputusan dengan menganalisa informasi pada *data warehouse*. Informasi dukungan keputusan berupa visualisasi dalam grafik vektor untuk membantu interpretasi sebuah informasi robotik. Sistem ini menggunakan OLAP pada proses pengolahan dan analisa informasi pada data warehouse. Aplikasi OLAP menghasilkan sebuah sistem pendukung keputusan yang interaktif dan informatif. Penyajian berupa web *server* dapat dengan cepat mengolah informasi dan ditampilkan pada *browser*. Pengolahan dan analisa dengan sistem ini menghasilkan informasi yang *relevan* dengan kesalahan relatif kecil (Pradhana dkk., 2013).

Persediaan air yang terbatas untuk pengolahan suatu pengukuran irigasi pertanian dan industri di Taiwan, sehingga Sistem web 2.0 dan OLAP digunakan untuk menganalisis data kebutuhan air dari didistribusikan sumber daya air untuk memberikan pemahaman informasi untuk meningkatkan kualitas air dari sistem komputerisasi. Dan dengan sistem ini bagi pengguna bisa dimanfaatkan untuk berkolaborasi dalam memantau kualitas air dan melindungi keterbatasan sumber daya air yang ada (Wang dkk., 2013).

2.2 Dasar Teori

Dalam rangka mendukung penelitian ini maka dibutuhkan beberapa landasan teori sesuai dengan keilmuan yang digunakan dalam penelitian ini.

2.2.1 Energi Listrik

Energi merupakan kemampuan untuk melakukan usaha, dan energi memiliki usaha yang tersimpan. Energi hanya dapat diubah dari suatu bentuk ke bentuk energi yang lain. Demikianlah pula energi listrik merupakan hasil perubahan energi mekanik (gerak) menjadi energi listrik. Energi listrik merupakan energi utama yang dibutuhkan bagi peralatan listrik/energi yang tersimpan dalam arus listrik dengan satuan amper (A) dan tegangan listrik dengan satuan volt (V) dengan ketentuan kebutuhan konsumsi daya listrik dengan satuan Watt (W). Keberadaan energi listrik ini dapat dimanfaatkan semaksimal mungkin. Adapun kegunaan

energi listrik dalam kehidupan sehari-hari merupakan penerangan, pemanas, motor-motor listrik dan lain-lain untuk menghasilkan bentuk energi yang lain. Energi yang digunakan alat listrik merupakan laju penggunaan energi (daya) dikalikan dengan waktu selama alat tersebut digunakan. Bila daya diukur dalam watt jam, maka (Maursudi, 2006):

$$W = p \times t \quad (2.1)$$

Dengan:

- p = daya dalam watt (W)
- t = waktu dalam jam (h)
- W = energi dalam watt jam (Wh)

2.2.2 Sistem Tenaga Listrik

Penyediaan tenaga listrik bagi para pelanggan, diperlukan berbagai peralatan listrik. Berbagai peralatan listrik ini dihubungkan satusama lain yang mempunyai interrelasi dan secara keseluruhan membentuk suatu sistem tenaga listrik. Adapun dimaksud dengan sistem tenaga listrik di sini adalah sekumpulan pusat listrik dan gardu induk (pusat beban) yang satu sama lain dihubungkan oleh jaringan transmisi sehingga merupakan satu kesatuan interkoneksi. Kebutuhan akan tenaga listrik dari pelanggan selalu bertambah dari waktu ke waktu. Untuk tetap dapat melayani kebutuhan tenaga listrik dari para pelanggan, maka sistem tenaga listrik harus dikembangkan seiring dengan kenaikan kebutuhan akan tenaga listrik dari para pelanggan. Untuk dapat melakukan hal ini dengan sebaik-baiknya maka hasil-hasil operasi perlu dianalisa dan dievaluasi antara lain untuk menentukan Bilamana, beberapa besar dan dimana perlu dibangun pusat listrik baru, gardu induk baru serta saluran transmisi baru. Dapat menambah unit pembangkit dan menambah transformator dan lain-lain. Bilamana dan dimana saja perlu penggantian PMT dengan yang lebih besar sebagai mana konsekuensi dari dua permasalahan sebelumnya (Marsudi, 2006).

2.2.3 Keandalan Sistem Tenaga Listrik

Keandalan adalah probabilitas dari perangkat atau sistem untuk melakukan fungsinya secara memadai, dalam periode waktu tertentu, di bawah kondisi operasi yang tertentu pula. Disini data pengamatan masa lalu berguna untuk membuat perkiraan kinerja di masa depan (Prada, 1999). Keandalan sistem tenaga listrik memiliki ukuran kuantitas dan kualitas. Ukuran kuantitas berupa jumlah cadangan kapasitas daya listrik di dalam sistem dan ukuran kualitas berupa nilai kinerja unit pembangkit listrik. kuantitas yang besar dari cadangan daya tidak akan berarti bila kualitas pembangkitnya sangat rendah. Aspek kuantitas dan kualitas secara bersama digunakan untuk perhitungan LOLP (*Loss of Load Probability*) sebagai indeks keandalan sistem tenaga listrik (Marsudi, 2006).

Dalam laporan tesis ini yang dimaksud dengan sistem adalah sistem tenaga listrik, kapasitas daya adalah kapasitas daya listrik, cadangan daya adalah cadangan daya listrik, unit adalah unit pembangkit tenaga listrik dan beban adalah beban tenaga listrik. Perhitungan cadangan daya diperoleh dari jumlah kapasitas daya semua unit yang beroperasi dikurangi dengan jumlah daya kebutuhan beban. Keandalan sistem secara kuantitas semakin baik bila cadangan daya yang tersedia semakin besar.

Ukuran kualitas berupa nilai kinerja unit pembangkit didapat dari angka gangguan unit. Kualitas unit berbanding terbalik dengan angka gangguannya. Semakin rendah angka gangguan unit ini dalam istilah asing disebut dengan *forced outage rate* (FOR), selanjutnya dalam laporan ini ditulis sebagai FOR dalam bentuk persen atau angka decimal. Misalnya suatu unit memiliki kapasitas daya 10 MW dan FOR 0,01 berarti probabilitas terjadinya gangguan di masa mendatang pada unit tersebut sebesar 0,01 atau 1%. Hal ini berarti pula probabilitas terjadinya kehilangan kapasitas daya sebesar 100 MW juga sebesar 0,01 atau 1%. Dalam penelitian ini digunakan nilai FOR dalam bentuk decimal.

Optimasi keandalan sistem diperlukan Karena dari sisi kuantitas diperlukan unit masuk dalam sistem dan beroperasi sebanyak-banyaknya, namun dari sisi kualitas diperlukan pemeliharaan pada setiap unit untuk mempertahankan kinerjanya sehingga unit harus keluar dari sistem. Dalam sistem interkoneksi,

beberapa unit pembangkit terhubung dalam satu sistem sehingga untuk probabilitas kehilangan sejumlah kapasitas daya dihitung dari total probabilitas gangguan pada unit-unit yang berkaitan dengan jumlah kapasitas daya tersebut.

2.2.4 Sistem Distribusi

Sistem distribusi merupakan salah satu sistem dalam tenaga listrik yang mempunyai peran penting karena berhubungan langsung dengan pemakai energi listrik, terutama pemakai energi listrik tegangan menengah dan tegangan rendah. Biasanya sering kali terjadi beban tidak seimbang pada fase-fasenya (sistem distribusi merupakan sistem 3 fase) atau terjadi kelebihan beban karena pemakaian alat-alat listrik dari konsumen energi listrik. Keseimbangan beban antar fasa diperlukan untuk pemerataan beban sehingga meminimalkan perubahan yang diakibatkan oleh beban penuh. Hal ini juga penting karena bermanfaat pada teknik optimasi untuk menghasilkan sistem yang handal dan efisien. Sebuah konfigurasi 1 fase dengan 3 kabel dapat dikatakan tidak seimbang jika arus netral tidak bernilai nol. Hal ini terjadi karena beban yang dikoneksikan, antara fase dan neutral tidak sama (Prada, 1999).

1. Jaringan Distribusi

Jaringan distribusi adalah semua bagian dari suatu sistem yang menunjang pendistribusian tenaga listrik yang berasal dari gardu-gardu induk. Klasifikasi jaringan distribusi menurut strukturnya antara lain struktur jaringan radial, struktur jaringan *loop*, dan struktur jaringan *spindle*.

2. Aliran Daya

Studi aliran daya adalah penentuan atau perhitungan tegangan, arus dan daya yang terdapat pada berbagai titik suatu jaringan pada keadaan pengoperasian normal, baik yang sedang berjalan maupun yang akan datang

3. Jatuh Tegangan (*Drop Voltage*)

Jatuh tegangan adalah selisih antara tegangan ujung pengiriman dan tegangan ujung penerimaan, jatuh tegangan disebabkan oleh hambatan dan arus, pada saluran bolak-balik besarnya tergantung dari impedansi dan admitansi saluran

serta pada beban dan faktor daya. Jatuh tegangan relatif dinamakan regulasi tegangan dan dinyatakan dengan rumus:

$$V_{reg} = \frac{V_s - V_r}{V_s} \times 100\% \quad (2.2)$$

Dengan :

V_{reg} = regulasi tegangan

V_s = tegangan pada pangkal pengiriman

V_r = tegangan pada ujung penerimaan

2.2.5 OLAP (*Online Analytical Processing*)

OLAP merupakan operasi basis data untuk mendapatkan dalam bentuk kesimpulan dengan menggunakan agregasi sebagai mekanisme utama. Mekanisme berupa analisis dan pengambilan keputusan. Kelebihan OLAP adalah kemampuannya untuk menciptakan struktur bisnis (wilayah penjualan, kategori produk, kalender fiskal, saluran partner, dan sebagainya) dan menggabungkannya dengan cara tertentu sehingga pengguna dapat dengan cepat menjawab berbagai pertanyaan bisnis. OLAP dapat menganalisis lebih dari yang disediakan laporan biasa pada umumnya dan memungkinkan OLAP untuk mengembangkan apa yang disediakan oleh data, mencari tren dan anomali (Bukhbinder dkk., 2004).

Tanpa adanya tampilan multidimensi yang tersedia pada laporan, user harus mengambil informasi dari berbagai tabel atau sumber lain dan melakukan analisis yang berorientasi obyek seperti pada spreadsheet elektronik, atau melakukan query database yang kompleks. Respon yang cepat dan kemampuan non-profesional untuk melakukan analisis tanpa keterlibatan programmer adalah keuntungan dan kelebihan dari OLAP. Pada dasarnya dengan adanya OLAP laporan-laporan spreadsheet standar menjadi ketinggalan jaman (Bukhbinder dkk., 2004).

Dengan menggunakan teknologi OLAP, user dapat menganalisis data secara interaktif dengan menggunakan fasilitas yang baik untuk membuat laporan, *user* diijinkan untuk merotasi *grid* laporan, menelusuri data yang meringkasnya, melakukan *filter* dan melakukan *sorting* terhadap data dan menghasilkan beberapa

view/bentuk laporan hanya dengan manipulasi *mouse*. Dengan demikian dapat diciptakan berbagai laporan yang kompleks dari satu tabel tanpa memerlukan pengetahuan ekstra tentang pembuatan *query* dan bantuan seorang *programmer*. Dengan pengujian data dari sudut yang berbeda, *user* akan dapat lebih memahami data sehingga dapat mengambil keputusan yang efektif (Hermawan, 2005).

Aplikasi kebanyakan melibatkan sistem pendukung keputusan. Seperti bisnis intelijen, manajemen kinerja bisnis, pelaporan keuangan, penemuan pengetahuan, data mining dan bidang terkait lainnya. Keuntungan dari sistem OLAP adalah penggunaan *end-user* menggunakan metode multidimensi untuk analisis data *ad hoc* dan efisien dalam penyajian informasi yang dapat dipercaya. Berdasarkan fitur ini, OLAP berguna dalam penelitian ilmiah lainnya dan merancang sistem yang membutuhkan analisis data multi-dimensi. Selain itu OLAP juga dapat sebagai implementasi untuk membuat alat data mining lebih efektif dalam mengekspresikan suatu basis data dan sistem informasi geografis yang terintegrasi dan OLAP dapat memfasilitasi penemuan pengetahuan geografis dan menganalisis data kesehatan lingkungan sekitar (Bourman dkk., 2009). OLAP juga dapat dipergunakan untuk menganalisis kejadian kutu putih pada tanaman kapas. OLAP juga berlaku untuk masalah lingkungan, seperti aplikasi OLAP di daerah ekologi sangat diperlukan karena data biasanya dianalisis dalam beberapa dimensi (Sings, 2011).

Pada proses OLAP, isi data akan dikelompokkan ke dalam kelompok yang berbeda, dan masing-masing pengelompokan akan diberi kode metadata. Untuk mengintegrasikan masing-masing sumber data, kode metadata telah ditugaskan untuk menandai setiap sumber daya *variable* yang ada, misalnya PL adalah kode untuk penomoran area. Setelah kode sumber daya telah ditetapkan, titik sampling juga akan diberikan kode metadata spesifik. Database metadata ini merupakan *extensible* dimana setiap item yang tersimpan di gudang data akan terdaftar di *layer*. Pengguna dapat memilih permintaan mereka mengirimnya ke server OLAP. *Server* akan menggunakan basis pengetahuan untuk membangun pohon keputusan dan menganalisis tingkat permintaan yang diinginkan (Wang dkk., 2013).

Metode dengan sistem OLAP bagian salah satu dari sistem *Bussiness Intelligence* (BI). Arsitektur yang di bangun pada BI menurut Crowsoll, 2012 terdiri dari empat bagian yaitu :

1. *Source Systems*

Sebagian besar perusahaan telah membagikan data-data yang berkaitan dengan departemen yang bersangkutan. Penjualan departemen biasanya memiliki metrik yang berhubungan dengan penjualan dan kinerja. Sumber-sumber data yang berbeda membentuk *landscape* BI bagi perusahaan.

2. *Data Staging Area*

Sumber data yang berbeda yang terdiri dari informasi bisnis perusahaan akhirnya disimpan dalam gudang data. Ada sebuah proses yang disebut *Extract Transform dan Load* (ETL) yang memfasilitasi sarana format data dari sumber yang berbeda sehingga data dapat dimuat ke dalam struktur *data warehouse*.

3. *Data Warehouse*

Bentuk-bentuk berbagai data bisa disimpan dalam sebuah *Data Warehouse* yang merupakan struktur yang memfasilitasi pekerjaan utama intelijen bisnis - pelaporan dan analisis. Bit data yang terstruktur dengan cara memfasilitasi matriks fleksibel yang pada akhirnya dapat dikonfigurasi untuk menampilkan data dalam berbagai dimensi dengan memanfaatkan konsep OLAP. Konsep ini menyediakan struktur dikonfigurasi untuk menganalisis data.

4. *User Interface*

Analisis BI dapat terwujud dalam beberapa cara berbeda. Bisa diwujudkan dengan aplikasi berbasis Web ataupun aplikasi *custom* yang menampilkan informasi.

2.2.6 Operasi-operasi OLAP

Slicing dan *dicing* merupakan operasi untuk melihat data sebagai visualisasi dari kubus. Dengan *slicing* dan *dicing* pengguna dapat melihat data dari beberapa perspektif atau dimensi. Pengguna dapat mengekstrak bagian dari data agregated dan dapat memeriksa dengan detail berdasarkan dimensi-dimensi yang diinginkan. Data *Agregated* merupakan data praperhitungan (*precalculated*) dalam bentuk

rangkuman data (*data summarized*) sehingga *query* pada kubus (*cube*) lebih cepat. *Slicing* memotong kubus sehingga dapat memfokuskan pada perspektif yang spesifik (pada suatu dimensi). Sedangkan *dicing* memberikan kemampuan untuk melihat pemilihan data pada dua dimensi atau lebih. Yaitu dengan merotasi *cube* pada perspektif yang lain sehingga pengguna dapat melihat lebih spesifik terhadap data yang dianalisis (Thorlund dkk., 2010). Konsep *slicing* dan *dicing* berfungsi untuk mengambil potongan *cubes* berdasarkan nilai tertentu pada satu atau beberapa dimensinya (Hendric, 2006).

Roll up dan *drill down* Merupakan operasi untuk melihat data global atau detail disepanjang level hirarki dimensi. *Roll up* untuk melihat data secara global atau rangkuman (*summary*). *Drill down* memandu pengguna untuk memperoleh data yang lebih detail (Kamber, 2006). *Roll up* adalah proses merangkum atau meringkas nilai-nilai ukurannya, sedangkan proses *drill down* adalah kebalikannya yaitu proses memilih dan menampilkan data rincian dalam satu atau beberapa dimensi (Hendric., 2006).

Ada dua cara untuk merealisasikan OLAP (Hermawan., 2005) yaitu :

1. Cara pertama, dengan mengimplementasikan sebuah OLAP *server* dimana perhitungan eksekusi dilakukan pada komputer yang terpisah. OLAP *server* memerlukan investasi tambahan dan pemeliharaan permanen karena sama sekali tidak memerlukan bantuan user untuk melakukan kalkulasi. Dengan cara demikian OLAP *server* dimungkinkan untuk menerima data dengan *volume* yang sangat besar.
2. Cara kedua adalah dengan mengimplementasikan sebuah OLAP *client* yang akan melakukan perhitungan di mesin user, OLAP *client* tergolong murah dan tidak membutuhkan *maintenance*. OLAP *client* dapat dipergunakan bilamana kebutuhan untuk melakukan manipulasi data relatif kecil.

2.2.7 Hubungan *Data Warehouse* dan OLAP

Kemampuan kita mengumpulkan dan menyimpan segala jenis data melampaui kemampuan kita melakukan analisis, peringkasan, dan ekstraksi pengetahuan dari data. Untuk membentuk analisis data diperlukan kaskas otomatis yang dapat membantu melakukan ekstraksi dan penemuan pengetahuan dari data. Saat ini orang sudah sepakat bahwa informasi berkualitas tinggi adalah sangat penting dalam dunia bisnis. OLAP menggunakan informasi basisdata untuk menyusun keputusan strategis. Basis data yang terlibat biasanya sangat besar dan seringkali tidak memerlukan data terbaru. Aplikasi OLAP dicirikan dengan *query* yang kompleks, pembaruan tidak sering dan mengakses sebagian besar basis data (Hermawan., 2005). Tujuan OLAP menganalisis data adalah untuk digunakan di suatu pengambilan keputusan taktis dan strategis. Terdapat dua isu teknis yang utama dalam analisis data, yaitu :

1. Prosedur analisis yang dilakukan dan data yang mendukung prosedur itu.
2. Metode-metode untuk memperoleh bagian besar data yang diperlukan secara efisien.

Basis data OLAP biasanya disimpan di OLAP Server khusus atau di data warehouse yang distrukturkan untuk mendukung OLAP Server. OLAP *query* sering begitu kompleks, memerlukan data yang sangat besar, yang bila dijalankan sekaligus di lingkungan OLTP akan dapat menyebabkan melambatnya transaksi OLTP secara drastis. *Data warehouse* adalah *repository* (arsip) informasi yang dikumpulkan dari banyak sumber, disimpan dengan skema yang disatukan di satu situs tunggal. Begitu dikumpulkan, data disimpan dalam kurun waktu yang lama. *Data warehouse* menyediakan satu antarmuka terkonsolidasi tunggal sehingga mempermudah pembuatan *query* yang mendukung pembuatan keputusan. Dengan mengakses informasi dari data warehouse, pembuat keputusan dapat menjamin bahwa Sistem pengolahan transaksi *online* tidak akan terganggu.

Data warehouse merupakan basis data dimana data dikumpulkan dari banyak Sistem untuk mendukung pelaporan dan pengambilan keputusan manajemen. Basis data *multidimensi* yang merupakan basisdata Sistem OLAP *multidimensi* memberi solusi yang berorientasi bisnis untuk menjawab pertanyaan

yang kompleks. Pendekatan ini mempunyai tingkat keberhasilan yang tinggi ketika jawaban disusun dari matriks atau data kuantitatif.

Skema tabel yang sering digunakan adalah skema bintang. Skema bintang merupakan skema yang digunakan OLAP, skema tabel ini membentuk struktur informasi multidimensi yang kompatibel dengan kebutuhan bisnis. Skema bintang untuk menyelesaikan isu-isu di seputar model ER yang dinormalkan untuk kebutuhan basisdata informasi bisnis. Karakteristik utama skema bintang adalah sebagai berikut :

1. Pusat skema bintang adalah tabel fakta (*fact table*).
2. Tabel fakta berisi indikator-indikator kinerja pokok.
3. Indikator-indikator kinerja pokok adalah atribut-atribut dari tabel fakta.
4. Obyek-obyek informasi dan waktu adalah kunci utama tabel fakta.
5. Tabel-tabel yang ada disekeliling tabel fakta adalah tabel dimensi.
6. Tabel dimensi berisi data mengenai obyek-obyek informasi atau waktu.
7. Tabel fakta dan dimensi digabungkan dengan kunci banyak di tabel fakta.
8. Skema bintang diimplementasikan menggunakan teknologi basisdata relasional.

Jika tabel-tabel dimensi dinormalisasi (sehingga dapat menjadi beberapa tabel), skema menjadi lebih kompleks, disebut bercak salju (*snowflake schema*). Namun terdapat alasan tabel dimensi jarang dinormalisasi, yaitu :

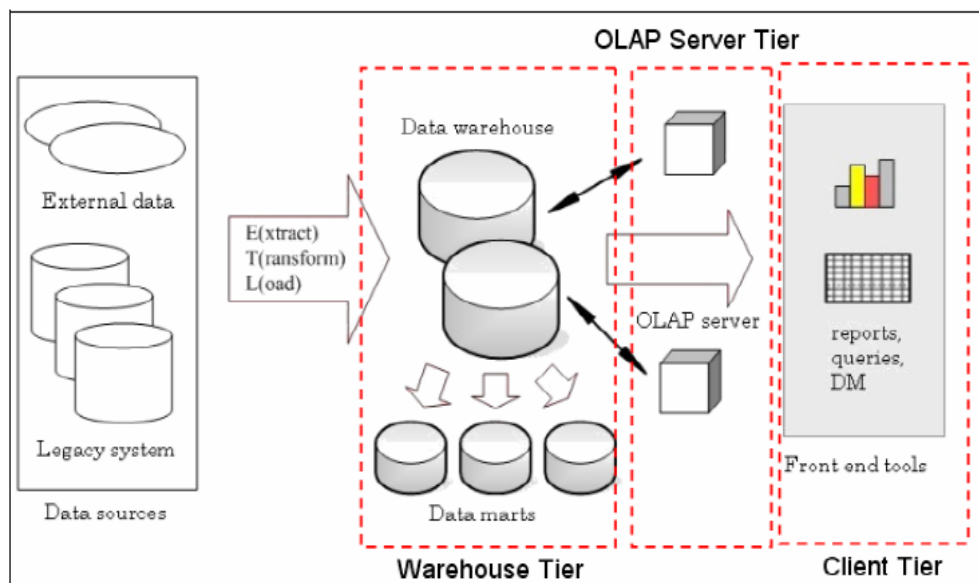
1. Tabel dimensi telah begitu kecil dibandingkan tabel fakta sehingga ruang yang dihemat melalui normalisasi tidak signifikan.
2. Tabel dimensi jarang diperbaharui sehingga *anomaly* pembaruan tidak merupakan isu utama. Bahkan dalam situasi eksekusi, analisis mendekomposisi relasi-relasi menjadi 3NF atau BCNF dapat menuntun overhead untuk pengolahan query secara signifikan.

2.2.8 Arsitektur *Data Warehouse* dan OLAP

OLAP dalam fungsinya menganalisa *data warehouse* memiliki kemampuan mengeksplorasi data multi dimensi dan menampilkannya secara lengkap dan terstruktur (Bimonte dan Tchounikine, 2007). Data disusun kedalam kubus yang merepresentasikan komponen-komponen informasi yang hendak dianalisa. Informasi pada OLAP disusun dari data-data yang ada pada *data warehouse* dengan berbagai metode *query*. Secara umum *data warehouse* disusun dalam tiga lapis diantaranya adalah sebagai berikut :

- a. *Warehouse tier*, pada lapisan ini seluruh data dikumpulkan dan di proses dengan berbagai operasi untuk menghasilkan informasi yang seragam.
- b. *OLAP tier*, pada lapisan ini komponen-komponen informasi disiapkan untuk menyusun kubus informasi yang siap dianalisa.
- c. *Client tier*, lapisan ketiga merupakan sisi dari sistem yang berhubungan langsung dengan pengguna. Pada lapisan ini informasi divisualisasikan agar mudah dibaca dan dipahami pengguna.

Ilustrasi arsitektur *data warehouse* dan OLAP ditampilkan pada gambar 2.1.



Gambar 2.1 Arsitektur *data warehouse* dengan OLAP (Bimonte dan Tchounikine, 2007)

Untuk kebutuhan pengambilan keputusan masih diperlukan lagi alat bantu yang menjembatani antara pengguna dengan *data warehouse*. Salah satu alat bantu dalam pengambilan keputusan adalah OLAP. OLAP pertama kali diusulkan oleh E.F Codd yang merepresentasikan informasi ke dalam larik multidimensi (Hammergren dan Simon, 2009). Operasi OLAP difungsikan untuk memandangi informasi dari berbagai sudut pandang berbeda (Hammergren dan Simon, 2009). Beberapa operasi yang didukung oleh OLAP multi dimensi diantaranya sebagai berikut.

- a. Konsolidasi dengan mengelompokkan beberapa kategori sesuai hirarkinya untuk memperoleh informasi yang lebih global.
- b. Perincian merupakan kebalikan dari konsolidasi, operasi ini memperinci satu kategori pada dimensi menjadi anggota-anggota yang lebih kecil lingkungannya.
- c. Pencacahan merupakan operasi membandingkan fakta-fakta yang ada dalam satu lingkup dimensi saja. Sering kali diperlukan untuk mencari hubungan antar dimensi yang dianalisa.

2.2.9 Analisa Multidimensi

Data perlu diorganisasi dalam bentuk lain berupa data multidimensi yang dinamakan MOLAP (*Multidimensional Online Analytical Processing*) atau data relasional ROLAP (*Relational Online Analytical Processing*) atau HOLAP (*Hybrid Online Analytical Processing*) yang merupakan kombinasi dari ROLAP dan MOLAP dimana sebagian data dapat disimpan dalam MOLAP dan sebagian yang lain dalam ROLAP. Data disimpan dalam *data warehouse* dalam bentuk multidimensi dioptimasi untuk pencarian kembali (*retrieval*) untuk OLAP. Setelah itu dilakukan analisis multidimensi yang memberikan kemampuan untuk melakukan *query* dan membuat laporan (*reporting*) (Thorlund dkk., 2010). Suatu cara melihat data dengan multidimensi tersebut dikenal dengan nama kubus (*cube*). Kubus ini menjadi struktur OLAP yang utama yang digunakan untuk melihat data. Analisis menggunakan kubus ini memberikan fasilitas banyak dimensi untuk melihat data yang diinginkan. Sehingga memungkinkan untuk mengakses data

dengan lebih mudah dan cepat untuk menjawab pertanyaan yang dikemukakan (Thorlund dkk., 2010).

2.2.10 Reporting

Reporting adalah berita publik yang berisi kumpulan informasi. *Business Reporting* adalah berita publik yang berisi kumpulan informasi proses bisnis dan keterangan dari perusahaan, atau penyediaan informasi secara berkala kepada para pengambil keputusan dalam suatu organisasi untuk mendukung mereka dalam pekerjaan mereka (Ponniah, 2001).

Reporting merupakan hasil dari proses yang berpengaruh terhadap peningkatan perusahaan seperti, *Business Intelligence* dan manajemen pengetahuan. Seringkali pelaksanaannya melibatkan *extract, transform, dan load* (ETL) yang berkoordinasi dengan *data warehouse* dan kemudian menggunakan reporting tools. Sementara *reporting* dapat didistribusikan dalam bentuk cetak atau melalui *web*, mereka juga dapat diakses melalui intranet perusahaan (Andrew, 1999).

2.2.11 Model Pengembangan Perangkat Lunak Waterfall

Menurut Roger S. Pressman, model pengembangan perangkat lunak meliputi model air terjun (*Waterfall Model*), Prototipe (*Prototyping*), RAD (*Rapid Application Development*), Evolusioner dan Formal. Dalam penelitian ini menggunakan model *Waterfall*, model ini dipilih dengan alasan untuk membangun sistem ini dibutuhkan beberapa tahap yang berbeda yang merupakan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial mulai pada tingkat dan kemajuan sistem pada seluruh *analysis, design, coding, dan testing* (Randy, 2012).

- a. Analisis kebutuhan. Proses pencarian kebutuhan diintegrasikan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para programmer harus mengerti tentang informasi dari *software*, misalnya fungsi yang dibutuhkan, *user interface*.
- b. Desain. Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk *software* sebelum *coding* dimulai. Design

harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

- c. Implementasi. Untuk dapat dimengerti oleh mesin, dalam hal ini komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap *design* yang secara teknis nantinya dikerjakan oleh *programmer*.
- d. Pengujian sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat

Penelitian ini mengambil tempat di PT. PLN (Persero) Distribusi Jakarta Raya dan Tangerang pada sub. Bagian distribusi beban pembebanan khusus gardu induk Jakarta raya. Adapun pengambilan data telah dilakukan mulai bulan Januari 2017 sampai dengan bulan Maret 2017 yang meliputi penyusunan laporan, penelitian, pengolahan data dan penyusunan laporan.

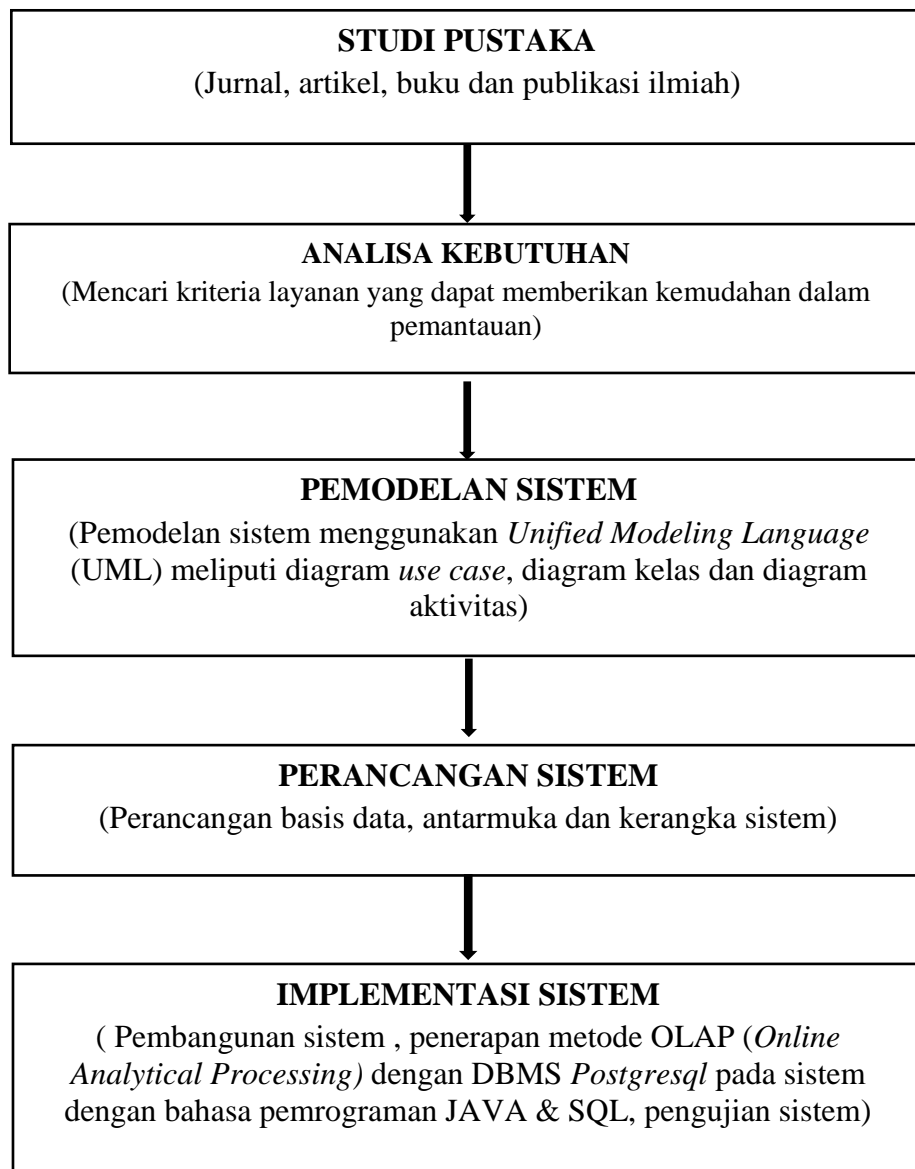
Sumber data yang digunakan dalam penelitian ini adalah data mengenai besar beban *feeder* daya energi listrik berupa ampere dan Kwh. Data *feeder* ini dengan inputan manual dengan interval perjam setiap harinya pada setiap Gardu Induk. data history yang disimpan selama 12 bulan pada tahun 2016 terhitung dari bulan Januari – Maret 2017.

3.2 Bahan dan Alat Penelitian

Bahan data yang digunakan untuk analisa berupa data *excel* total beban *feeder* dan trafo gardu induk distribusi jakarta raya dan tangerang, yang kemudian data tersebut disimpan dengan *format* csv. Untuk mendukung penelitian ini dibutuhkan perangkat keras meliputi: satu unit komputer dengan bahasa pemrograman yang digunakan Java dan SQL

3.3 Prosedure Penelitian

Dalam upaya untuk memperoleh hasil yang optimal, penelitian ini dilakukan melalui tahapan-tahapan yang akan dikerjakan pada penelitian untuk analisis efisiensi pemakaian besar beban daya energi listrik menggunakan OLAP. Tahapan-tahapan yang akan dilakukan ditunjukkan pada gambar 3.1.



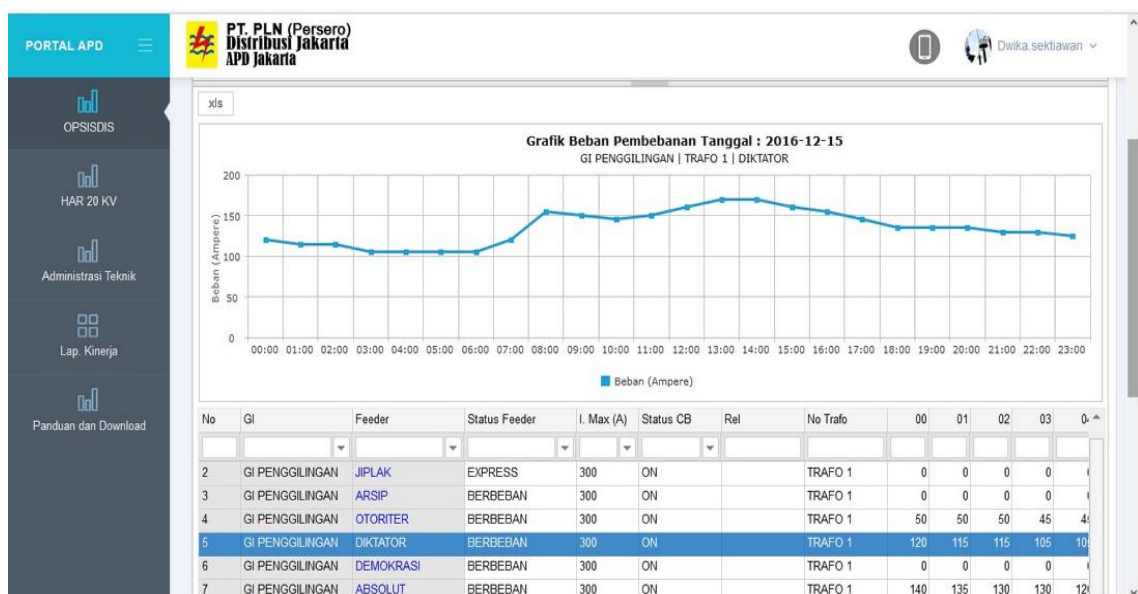
Gambar 3.1 Prosedur penelitian

Literatur yang digunakan dalam penelitian ini berupa jurnal, buku, artikel, dan publikasi ilmiah yang berkaitan dengan penerapan metode OLAP dan sistem tenaga listrik khususnya pada beban distribusi energi listrik. Pada tahapan ini juga dilakukan kajian terhadap jurnal-jurnal pendukung yang dapat memberikan referensi tambahan dalam landasan penelitian.

Pembuatan program aplikasi penelitian ini sesuai dengan pengembangan perangkat lunak model *waterfall* (air terjun). Diawali dengan tahap analisa

kebutuhan, permodelan (perancangan), imlementasi, dan pengujian sistem. Pada tahap analisa kebutuhan dimulai dengan mempelajari sistem informasi portal internal perusahaan berbasis *Web* yang sedang berjalan pada PT. PLN Distribusi Jakarta Raya dan Tangerang untuk mengidentifikasi permasalahan. Langkah selanjutnya adalah menganalisa kebutuhan data dan informasi yang dibutuhkan untuk sistem informasi yang diusulkan. Pada tahap ini yang dilakukan adalah mengumpulkan data yang berkaitan dengan total beban *feeder* dan trafo gardu induk distribusi daerah khusus DKI Jakarta Raya pada PT. PLN Distribusi Jakarta Raya dan Tangerang.

PT. PLN Distribusi Jakarta Raya dan Tangerang, khususnya Jakarta Raya memiliki 22 Area Distribusi, 55 Gardu Induk dan 1906 *feeder*. Data yang masuk pada *web* portal tidak *real time*, setiap gardu memiliki operator yang shift bertugas menginput besar beban *feeder* pada setiap gardu induk dengan interval inputan satu jam sekali. Maka setiap area distribusi memiliki grafik pemantauan beban *feeder* dan grafik pemantau beban trafo pada Web portal APD PT. PLN Distribusi Jakarta. Berikut contoh gambaran grafik pemantauan beban *feeder* pada tanggal 15 Desember 2016 dapat di lihat pada gambar 3.2.



Gambar 3.2. Contoh gambar grafik beban *feeder*

Perancangan Sistem Informasi analisis beban *feeder* dengan OLAP yang diusulkan dengan pemodelan sistem yaitu :

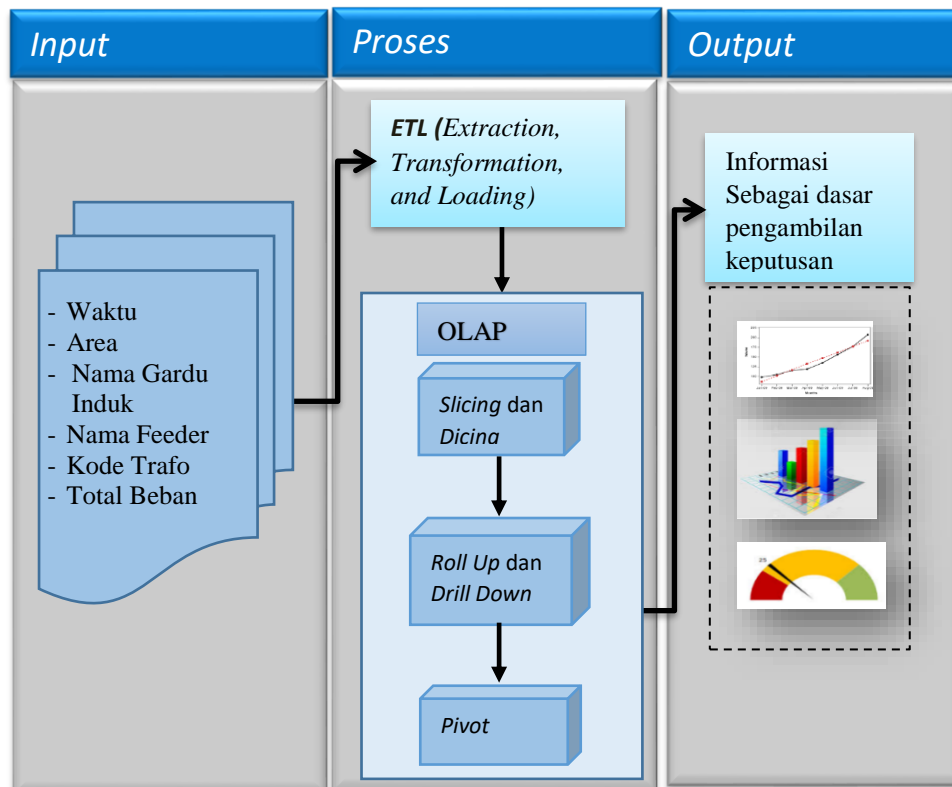
- a) Membuat konteks diagram yang merupakan level tertinggi (*top level*), diagram yang menggambarkan hubungan antar sistem dengan entitas diluar sistem, merupakan gambaran sistem secara keseluruhan.
- b) Membuat diagram arus data yang akan menjelaskan sistem lebih detail yang merupakan turunan dari sebuah konteks diagram. Diagram arus data akan menjelaskan setiap proses yang terjadi dalam sistem.
- c) Membuat desain basis data Sistem Informasi untuk menunjang analisis.

Sistem yang dirancang dan dibangun kemudian diimplementasikan dengan menggunakan data yang berupa *excel* dengan menggunakan *tools Pentaho Community 6.1*. Dengan proses awal data yang diperoleh tersebut di ekstrak sebagai tahap awal pada proses ETL dengan mengedit data yang ada pada *excel*, kemudian data tersebut diproses transform dan *load* pada aplikasi *data integration*. Tahapan proses ETL data tersimpan pada Gudang data yang biasa disebut dengan *data warehouse*. Penyimpanan *data warehouse* membutuhkan aplikasi *database server* untuk penyimpanannya. *Postgresql* merupakan salah satu aplikasi database server, database server ini harus di hubungkan dengan aplikasi *pentaho* dengan nama *database connection*. Sehingga Aplikasi *database server Postgresql* untuk pengolahan manajemen basis data, program *Pentaho Community 6.1* untuk aplikasi sistemnya.

Pengujian sistem dilakukan setelah Sistem analisis Informasi distribusi beban *feeder* energi listrik dibangun dan diimplementasikan. Dengan menguji fungsional yang ada dalam sistem dengan melihat keluaran yang dihasilkan dengan menggunakan metode *black box*. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji di bangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak di cek apakah sudah sesuai dengan yang di harapkan atau tidak. Pengujian dilakukan dengan cara memberi inputan dari pengguna kepada sistem yang sudah berjalan dan mengamati *output* yang dihasilkan.

3.4 Kerangka Sistem

Kerangka sistem rancang bangun untuk Analisis distribusi daya pemakaian listrik pada area pengaturan distribusi PT. PLN dengan menggunakan OLAP dapat dilihat pada gambar 3.3.



Gambar 3.3 Kerangka sistem

3.5 Tahapan Pengolahan

Pembuatan sistem analisa ini dapat digambarkan pada tahapan pengolahan untuk penyimpanan *data warehouse*. Diawali dengan tahap pengolahan data mentah, ETL, *cube* pada OLAP untuk melihat dimensi analisa. tahapan pengolahan yang akan digunakan untuk membangun *data warehouse* untuk Sistem analisis Informasi distribusi beban *feeder* energi listrik PT. PLN yang dapat dijabarkan sebagai berikut :

1. Pengolahan Data

Pada tahap ini yang yang dilakukan adalah mengumpulkan data yang berkaitan dengan total beban *feeder* dan trafo gardu induk distribusi daerah khusus DKI Jakarta Raya pada PT. PLN (Persero) Distribusi Jakarta Raya dan Tangerang. Data yang diperoleh merupakan basis data yang menjelaskan keseluruhan data berupa file *excel* yang menjelaskan informasi distribusi beban *feeder* dan siap di analisa untuk mengetahui puncak pemakain daya pada setiap harinya.

2. ETL (Ekstraksi dan Transformasi) Tahap 1

Tahap Ekstraksi Data, Pada tahap ini dilakukan pemilihan data yang akan digunakan dalam perancangan *data warehouse*. Pemilihan ini dilakukan terhadap data yang sudah dikumpulkan tadi. Pada data total beban *feeder* dan trafo induk distribusi Jakarta Raya pada tahun 2016 terdapat 39 - 41 fields tiap sheet data yaitu NAMA GI, STATUS *FEEDER*, NO URUT, NAMA *FEEDER*, AREA, ID TRAFO, NO TRAFO, MAX ARUS, Bulan yang terdiri dari tanggal 1, 2, 3, 4, 5, 6,, 31 (tanggal tergantung dari nama bulan), Total, Kwh. Pada data setiap tanggal terdapat data terpisah yang terdiri dari 24 fields data total beban *feeder* waktu yang di input manual sesuai dengan interval jam nya, yaitu J00, J01, J02, J03, ..., J23. Pada data gardu induk Jakarta Raya terdapat field jumlah area, jumlah trafo, jumlah gardu induk.

Tahap Transformasi Data (*Data Transformation*), merupakan tahapan penyesuaian data yang sudah diekstrak agar kompatibel dengan *data warehouse* yang akan dibangun. Pada data total beban *feeder* dan trafo distribusi banyak terdapat nama *field* yang masih menggunakan spasi sementara pada saat kita membangun data model tidak diperkenankan untuk menggunakan spasi maka kesalahan tersebut harus diperbaiki. Contohnya NAMA GI menjadi NAMA_GARDU_INDUK, STATUS *FEEDER* menjadi STATUS_*FEEDER*. Pada data total beban *feeder* dan trafo distribusi masih belum adanya kode data, seperti pada NAMA GI, pada fields ini hanya menjelaskan nama gardu induk tanpa ada kode dari nama nama gardu induk tersebut. Juga masih terdapat pemberian kode gardu induk yang tidak beraturan. Perlu diberikan pengaturan yang baru dengan

cara memberikan kode gabungan yaitu gabungan dari kode inisial ditambah kode organisasi perusahaan dan kemudian diikuti dengan nomor urut pada di NO URUT.

3. ETL (*Loading*) Tahap II

Tahapan ETL pada loading merupakan merancang model data yang akan digunakan sebagai tempat penyimpanan data yang sudah ditransformasi, model yang dirancang dengan Model Star Schema. Berdasarkan data yang didapatkan di PT. PLN, maka model data Star Schema yang akan dibangun dapat dilihat pada gambar 3.4. Pada proses data loading atau memasukkan data ke dalam *data warehouse* dapat menggunakan sebuah user interface yang dibangun dengan bahasa pemrograman. tahapan pengolahan yang akan digunakan untuk membangun *data warehouse* untuk Sistem analisis Informasi distribusi beban *feeder* energi listrik PT. PLN yang dapat dijabarkan sebagai berikut :

1. Pengolahan Data

Pada tahap ini yang yang dilakukan adalah mengumpulkan data yang berkaitan dengan total beban *feeder* dan trafo gardu induk distribusi daerah khusus DKI Jakarta Raya pada PT. PLN (Persero) Distribusi Jakarta Raya dan Tangerang. Data yang diperoleh merupakan basis data yang menjelaskan keseluruhan data berupa file *excel* yang menjelaskan informasi distribusi beban *feeder* dan siap di analisa untuk mengetahui puncak pemakain daya pada setiap harinya.

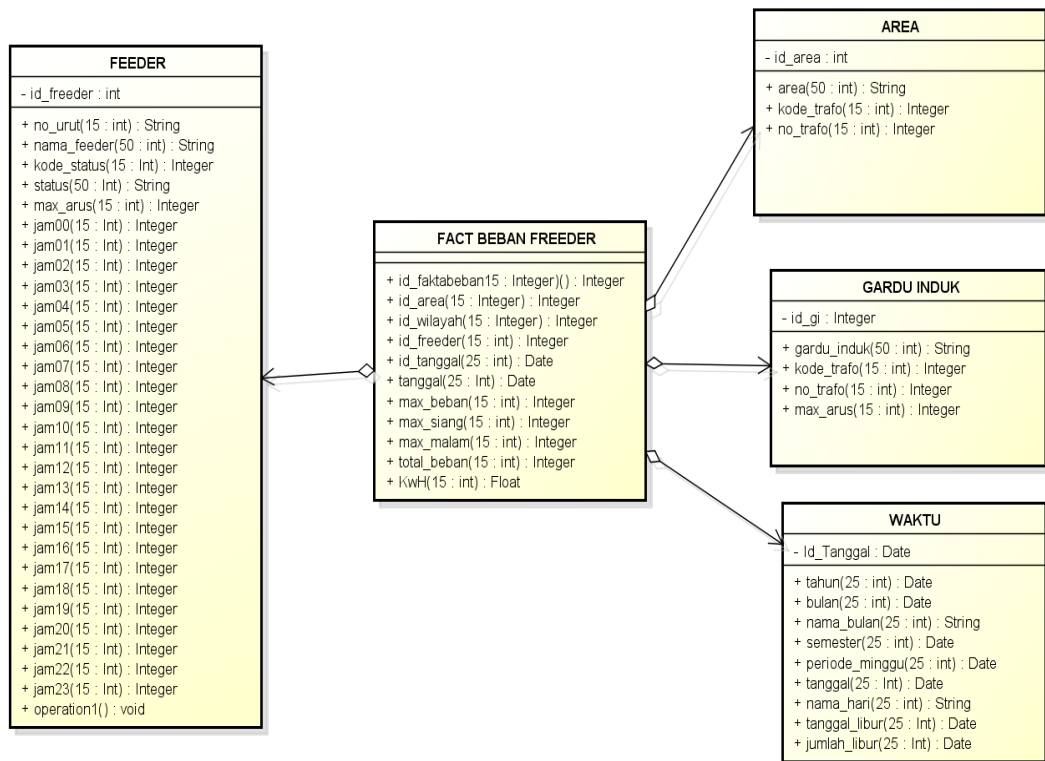
2. ETL (Ekstraksi dan Transformasi) Tahap 1

Tahap Ekstraksi Data, Pada tahap ini dilakukan pemilihan data yang akan digunakan dalam perancangan *data warehouse*. Pemilihan ini dilakukan terhadap data yang sudah dikumpulkan tadi. Pada data total beban *feeder* dan trafo induk distribusi Jakarta Raya pada tahun 2016 terdapat 39 - 41 fields tiap sheet data yaitu NAMA GI, STATUS *FEEDER*, NO URUT, NAMA *FEEDER*, AREA, ID TRAFO, NO TRAFO, MAX ARUS, Bulan yang terdiri dari tanggal 1, 2, 3, 4, 5, 6,, 31 (tanggal tergantung dari nama bulan), Total, kWh. Pada data setiap tanggal terdapat data terpisah yang terdiri dari 24 fields data total beban *feeder* waktu yang di input manual sesuai dengan *interval* jam nya, yaitu J00, J01, J02, J03, ..., J23. Pada data gardu induk Jakarta Raya terdapat *field* jumlah area, jumlah trafo, jumlah gardu induk.

Tahap Transformasi Data (*Data Transformation*), merupakan tahapan penyesuaian data yang sudah diekstrak agar kompatibel dengan *data warehouse* yang akan dibangun. Pada data total beban *feeder* dan trafo distribusi banyak terdapat nama *field* yang masih menggunakan spasi sementara pada saat kita membangun data model tidak diperkenankan untuk menggunakan spasi maka kesalahan tersebut harus diperbaiki. Contohnya NAMA GI menjadi NAMA_GARDU_INDUK, STATUS *FEEDER* menjadi STATUS_*FEEDER*. Pada data total beban *feeder* dan trafo distribusi masih belum adanya kode data, seperti pada NAMA GI, pada *fields* ini hanya menjelaskan nama gardu induk tanpa ada kode dari nama nama gardu induk tersebut. Juga masih terdapat pemberian kode gardu induk yang tidak beraturan. Perlu diberikan pengaturan yang baru dengan cara memberikan kode gabungan yaitu gabungan dari kode inisial ditambah kode organisasi perusahaan dan kemudian diikuti dengan nomor urut pada di NO URUT.

3. ETL (*Loading*) Tahap II

Tahapan ETL pada *loading* merupakan merancang model data yang akan digunakan sebagai tempat penyimpanan data yang sudah ditransformasi, model yang dirancang dengan Model *Star Schema*. Berdasarkan data yang didapatkan di PT. PLN (Persero), maka model data *Star Schema* yang akan dibangun dapat dilihat pada gambar 3.4. Pada proses data loading atau memasukkan data ke dalam *data warehouse* dapat menggunakan sebuah *user interface* yang dibangun dengan bahasa pemrograman.



Gambar 3.4 Model *star schema* untuk *data warehouse* beban *feeder* dan trafo gardu induk ditribusi

4. Tahap Desain OLAP

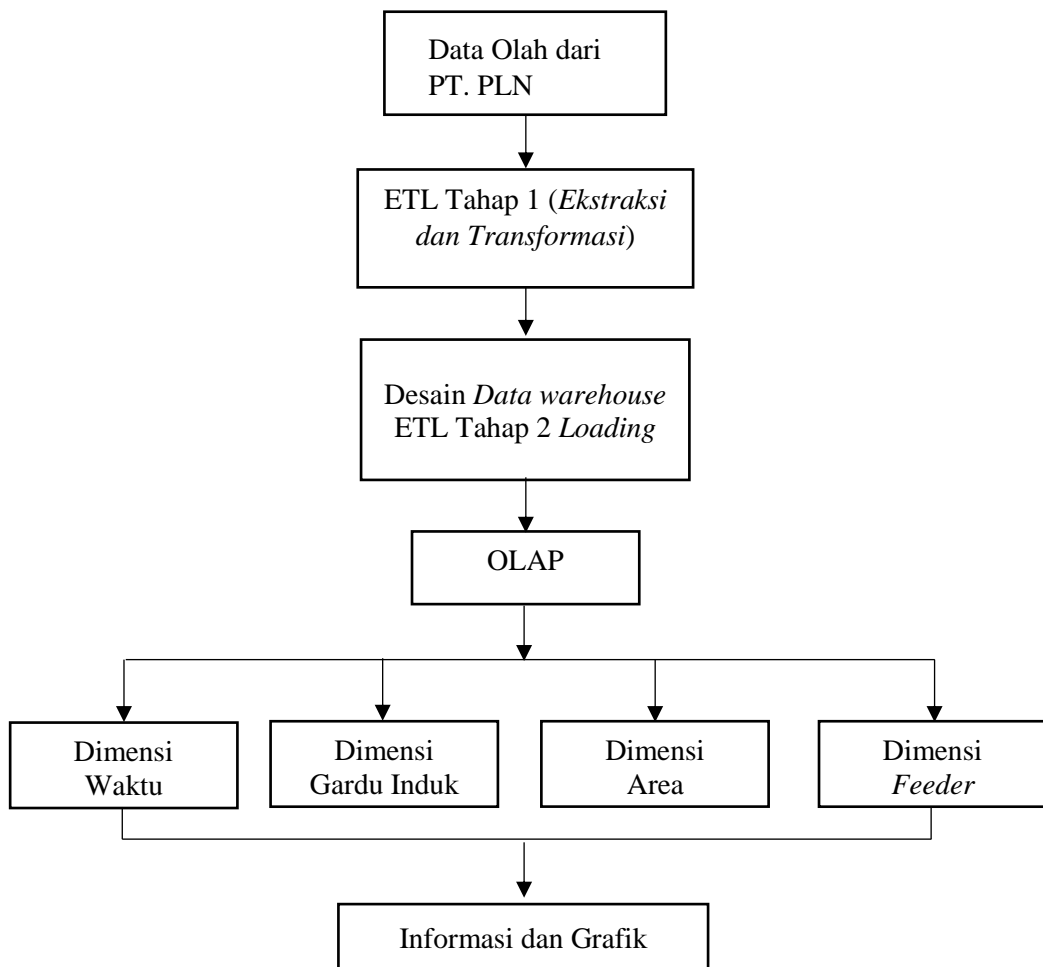
Kemampuan yang dimiliki oleh OLAP (*Online Analytical Processing*) memungkinkan user untuk dapat melihat suatu data dengan analisa yang di lihat dari sisi dimensi yang berbeda. Pada penelitian ini OLAP akan didesain dari beberapa dimensi yaitu : dimensi waktu, dimensi area, dimensi gardu induk dan dimensi *feeder*.

5. *Cube* dimensi

dimensi yang dibuat terdiri dari dimensi waktu, dimensi area, dimensi gardu induk dan dimensi *feeder*. Dimensi-dimensi ini memiliki keterikatan satu dengan lainnya. Seperti *id_trafo* yang di letakan pada dimensi *feeder*, juga di letakan pada dimensi gardu induk, serta dimensi *id_beban feeder* di letakan pada dimensi *feeder* dan dimensi area. Pada dimensi waktu melihat tampilan dari analisa perhitungan tahun, bulan, semester, minggu serta hari.

6. Informasi dan Grafik

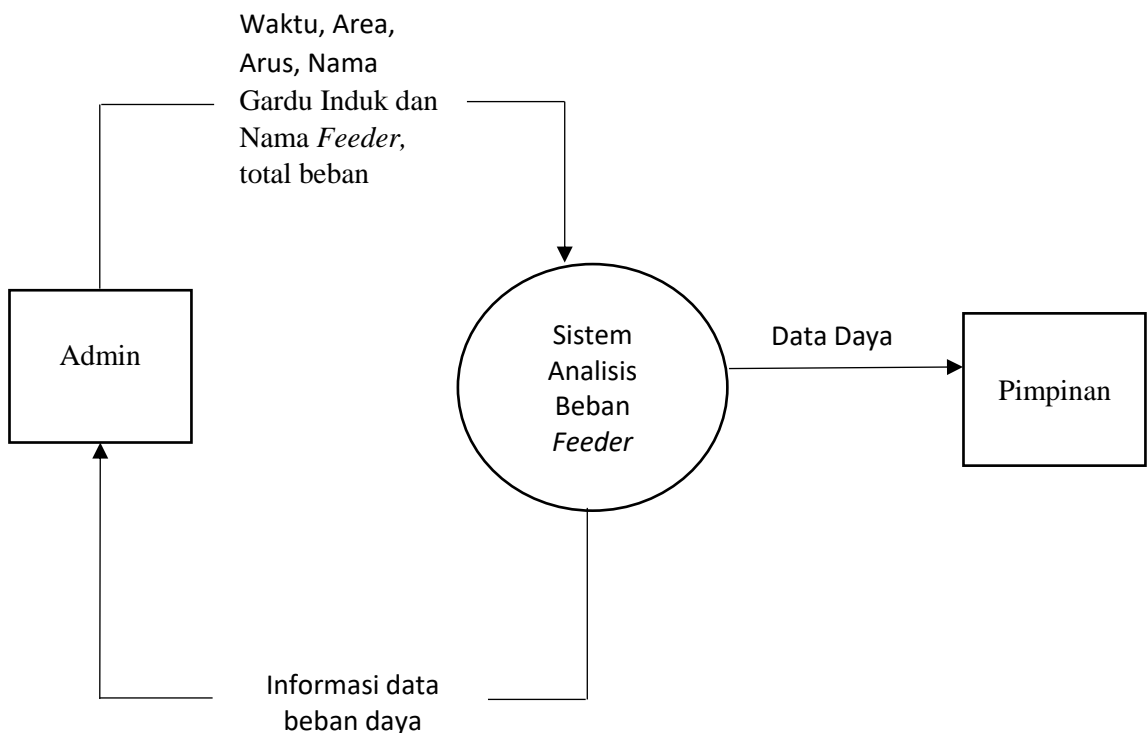
Informasi dan grafik merupakan tahapan akhir dari hasil tampilan pada proses analisis distribusi beban *feeder*. Hasil *output* pengolahan yang disajikan berupa analisa penyajian informasi data, grafik, dan *dashboard*. Hasil akhir yang akan ditampilkan berupa analisa berdasarkan dimensi yang di butuhkan untuk dilihat hasilnya. Hasil akhir juga bisa menampilkan penyajian informasi data dari dua dimensi atau lebih. Sehingga output yang diberikan akan merincikan tahapan dari suatu distribusi besar seperti area hingga distribusi kecil seperti *feeder* dengan rincian waktu dari perhitungan tahun hingga perhitungan waktu pada tanggal. Penjelasan tahapan-tahapan pengolahan yang telah dijelaskan di atas, dapat dilihat pada gambar 3.5.



Gambar 3.5 Tahapan pengolahan

3.6 Data Flow Diagram (DFD)

Perancangan sistem dimulai dengan diagram konteks yang menggambarkan sistem analisis beban *feeder* dan trafo gardu induk distribusi secara keseluruhan berupa entitas yang berperan dalam sistem, proses input yang terjadi, dan informasi yang dihasilkan berupa laporan. Diagram konteks sistem analisis beban *feeder* dan trafo gardu induk distribusi dapat dilihat pada Gambar 3.5.



Gambar 3.6 Diagram konteks level 0 sistem analisis beban *feeder*

Pada Gambar 3.6 diagram konteks terdapat dua entitas, yaitu Admin dan pimpinan. Pada *web* portal APD PT. PLN (Persero) Admin bertugas untuk menginput semua data beban *feeder* dengan interval tiap jam setiap harinya secara manual, gardu induk distribusi *feeder* area, nama gardu induk, status *feeder*, nama *feeder* dan no trafo. Admin mengekstraksi data dari portal di *insert* pada sistem analisa OLAP yang kemudian data tersebut diolah pada sistem analisis sehingga menghasilkan informasi berupa laporan. Pimpinan Distribusi Disjaya PT.PLN

(Persero) sebagai pihak pengambil keputusan. Laporan-laporan diajukan kepada Pimpinan Distribusi Disjaya PT.PLN (Persero) sebagai rekomendasi dalam perbaikan kondisi ke depan.

3.7 Perancangan Basis Data

Data mentah total beban *feeder* dan trafo gardu induk distribusi Jakarta sekitarnya yang merinci *fields* tabel data yang dibutuhkan di proses dan di analisa, berikut gambar 3.7 yang menggambarkan data mentah total beban *feeder*:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	
1	TANGGAL_NM_LG	STATUS_PNYL	NO_UPLU_NML_PMYL	aj	id_trafo	NO_TRA	MAX_ARUS	J00	J01	J02	J03	J04	J05	J06	J07	J08	J09	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	Max	Total	Max Sian	Max Malam	AL			
2	1/31/2016	AGP	BERBEBAN	1 HADIAH	MENTENG	1011 TRAFO 1	300	45	40	40	40	40	40	55	55	55	60	70	75	75	75	70	70	65	70	70	70	65	60	55	50	75	1405	75	70				
3	1/31/2016	AGP	SPOOT LOAD	2 CAKAP	MENTENG	1011 TRAFO 1	300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
4	1/31/2016	AGP	BERBEBAN	3 NOTES	MENTENG	1011 TRAFO 1	300	20	20	20	20	20	20	20	20	30	90	110	110	120	120	125	125	115	125	100	95	85	75	65	55	45	125	1775	125	125			

Gambar 3.7 Tabel data excel total beban *feeder* dan trafo gardu induk

Berdasarkan pemodelan *start schema*, yang telah dilakukan sebelumnya, dilakukan perancangan tabel-tabel yang dibutuhkan oleh sistem pendukung keputusan pada penjaminan mutu layanan perpustakaan dalam melakukan manajemen data. Tabel-tabel tersebut dijelaskan sebagai berikut:

a. Tabel Waktu

Nama tabel : db_tanggal

Kunci Utama : Id_tanggal

Fungsi : Untuk menyimpan data dimensi waktu

Tabel 3.1 Struktur tabel db_tanggal

No	Nama <i>Field</i>	<i>Type</i> dan Panjang <i>Field</i>	Keterangan
1	id_tanggal	Integer	Nomor id tanggal
2	tahun	Date	Tahun
3	bulan	Date	Bulan
4	semester	Date	Semester
5	periode_minggu	Date	Mingguan
6	tanggal	Date	Tanggal
7	nama_bulan	nvarchar(24)	Nama bulan
8	nama_hari	nvarchar(24)	Nama hari
9	tanggal_libur	Date	Tanggal libur
10	Jumlah_libur	Date	Jumlah hari libur

b. Tabel Area

Nama tabel : db_Area

Kunci Utama : Id_Area

Fungsi : Untuk menyimpan data dimensi Area

Tabel 3.2 Struktur tabel db_Area

No	Nama <i>Field</i>	<i>Type</i> dan Panjang <i>Field</i>	Keterangan
1	id_area	Integer	Nomor id area
2	area	nvarchar(50)	Nama area
3	kode_trafo	nvarchar(15)	kode trafo
4	no_trafo	nvarchar(15)	no trafo

c. Tabel *Feeder*

Nama tabel : *db_feeder*

Kunci Utama : *Id_feeder*

Fungsi : Untuk menyimpan data dimensi *feeder*

Tabel 3.3 Struktur tabel *db_feeder*

No	Nama <i>Field</i>	Type dan Panjang <i>Field</i>	Keterangan
1	<i>id_feeder</i>	Integer	Nomor id <i>feeder</i>
2	<i>no_urut</i>	Integer	No urut <i>feeder</i>
3	<i>nama_feeder</i>	nvarchar(50)	Nama <i>feeder</i>
4	<i>id_status</i>	nvarchar(15)	Nomor id status <i>feeder</i>
5	<i>status</i>	nvarchar(50)	Nama status <i>feeder</i>
6	<i>j00 – j23</i>	Nvarchar(15)	Jam 00 - 23

d. Tabel Gardu induk

Nama tabel : *db_gardu induk*

Kunci Utama : *Id_Area*

Fungsi : Untuk menyimpan data dimensi gardu induk

Tabel 3.4 Struktur tabel *db_gardu induk*

No	Nama <i>Field</i>	Type dan Panjang <i>Field</i>	Keterangan
1	<i>id_gi</i>	nvarchar(15)	Nomor id gardu induk
2	<i>nama_gardu_induk</i>	nvarchar(50)	Nama gardu induk
3	<i>kode_trafo</i>	nvarchar(15)	Id trafo
4	<i>no_trafo</i>	nvarchar(15)	No trafo
5	<i>max_arus</i>	Nvarchar(15)	Max arus

e. Tabel Fakta Beban *Feeder*

Nama tabel : *db_fact_beban_feeder*

Kunci Utama : *Id_Area*

Fungsi : Untuk menyimpan data fakta beban *feeder*

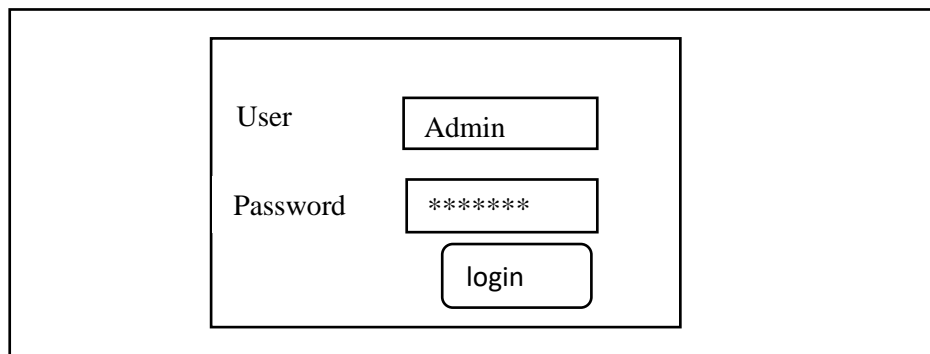
Tabel 3.5 Struktur tabel *db_fakta_beban_feeder*

No	Nama <i>Field</i>	<i>Type</i> dan Panjang <i>Field</i>	Keterangan
1	<i>id_fakta_feeder</i>	Integer	Nomor id fakta <i>feeder</i>
2	<i>id_area</i>	Integer	Nomor id area
3	<i>id_gi</i>	Integer	Nomor id gardu induk
4	<i>id_feeder</i>	Integer	Nama id <i>feeder</i>
5	<i>id_tanggal</i>	Integer	Nomor id tanggal
6	<i>Max_beban</i>	nvarchar(15)	Maximum beban
7	<i>min_beban</i>	nvarchar(15)	minimum beban
8	<i>max_siang</i>	Nvarchar(15)	Mak beban siang
9	<i>max_malam</i>	Nvarchar(15)	Mak beban malam
10	<i>Total_beban</i>	nvarchar(15)	Total arus beban
11	<i>KwH</i>	nvarchar(15)	Besaran dalam KwH

3.8 Perancangan Antarmuka

Perancangan ini bertujuan memberikan bentuk-bentuk antarmuka (*interface*) yang dibutuhkan didalam sistem yang merupakan rancang bangun dari interaksi antara pengguna sistem analisis. Fungsi-fungsi yang ada dalam sistem informasi analisis dengan OLAP masing-masing fasilitas akan dijelaskan diantaranya, *Graphical User Interface* digunakan untuk menyajikan serangkaian informasi yang dibutuhkan oleh *user* dalam melakukan analisis distribusi beban *feeder* dan analisis besar beban maksimum. Perancangan pada proses tersebut adalah sebagai berikut:

Antarmuka ini dirancang untuk memberikan keamanan pada sistem informasi yang akan dibangun, sehingga data dan informasi yang ada di dalam sistem dapat terlindungi dari pihak-pihak yang tidak memiliki kewenangan dalam mengakses data maupun informasi tersebut. Antarmuka *login* ditunjukkan pada Gambar 3.8.

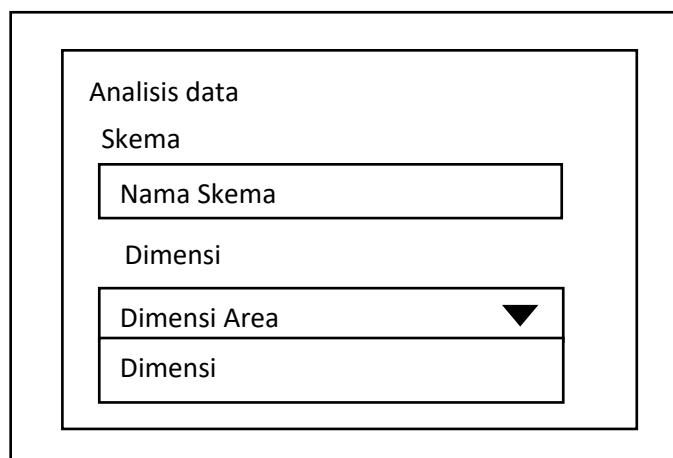


The image shows a login form with the following elements:

- A label "User" next to a text input field containing the text "Admin".
- A label "Password" next to a text input field containing seven asterisks "*****".
- A rounded rectangular button labeled "login" positioned below the password field.

Gambar 3.8 Antarmuka *login*

Seanjutnya antarmuka halaman utama administrator berisi fitur-fitur utama yang dapat diakses oleh administrator meliputi: tampilan untuk melihat analisa baru pada skema pemilihan analisis data sesuai dengan analisa dimensi-dimensi yang dibutuhkan dalam pengolahan data. Rancangan antarmuka tersebut ditunjukkan pada Gambar 3.9.

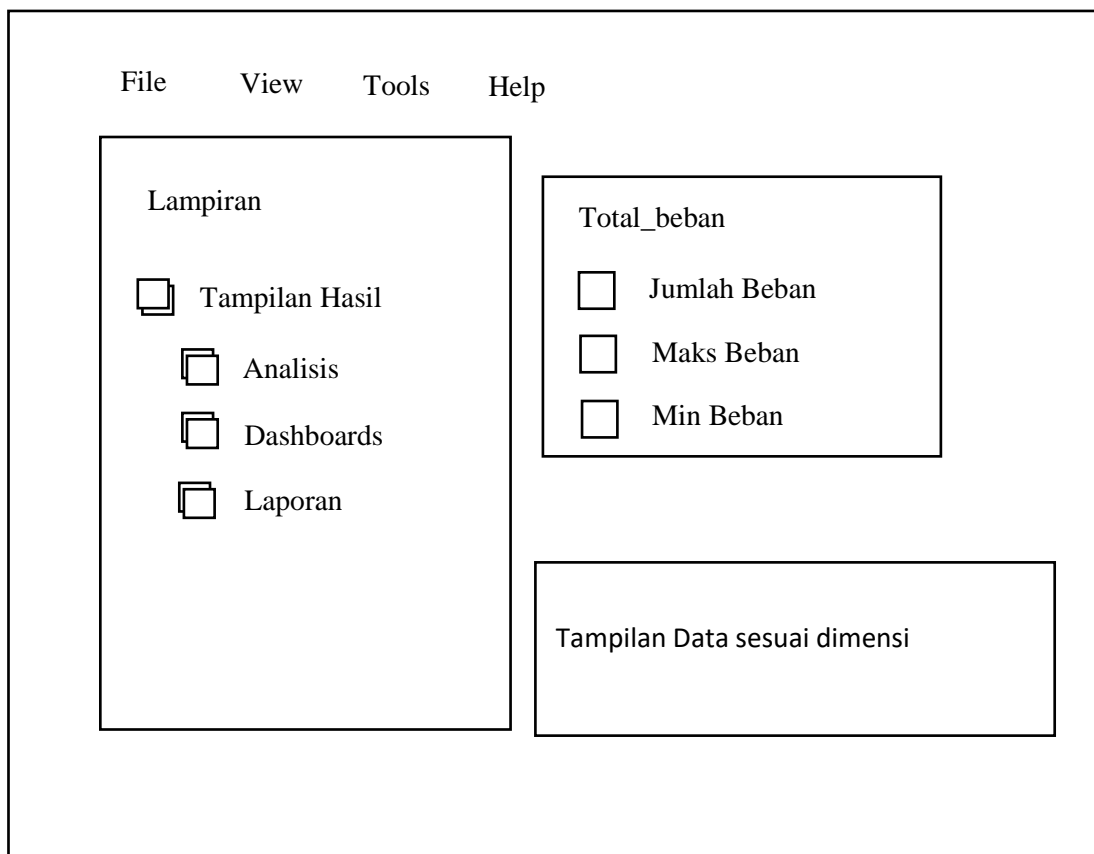


The image shows an administrator interface with the following structure:

- A main container box labeled "Analisis data".
- Inside this container, a sub-section labeled "Skema" contains a text input field for "Nama Skema".
- Below "Skema", another sub-section labeled "Dimensi" contains a dropdown menu for "Dimensi Area" with a downward-pointing triangle icon, and a text input field for "Dimensi".

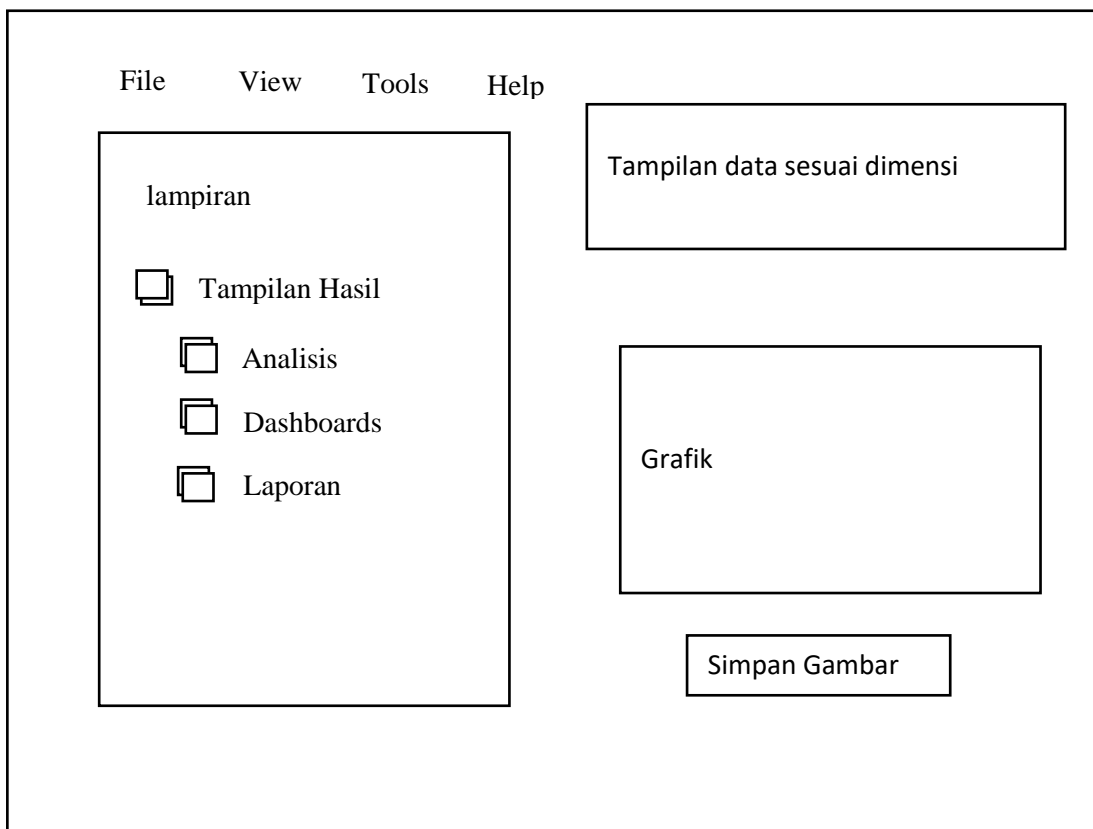
Gambar 3.9 Antarmuka halaman utama *administrator*

Antarmuka halaman analisis pengambilan data meliputi pemilihan tampilan analisis yang dibutuhkan seperti grafik, dashboard, dan *reportingnya*. Tahapan ini memilih tampilan untuk memiliki pada pilihan total beban. Pilihan ini terdiri dari jumlah beban, maksimum beban dan minimum beban. Tahapan ini bisa di lihat pada gambar 3.10.



Gambar 3.10 Antarmuka halaman analisis pengambilan data

Informasi-informasi yang dihasilkan dari proses OLAP akan lebih diperjelas dalam bentuk grafik. Grafik ini berfungsi untuk menyajikan laporan sehingga lebih mudah untuk dianalisis. Tombol *save as excel* berfungsi untuk menyimpan data tabel ke dalam *file excel*. Tombol *save as image* berfungsi untuk menyimpan tampilan grafik berupa gambar sehingga dapat dicetak. Desain halaman *Reporting* dapat dilihat pada Gambar 3.11.



Gambar 3.11 Antarmuka halaman analisis pemilihan perhitungan

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

Hasil yang dicapai dalam merancang sistem informasi untuk menganalisis pemakaian energi listrik yang dapat membantu dalam menyajikan informasi yang berkaitan dengan pada distribusi beban *feeder*. Sistem Analisis yang dibangun dengan menggunakan Bahasa pemrograman berbasis *Java, SQL, database server* sebagai penyimpanan *data warehouse* dengan *postgresql* dan sistem aplikasi *open saource Pentaho Community* dengan memiliki *data integration* berfungsi sebagai *ETL tools* data ekstraksi dengan dengan format *csv* transformasi dan *loading* pada *tools* yang tersedia yang *output* tersimpan pada *database server* sebagai *data warehouse*, *Schema workbench* berfungsi sebagai *cube tools* untuk mendesai *cube* dapa *start schema*, *Report designer* berfungsi sebagai desain laporan *tools* dan *biserver* berfungsi sebagai *web server* yang menampilkan hasil *output* OLAP dengan operasi *drill down* dan *pivot*. Selanjutnya aplikasi pentaho diimplementasikan dengan menggunakan data histori pada PT. PLN (Persero) Disjaya.

4.1.1 Pengumpulan Data

Tahap awal penelitian ini adalah mengumpulkan data yang akan digunakan untuk membangun *data warehouse*. Data yang di kumpulkan bersumber dari *database* histori dari Aplikasi portal PT. PLN Disjaya APD Jakarta yang disimpan dalam *file excel*, aplikasi portal merupakan aplikasi yang digunakan PT. PLN Disjaya serta setiap wilayah area dan gardu induk sebagai *user* pengguna untuk melihat informasi distribusi beban *feeder*. Setiap gardu induk memiliki puluhan *feeder*, *user* pada gardu induk bertugas menginput total beban *feeder* yang didistribusikan pada setiap *feeder*, penginputan dilakukan yang pada setiap jam. Adapun data yang dikumpulkan dapat dilihat pada *table 4.1*

Tabel 4.1 Data yang dikumpulkan dan sumber data

No	Data Yang Dikumpulkan	Keterangan
1	Data Area Jakarta Raya	Data yang dikumpulkan dari Januari 2016 - Maret 2017. Bersumber dari Aplikasi Portal APD di PT. PLN (Persero) Distribusi Jakarta
2	Data Gardu Induk	
3	Data Trafo	
4	Data Beban <i>Feeder</i> Harian	
5	Data Laporan bulanan	Data yang dikumpulkan dari Januari 2016 - Maret 2017. Bersumber dari Bagian Distribusi di PT. PLN (Persero). Data ini didapat secara manual.
6	Data beban puncak dan grafik	

4.1.2 Data Ekstraksi

Data histori bersumber pada aplikasi portal dan data olahan manual pada *file* format *excel* disatukan dan disimpan dalam *file* format *excel*. Sehingga data yang di peroleh peneliti dari pihak PT. PLN Disjaya hanya terdapat satu *file* dalam format *excel*. Satu jenis *file excel* dengan memiliki beberapa *fields* di ekstraksi menjadi empat *file data excel* yang disesuaikan dengan dimensi analisis. *File* yang diekstraksi ditentukan untuk dapat melihat analisis data total distribusi beban feeder dari empat dimensi, disesuaikan dengan perancangan pada *start schema*, dimensi Area, dimensi wilayah pada gardu induk, dimensi feeder sebagai rincian detail beban distribusi yang dikeluarkan, dan dimensi waktu memberikan informasi dari sisi waktu terdapat total beban maksimum. Pada tahapan data ekstraksi dilakukan pemilihan data yang akan dijadikan input untuk menghasilkan informasi dan penambahan beberapa data. Tahap data awal sebelum ekstraksi merupakan data histori yang diperoleh dari PT. PLN Disjaya, tahap data akhir merupakan data setelah diproses ekstraksi, sehingga memiliki perbedaan pada kedua tahapan tersebut. Pada data Hasil dari ekstraksi dapat dilihat pada tabel 4.2.

Tabel 4.2 Data Ekstraksi

Data Awal (Sebelum Ekstraksi)									
Tanggal	NM_GI	status_ <i>Feeder</i>	NM_ <i>Feeder</i>	Area	kode_ trafo	NO_ TRAFO	MAX_ ARUS	TOTAL	kWh
Data Akhir (Setelah Ekstraksi)									
No.	Data Yang di Ekstraksi			Keterangan					
1	Data Area			Kode_Area, Area, id_Trafo, No_Trafo					
2	Data Gardu induk			Kode_GI, NM_GI, Kode_Status, Status_ <i>Feeder</i> , <i>Feeder</i>					
3	Data Beban			Kode_ <i>Feeder</i> , NM_ <i>Feeder</i> , Kode_Trafo, No_Trafo, MAX_ARUS, J00 - J23					
4	Fakta <i>Feeder</i>			Tanggal, Kode_Trafo, No_Trafo, Max Arus, J00 – J23, Max, Max_Siang, Max_Malam, Total, Kwh					

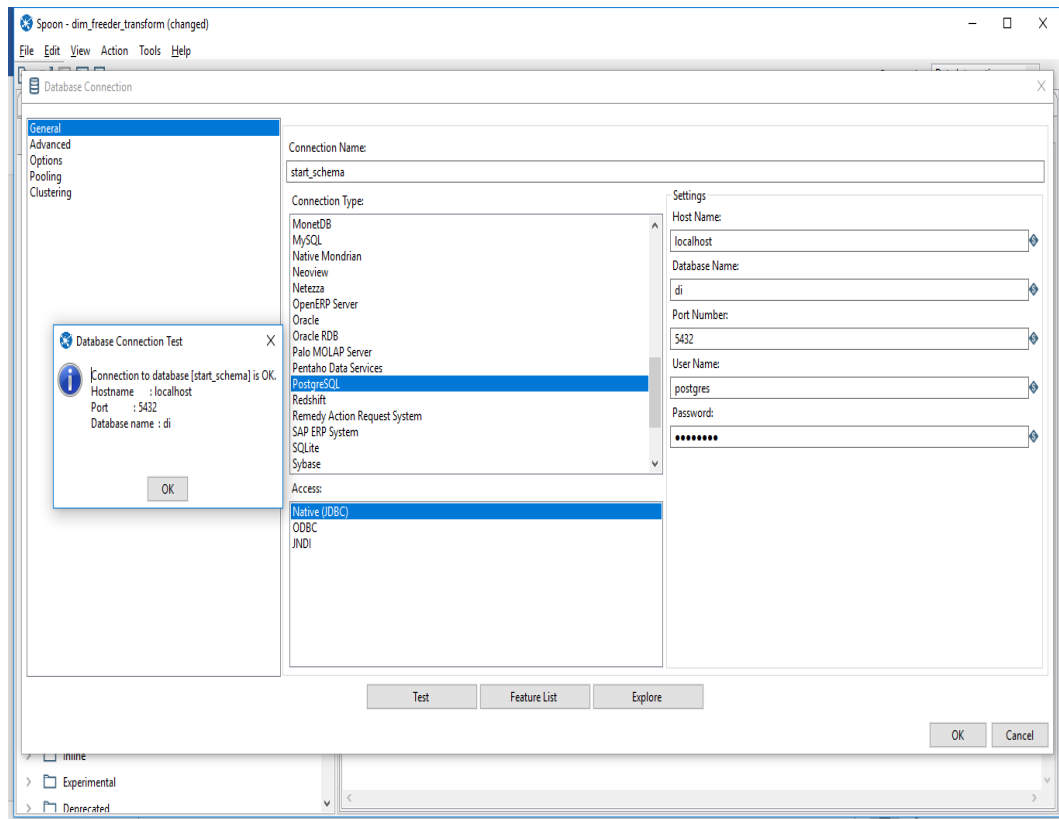
4.1.3 Data Transformasi

Pada hasil *output* tahap akhir data ekstraksi, menjadi acuan pada proses transformasi. Ada beberapa aktivitas yang dilakukan dalam tahapan data transformasi, diantaranya adalah pengaturan ulang pada hasil *output* ekstraksi yang tersimpan dalam format *excel* dengan rincian pengubahan nama *file*, nama *field*, sistem pengkodean data dan format isi *file*. Setelah proses pengubahan dari proses transformasi masih tersimpan dalam *file* format *excel* disimpan kembali dengan *file* format *csv*. format *csv* yang dapat diproses dengan Bahasa pemrograman *sql*. Aktivitas yang dilakukan dalam data transformation dapat dilihat pada tabel 4.3

Tabel 4.3 Data Transformasi

No.	Data	Proses Transformasi yang dilakukan
1	Data Area	kode_area, area, kode_trafo, no_trafo, total
2	Data Gardu induk	kode_gi, nama_gardu_induk, kode_status, status_feeder, kode_feeder, max, kwh
3	Data Beban	kode_feeder, no_urut, nama_feeder, kode_trafo, no_trafo, max_arus, j00, j01, j02, j03, j04, j05, j06, j07, j08, j09, j10, j11, j12, j13, j14, j15, j16, j17, j18, j19, j20, j21, j22, j23, max_siang, max_malam
4	Data Waktu	kode_tanggal, tgl_beban, tahun, bulan, hari, semester, nama_bulan, nama_hari, tanggal_libur, jumlah_libur
5	Data Fakta	Kode_faktafeeder, kode_tanggal, tanggal, kode_area, kode_gi, kode_feeder, kode_trafo, max_arus, j00, j01, j02, j03, j04, j05, j06, j07, j08, j09, j10, j11, j12, j13, j14, j15, j16, j17, j18, j19, j20, j21, j22, j23, total_beban, kwh, max, rata_rata_max_siang, max_malam

Proses data transformasi melakukan proses tahapan penyesuaian data dengan proses hasil *outputnya* pada *data warehouse* yang telah dibangun. Data transformasi diproses aplikasi *data integration* salah satu aplikasi *Pentaho Community*. *Data integration* berfungsi sebagai *ETL tools*. Pada proses awal dilakukan membangun sistem *data warehouse*, dimana *data warehouse* sebagai penyimpanan *database server*. *Database server* yang di instal menggunakan *software* aplikasi *postgresql*. Pada aplikasi *postgresql* untuk membuat *server* utama dan *password* untuk *database connection* dan *folder database* baru untuk penyimpanan data yang diproses dengan aplikasi lainnya. Tahapan awal pada aplikasi *data integration* dengan membuat *database connection* baru dengan *link data warehouse* pada *postgresql* yang telah dibuat. Berikut gambar 4.1 tampilan *database connection* pada proses data transformasi *data integration* dan *data warehouse* pada *postgresql*.

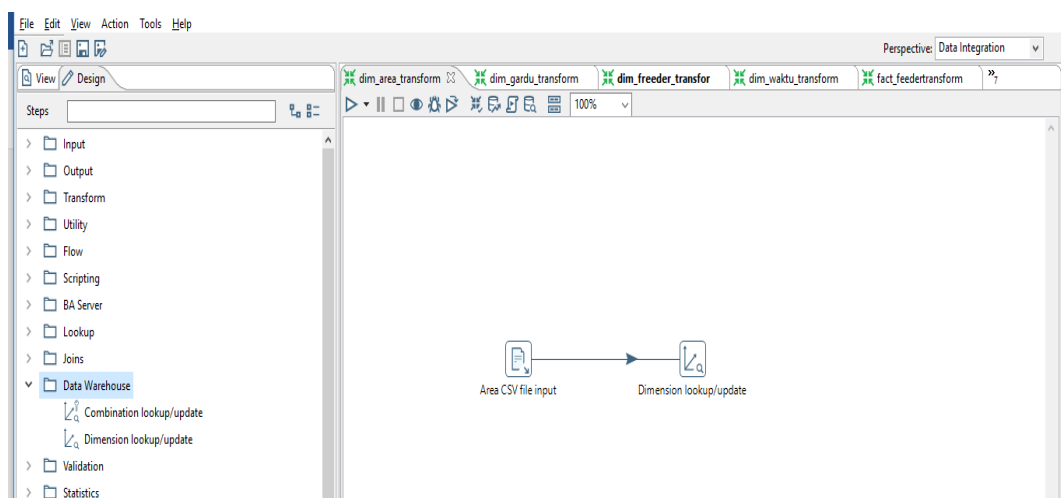


Gambar 4.1 Database connection pada data integration

Penyimpanan *data warehouse* pada *postgresql* dibangun dengan *server* dan password baru dengan nama *postgres*, dan folder *database* baru dengan nama *di*. Seperti pada gambar 4., proses penyimpanan pada *database server* masih bersifat internal *localhost*. Aplikasi *data integration* dengan awal proses membuat *database connection* baru dengan memilih menu *file* pilih *new* pilih *database connection*. Setelah muncul lembar aplikasi *database connection* dengan menu *general*, pada *connection name* di isi bebas, peneliti membuat *connection name* dengan nama *start_schema*, kemudian pilih *connection type*, sebelumnya penyimpanan *data warehouse* di instal pada aplikasi *postgresql*, sehingga *connection type* di pilih *postgresql* dengan port 5432. Langkah selanjutnya mengisi pada tabel *settings*, pada tabel *host name* di isi dengan *localhost*, *database name* dengan *di* disesuaikan dengan *folder* yang di bangun pada *postgresql*, user name dengan *postgres* dan *password* yang sesuai dengan *server* utama yang telah di bangun pada aplikasi

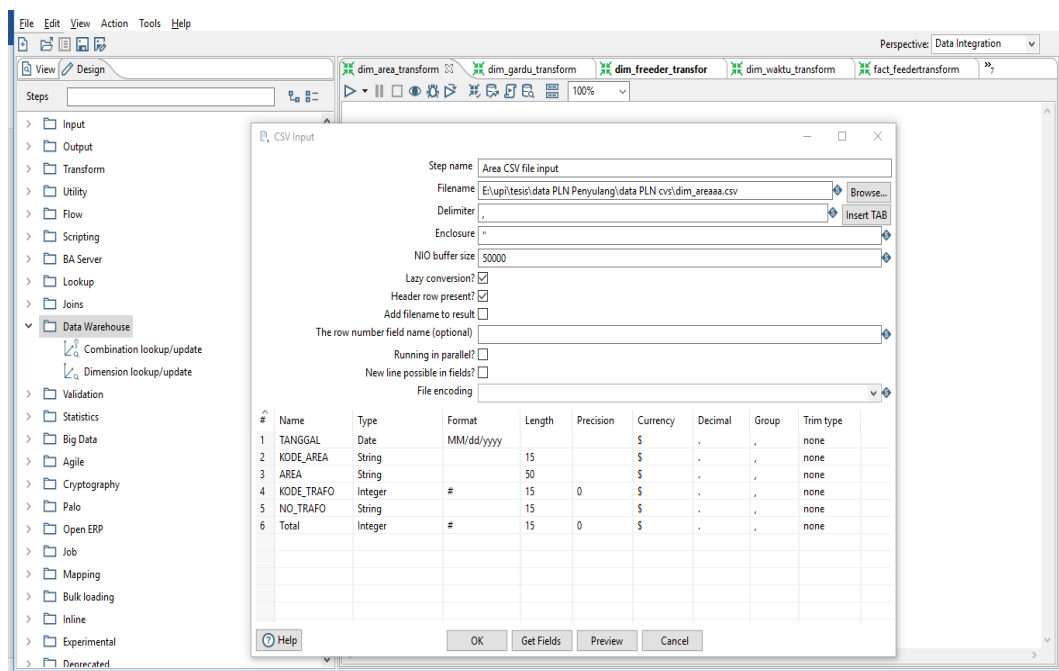
postgresql. Setelah di isi pada lembar *database connection* kemudian klik test, maka akan muncul lembar *database connection test*, dengan keterangan *connection to database [start_schema] is oke*, maka *connection* proses *data integration* hasil *outputnya* masuk pada *database server postgresql* sebagai penyimpanan *data warehouse*.

Proses selanjutnya membuat lembar new transformation pada pertama mentransformasi Aplikasi *data integration*. Menu yang di pilih *file* pilih new dan pilih transformation. Pada lembar *data integration* terdapat dua *tools* perancangan diantaranya *view* dan *design* yang kompleks dengan sumber data yang sangat beragam. Proses transformasi pertama membuat hasil *output* pada *data warehouse* data Area, langkah-langkah pada proses ini dengan pilih *design* klik *input*, maka pada step *input* akan muncul *file* sebagai *input* data, diantaranya *csv file* input, Klik *csv file* input ke lembar kerja transformasi,. Kemudian pilih *tools data warehouse* dan pilih *Dimension lookup/update* sebagai *output* yang akan masuk pada prnyimpanan *data warehouse* pada aplikasi *postgresql* dan letakan pada lembar kerja transformasi. Selanjutnya kita akan menambahkan penghubung antara step input *csv* dan *Dimension lookup/update* untuk *output* yang disebut *hop*. Untuk menambahkan *hop*, klik *csv* input, tekan *SHIFT* dan *drag* ke *xml output*. Kemudian pilih “Main output of step”. Tahapan berikut dapat di lihat pada gambar 4.2



Gambar 4.2 Proses data transformasi

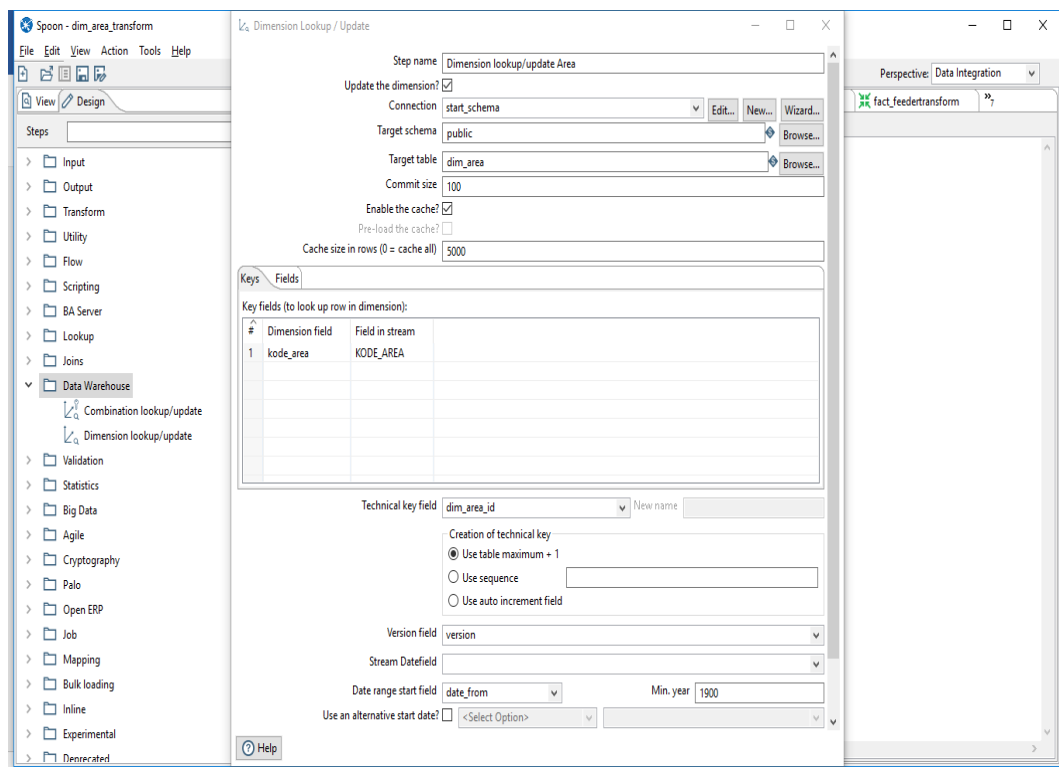
Selanjutnya pengaturan lembar csv *file* input, pada lembar ini merupakan data input proses dari ekstraksi data csv dengan klik *browse* pada *file name* untuk mengambil data histori dengan format csv. Kemudian klik *Get Fields* untuk menampilkan *fields* yang sesuai dengan data pada csv, ubah tipe *file*, format dan panjang karakter sesuai dengan database server *postgresql*. Setelah selesai klik ok. Proses settings data input pada csv *file* input dapat di lihat pada gambar 4.3



Gambar 4.3 Pengaturan CSV file input

Pengaturan *Dimension lookup/update* untuk mendapatkan hasil *output* pada *data warehouse*, dapat di lihat pada gambar 4.4. *connection* yang telah terhubung dengan nama *start_schema*, target *schema* dengan kondisi *public* dan membuat tabel baru pada target *table* untuk tersimpan pada *data warehouse* dengan nama *dim_area*. Menentukan *technical key field* dengan nama *dim_area_id* yang terkoneksi dengan *key fields* pada *file csv* data area dengan nama *fields* kode_area. Pada lembar *fields* klik pada tombol di bayar *get fields* untuk menampilkan field yang telah di transform dari *file csv* file input. Kemudian klik *sql* untuk melihat

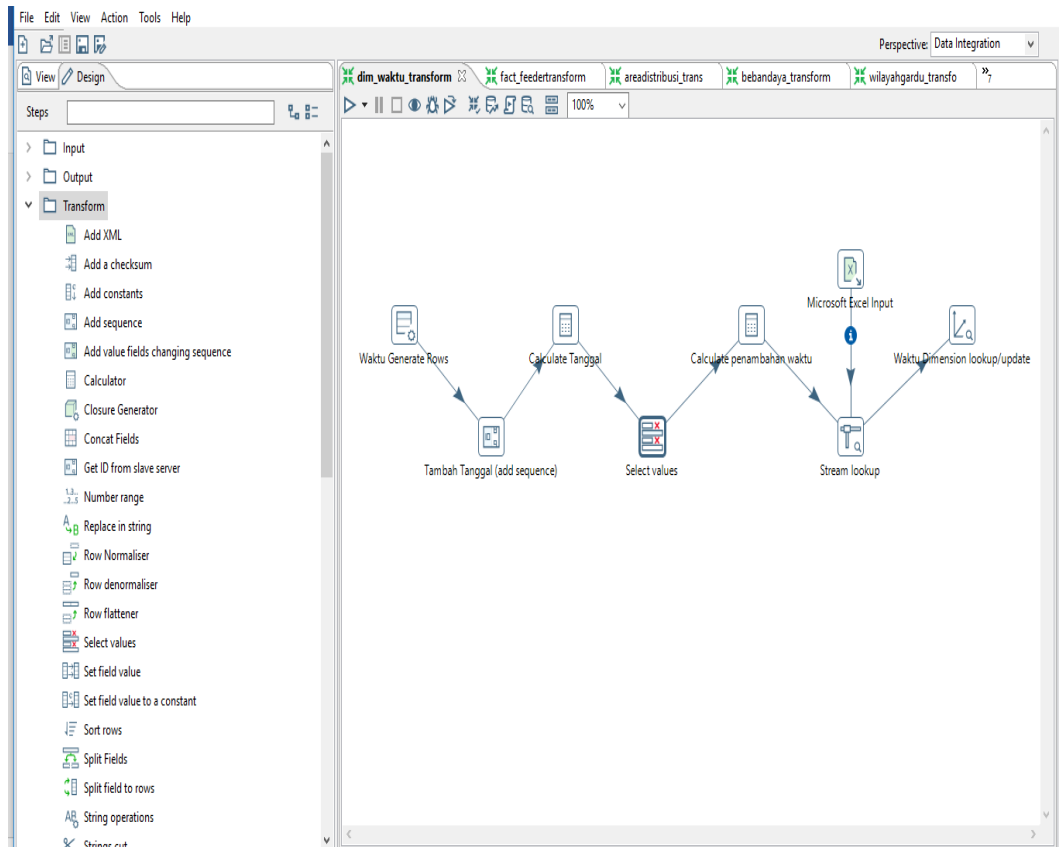
source code hasil transform yang telah berhasil dan dapat di jalankan. Kemudian pada lembar *transformation* disimpan dan dijalankan dengan tombol segitiga atau run. *Run* pada *file* *dim_area* menampilkan hasil proses data berpindah dari *file* *csv* ke database server, tanda *check list* hijau pada *tools* di lembar kerja *transformation* menandakan proses data dimensi area telah berhasil, data telah tersimpan pada *data warehouse* dengan aplikasi *postgresql*.



Gambar 4.4 Pengaturan *dimension lookup/update* dimensi area

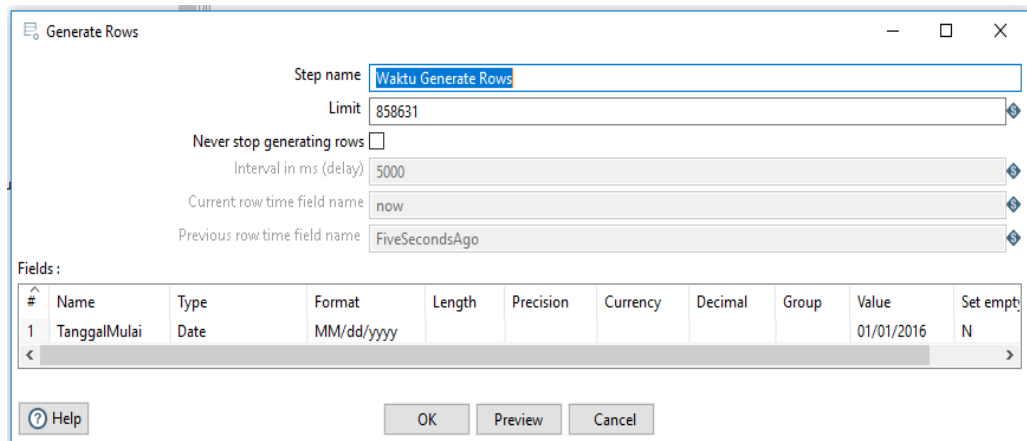
Proses data transformation untuk dimensi gardu induk dan dimensi *feeder* tidak jauh berbeda dengan cara proses data area pada lembar kerja *transformation* pada aplikasi pentaho *data integration*. Yang berbeda hanya hasil simpan data yang telah di ekstraksi dengan nama *file* dan *fields* yang berbeda. Pada target *table* untuk dimensi gardu induk diberi nama dengan nama *dim_gardu* dengan key *fields* *dim_gardu_id*, sedangkan pada dimensi *feeder* target *table* diberi nama *dim_feeder* dengan key *fields* *dim_feeder_id*.

Proses data transformasi pada data waktu memiliki proses yang berbeda, seperti gambar 4.5 menjelaskan step pada prose transformasi untuk data waktu. proses ekstraksi untuk data waktu tidak ada, waktu memiliki step secara otomatis dalam menentukan waktu dari tahun hingga hari.



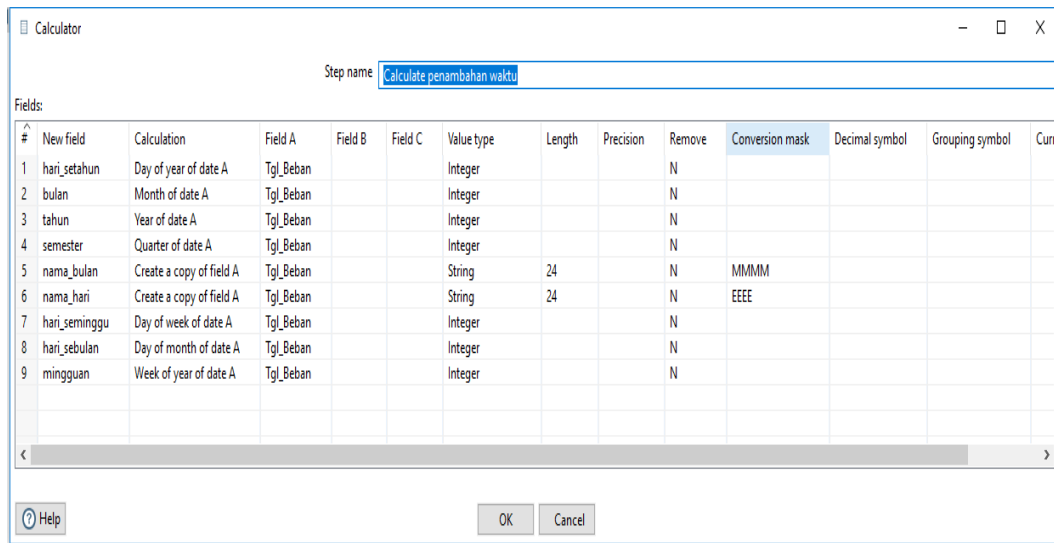
Gambar 4.5 Proses data transformasi waktu

Proses data transformasi untuk mendapatkan data untuk dimensi waktu, dimulai dengan data input, maka *tools* pada *design* di cari step *input* dan pilih *file generate rows*. Letakan *file generate rows* pada lembar kerja transformasi dan klik untuk ubah pada pengaturannya. Pada pengaturan *generate rows* isi limit data, dimana limit ini diisi dengan banyaknya data yang ada pada data fakta *feeder*. Lanjutkan dengan ubah *field* yang ada dibawahnya. Lebih jelasnya dapat dilihat pada gambar 4.6 pengaturan *generate rows*.



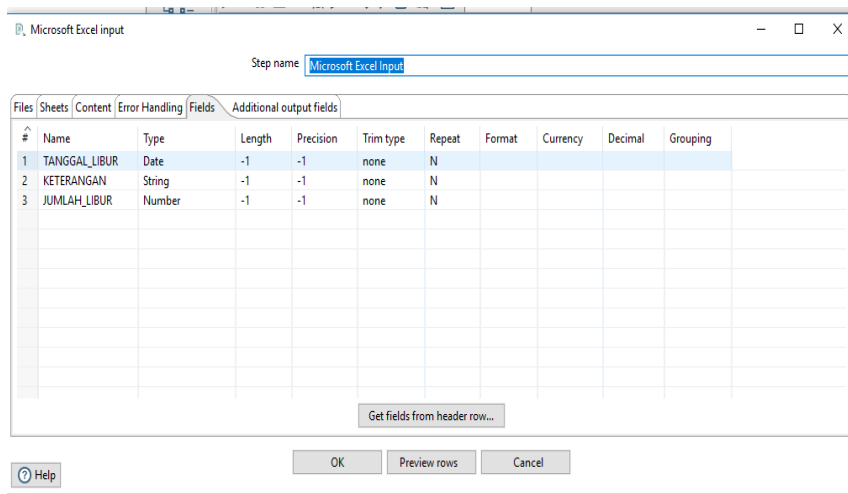
Gambar 4.6 Pengaturan pada waktu *generate row*

Step selanjutnya memilih step transform, dan kemudian memilih *file add sequence*, *file* ini untuk menentukan data bertambah satu atau *increment* secara otomatis, secara data di mulai dari tanggal 1 januari 2016, maka pada pengaturan *add sequence increment* di mulai dengan angka 0, agar pada saat di eksekusi dengan *run* dan hasil pada *data warehouse* sesuai dengan data histori di mulai dengan angka tanggal 1. Lalu dihubungkan pada *hop add sequence* dengan *generate rows*. Selanjutnya dari step yang sama step transform memilih *file calculator 1*, pada tahap proses ini menentukan *key field* terlebih dahulu, pada penelitian ini *tgl_beban* sebagai *key fields*. Hubungkan *hop* antara *calculator* dengan *add sequence*. Berikutnya step tranform dan memilih *file select value*, dan dihubungkan *hop* dengan *calculator 1*. *File calculator 2* di pilih kembali pada step transform, pada *file* ini diubah pada penganturan dengan di isi *fields* yang telah tersedia. *File select value* berfungsi sebagai pengatur dari dua *file calculator* yang akan di proses. *Calculator 2 fields* ditentukan sesuai dimensi waktu yang dibutuhkan, seperti tahun, semester dalam setahun terdapat 4 semester, bulan, mingguan, dan hari. Proses pada pengaturan *calculator 2* diberi nama *calculator* penambahan waktu dapat di lihat pada gambar 4.7.



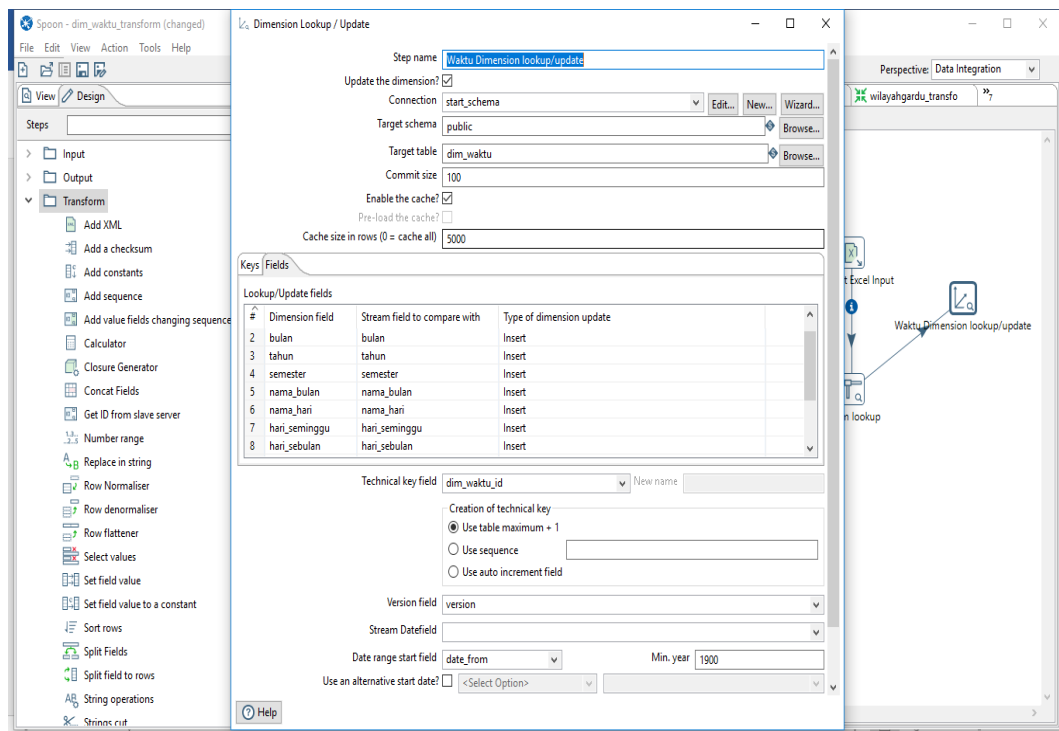
Gambar 4.7 Pengaturan pada *calculator* untuk penambahan waktu

Proses selanjutnya dengan step input dengan memilih *file Microsoft excel* input, sebelumnya membuat *file* tanggal libur nasional selama tahun 2016 dan 2017. Pada *file* tanggal libur nasional di isi dengan *fields* tanggal_libur, keterangan hari libur dan jumlah_libur. *File excel* yang telah di buat, merupakan input untuk tambahan waktu pada dimensi waktu, *file* input pada *Microsoft excel* input di klik lalu di ubah pengaturannya. Pengaturannya dapat di lihat pada gambar 4.8. pada gambar di bawah ini yang di isi *file*, menentukan pada browse penyimpanan *excel* pada tanggal libur, kemudian pada *fields* klik pada tombol di bawah *get fields* untuk menampilkan *fields* yang ada pada *file excel*. Kemudian klik oke, pemindahan data dari *excel* ke *data warehouse* telah berhasil. Input data libur dapat di lihat pada gambar 4.8.



Gambar 4.8 Pengaturan *input* data tanggal libur

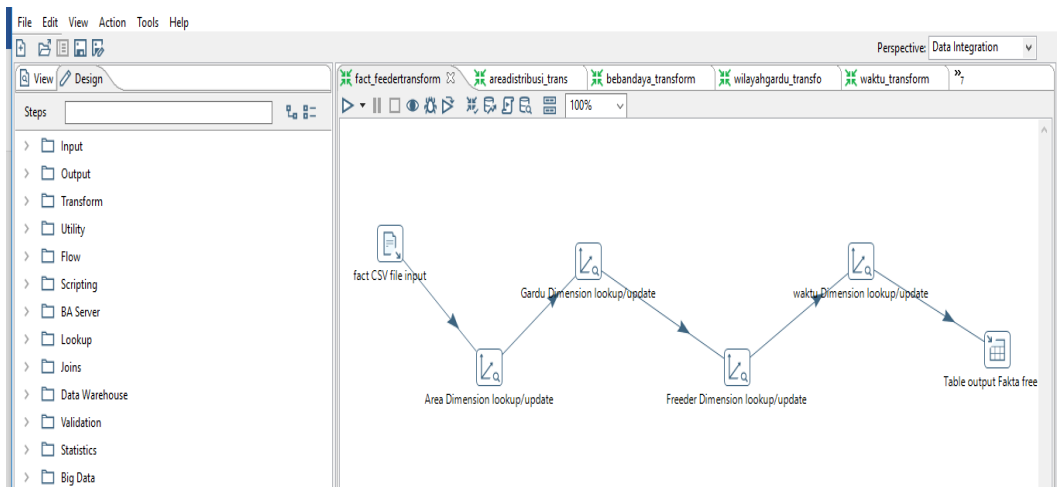
Pengaturan *Dimension lookup/update* untuk dimensi waktu mendapatkan hasil *output* pada *data warehouse*, dapat di lihat pada gambar 4.9. *connection* yang telah terhubung dengan nama *start_schema*, target *schema* dengan kondisi *public* dan membuat tabel baru pada target *table* untuk tersimpan pada *data warehouse* dengan nama *dim_area*. Menentukan *technical key field* dengan nama *dim_waktu_id*. Pada lembar *fields* klik pada tombol di bawah *get fields* untuk menampilkan *fields* yang telah di transform dari *file csv file input*. Kemudian klik *sql* untuk melihat *source code* hasil transform yang telah berhasil dan dapat di jalankan. Kemudian pada lembar *transformation* disimpan dan dijalankan dengan tombol segitiga atau *run*. Proses *run* pada *file dim_area* menampilkan hasil proses memindahkan data dari *file csv* ke database server, tanda *check list* hijau pada *tools* di lembar kerja transform menandakan proses data dimensi area telah berhasil, data telah tersimpan pada *data warehouse* dengan aplikasi *postgresql*.



Gambar 4.9 Pengaturan *dimension lookup/update* dimensi waktu

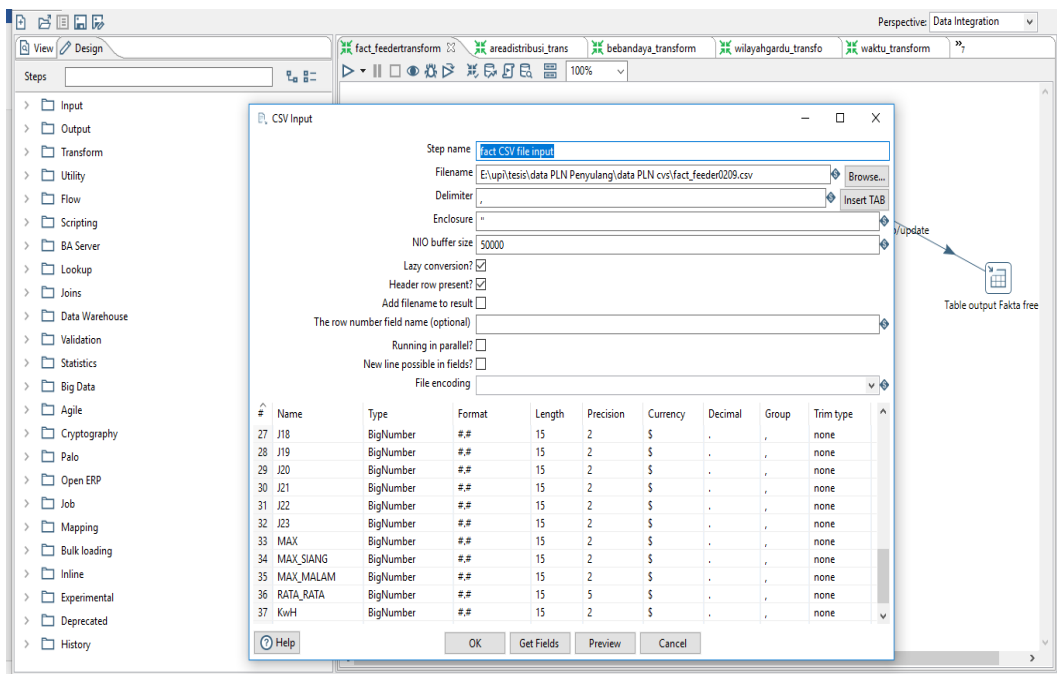
4.1.4 Data Loading

Data loading merupakan proses memasukan data yang sudah di ekstraksi pada data olahan *excel* dan di transformasikan sesuai dengan kebutuhan untuk penyimpanan pada Gudang data yang akan dibangun ke *database server (data warehouse)*. Proses data Transformasi dengan file name data fakta memiliki beberapa *fields* yang akan di proses pada data loading. Pada proses loading data fakta, *file name* proses *connection database server* dengan nama output file *dim_fact_feeder*. Tahapan pada *design* pilih *file csv input* dan diletakan pada lembar kerja *transformation*. Step berikutnya manampilkan pada lembar kerja *transformation* untuk *dimension lookup/update* untuk dimensi area, dimensi gardu induk, dimensi feeder, dimensi waktu dan *output* pada *table output* untuk fakta *feeder* serta dihubungkan dengan *hop*. *Output* fakta *feeder* yang pada *data warehouse* berfungsi untuk menghubungkan dimensi satu dengan dimensi lainnya. Gambar 4.10 merupakan proses loading setelah di transformasikan.



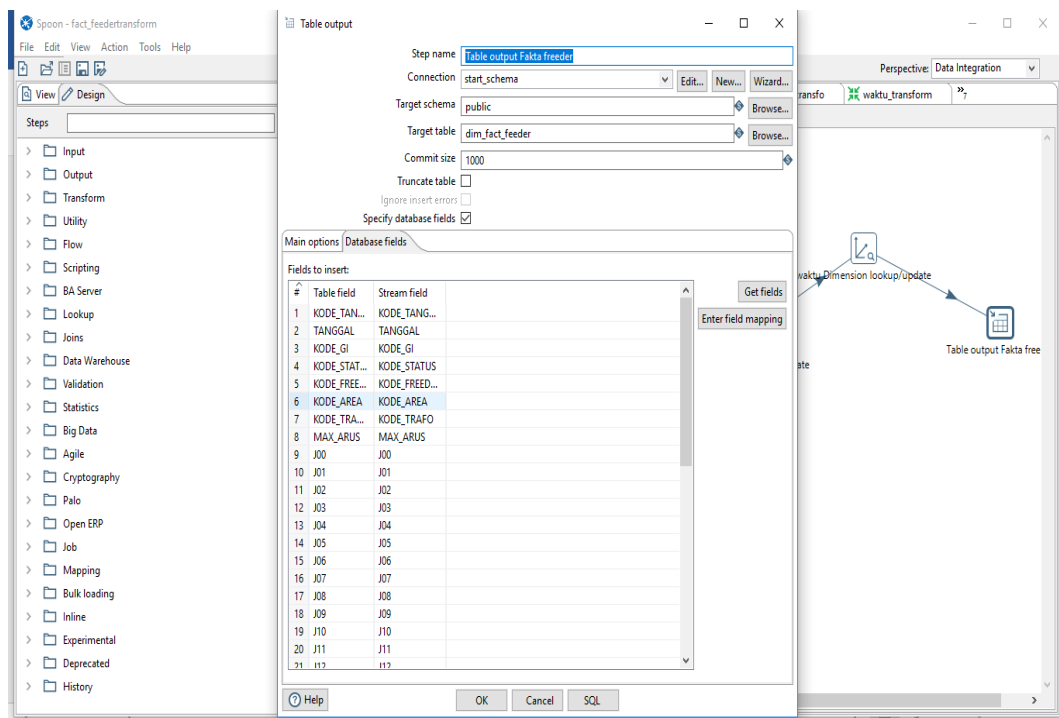
Gambar 4.10 Proses *Data Loading*

Pengaturan pada step *file cvs input* dengan mengambil data cvs pada *browse* yang telah di transformasikan, seperti pengaturan pada proses sebelumnya, pengaturan ini mengubah pada *type*, format dan *length* untuk panjang karakter, Gambar 4.11 tahapan proses pengaturan *file cvs input* untuk *file fakta feeder*.



Gambar 4.11 Pengaturan *loading proses file csv input*

File fakta feeder di proses ke *data warehouse* dengan *connection database start_schema* memiliki fungsi sebagai perhitungan beban total distribusi *feeder*, Tabelnya perhitungan terdiri dari *total_beban*, *kwh*, *max*, *max_siang*, *max_malam* dan *rata_rata*. Proses tabel *output* dengan *file name dim_fact_feeder*, *target schema public* dan dengan *get fields* seperti data *fieds* pada *file csv*. Pada perancangan selanjutnya proses loading untuk fakta *feeder* dapat menampilkan dengan satu dimensi, dua dimensi atau empat dimensi. Fakta *feeder* didesain sebagai *database* utama pada menentukan *cube* dimensi yang di rancang dengan hasil *output* yang diinginkan. Proses loading untuk pengaturan tabel *output* fakta *feeder* dapat di lihat pada gambar 4.12. hasil *run* membutuhkan waktu cukup lama dari proses *loading* untuk menampilkan pada *data warehouse*. Pada tahap ini merupakan tahap akhir pada *ETL tools*, serta berfungsi sebagai *input* data untuk proses data histori selanjutnya. *Fields* data olahan pada *file csv* di transformasikan harus sesuai dengan *file csv input* fakta feeder pada proses *loading* hanya tinggal *run* untuk hasil *output* pada *data warehouse*.



Gambar 4.12 Pengaturan *loading* tabel *output* fakta *feeder*

4.1.5 Penyimpanan pada *Data Warehouse*

Proses *transformation* dengan di input dari data *extraction excel format csv*, menghasilkan data penyimpanan pada *database server* yang disebut Gudang data atau *data warehouse*. Pada penyimpanan *data warehouse*, pengolahan data selanjutnya hanya mengoneksi proses pada *data source* yang ada di *database server*. Proses *loading*, perancangan *start schema* untuk menghasilkan dimensi-dimensi yang dirancang dengan data yang ada di *database server*, begitu juga dengan *report* hasil pelaporan juga mengambil data yang ada pada *database server*. Berikut gambar 4.13 yang merupakan penyimpanan pada *data warehouse*.

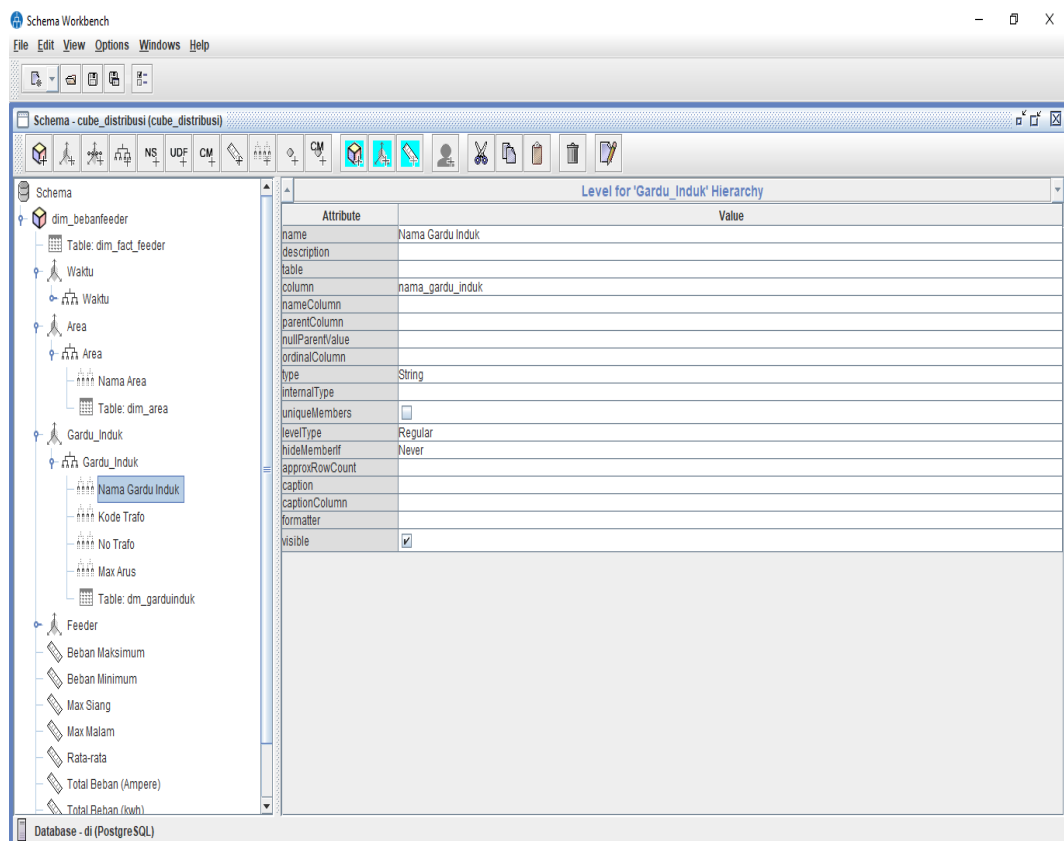
The screenshot shows a PostgreSQL database window titled 'Edit Data - postgres (localhost:5432) - di - public.bebanja_dim'. The table contains 31 rows of data. The columns are: tanggal, nama_penyulang, id_traf, no_traf, max_arus, and J00 through J10. The data is as follows:

	tanggal	nama_penyulang	id_traf	no_traf	max_arus	J00	J01	J02	J03	J04	J05	J06	J07	J08	J09	J10	J11
1																	
2	2016-01-03 00:00:00	(AYODYA	1221	TRAF0	1 300	5	5	5	5	5	5	5	5	5	5	5	5
3	2016-01-25 00:00:00	(BOHIR	1333	TRAF0	3 300	10	10	10	10	10	10	10	10	10	10	10	10
4	2016-01-03 00:00:00	(MACHO	1221	TRAF0	1 300	0	0	0	0	0	0	0	0	0	0	0	0
5	2016-01-03 00:00:00	(GAJAH MADA	1221	TRAF0	1 300	40	40	40	40	40	40	40	40	40	45	45	45
6	2016-01-25 00:00:00	(MODAL	1333	TRAF0	3 300	50	45	45	45	45	55	65	75	85	90	105	1
7	2016-01-03 00:00:00	(PANDAWA	1221	TRAF0	1 300	55	55	55	55	55	55	55	55	55	60	60	60
8	2016-01-03 00:00:00	(TONIL	1221	TRAF0	1 300	15	15	15	15	15	15	15	20	20	30	35	40
9	2016-01-25 00:00:00	(USAHA	1333	TRAF0	3 300	35	30	30	30	30	70	115	135	145	155	150	1
10	2016-01-03 00:00:00	(KUTAI	1221	TRAF0	1 300	10	10	10	10	10	10	10	20	20	20	20	20
11	2016-01-03 00:00:00	(HAYAMURUK	1221	TRAF0	1 300	0	0	0	0	0	0	0	0	0	0	0	0
12	2016-01-25 00:00:00	(PASAR	1333	TRAF0	3 300	10	5	5	5	5	10	20	25	30	30	30	30
13	2016-01-03 00:00:00	(BITI	1221	TRAF0	1 300	35	35	35	35	35	45	55	55	55	75	95	90
14	2016-01-03 00:00:00	(SANGKURIANG	1221	TRAF0	1 300	0	0	0	0	0	0	0	0	0	0	0	0
15	2016-01-25 00:00:00	(KONGSI	1333	TRAF0	3 300	55	50	50	50	50	50	50	55	60	55	60	60
16	2016-01-03 00:00:00	(PETRUK	1222	TRAF0	2 300	5	5	5	5	5	5	5	5	5	10	10	10
17	2016-01-03 00:00:00	(LUDRUK	1222	TRAF0	2 300	35	35	35	35	35	35	35	40	40	40	40	40
18	2016-01-25 00:00:00	(KOMPENI	1333	TRAF0	3 300	35	30	30	30	30	40	55	60	60	65	65	60
19	2016-01-03 00:00:00	(MAJAPAHIT	1222	TRAF0	2 300	20	20	20	20	20	20	20	20	20	25	25	20
20	2016-01-03 00:00:00	(HASANUDIN	1222	TRAF0	2 300	55	55	55	55	55	55	55	60	60	65	65	60
21	2016-01-25 00:00:00	(VENDOR	1333	TRAF0	3 300	5	5	5	5	5	10	20	25	35	40	40	20
22	2016-01-03 00:00:00	(ISTANA	1222	TRAF0	2 300	5	5	5	5	5	5	5	5	5	5	5	5
23	2016-01-03 00:00:00	(JAMPANG2	1222	TRAF0	2 300	5	5	5	5	5	5	5	5	5	5	5	5
24	2016-01-25 00:00:00	(SAKURA	1343	TRAF0	3 300	105	100	95	90	85	80	70	80	95	105	135	10
25	2016-01-03 00:00:00	(JAMPANG1	1222	TRAF0	2 300	10	10	10	10	10	10	10	10	10	10	10	10
26	2016-01-03 00:00:00	(BALADENA	1222	TRAF0	2 300	15	15	15	15	15	15	15	20	30	30	30	30
27	2016-01-25 00:00:00	(ADENIUM	1342	TRAF0	2 300	15	15	15	10	10	10	10	15	15	20	25	20
28	2016-01-03 00:00:00	(JAMPANG	1222	TRAF0	2 300	15	15	15	15	15	15	15	15	15	15	15	10
29	2016-01-03 00:00:00	(PAJAJARAN	1222	TRAF0	2 300	0	0	0	0	0	0	0	0	0	0	0	0
30	2016-01-25 00:00:00	(ARBEBY	1342	TRAF0	2 300	125	120	115	110	105	100	95	90	90	95	95	90
31	2016-01-03 00:00:00	(HANOMAN	1223	TRAF0	3 300	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 4.13 Penyimpanan pada *data warehouse*

4.1.6 Model Perancangan *Start Schema* pada OLAP

Dalam membangun *data warehouse* untuk analisa distribusi beban *feeder* PT. PLN (Persero) Disjaya menggunakan model data yang dirancang pada model *start schema*, model untuk merancang *cube* OLAP untuk menganalisa total beban distribusi berdasarkan dimensi yang sudah di rancang. Proses model *start schema* di rancang pada aplikasi *tools schema workbench*. Berikut Gambar 4.14 proses perancangan *cube* OLAP dengan menampilkan rancangan *schema* dengan *cube* name *dim_bebanfeeder* pada *table* data yang diambil pada *data warehouse* setelah proses *loading* tabel fakta *feeder*. Desain dimensi dirancang dari dimensi waktu, dimensi area, dimensi gardu induk dan dimensi *feeder*, hasil perhitungan pada perancangan *measure* pada perhitungan beban maksimum, beban minimum rata-rata dan lain sebagainya, semua proses dapat di lihat pada gambar di bawah ini.



Gambar 4.14 Perancangan model *start schema* OLAP

4.2 Pembahasan

Berdasarkan pemodelan dan perancangan sistem yang telah dilakukan sebelumnya, selanjutnya dilakukan pembangunan sistem dan menerapkan metode OLAP ke dalam sistem dengan menggunakan bahan dan alat (data, perangkat keras, perangkat lunak, dan Bahasa pemrograman JAVA dan SQL) yang telah disiapkan.

4.2.1 Halaman Login

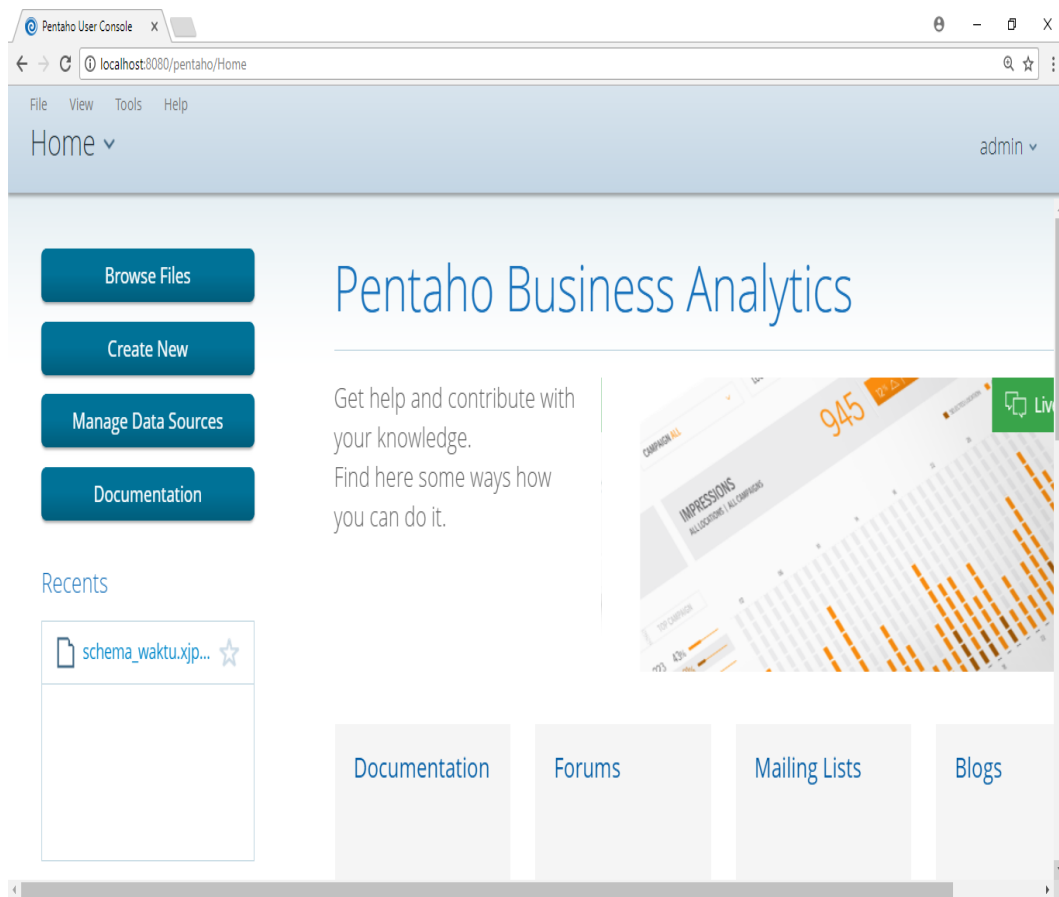
Halaman ini merupakan antarmuka yang digunakan untuk membatasi akses ke dalam sistem dan memberikan keamanan pada sistem informasi, sehingga data dan informasi dapat terlindungi dari pihak-pihak yang tidak memiliki kewenangan. Halaman login ditunjukkan pada Gambar 4.15.



Gambar 4.15 Halaman *login*

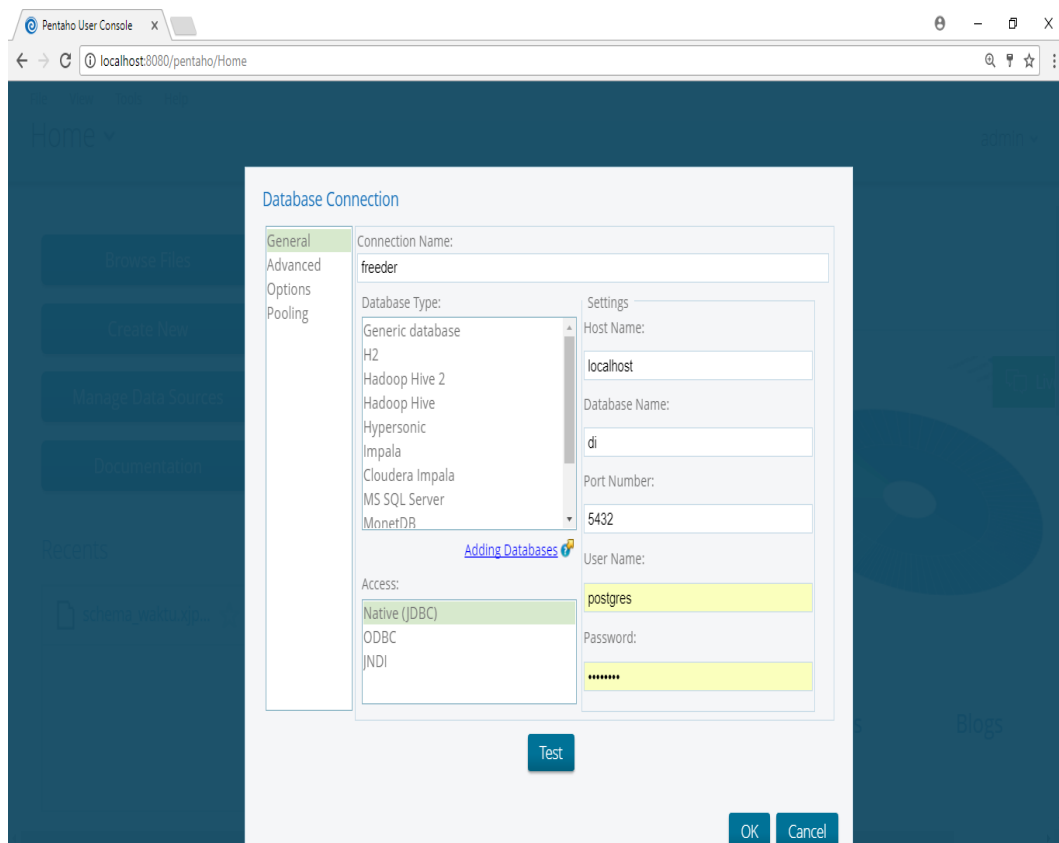
4.2.2 Halaman Utama Analisa OLAP

Sistem akan memasuki *pentaho business analytics*, yang pada penyajian halaman ini untuk membuat proses analisa baru pada skema dimensi untuk menghasilkan pada proses *jpivot* dan *dashboard*. Analisa yang sudah di buat dapat disimpan, dan dapat dibuka pada *browse files*. *create new* disini untuk membuat baru data analisa bedasarkan dimensi skema dengan hasil *output* berupa data dan grafik. *Manage data source* untuk mengkoneksikan *web server* dengan *database server localhost*, yang disini peneliti menggunakan *postgresql* sebagai *databases conecction*, tampilan lembar utama pada aplikasi pentaho ini dapat di lihat pada gambar 4.16.



Gambar 4.16 Halaman utama analisa OLAP

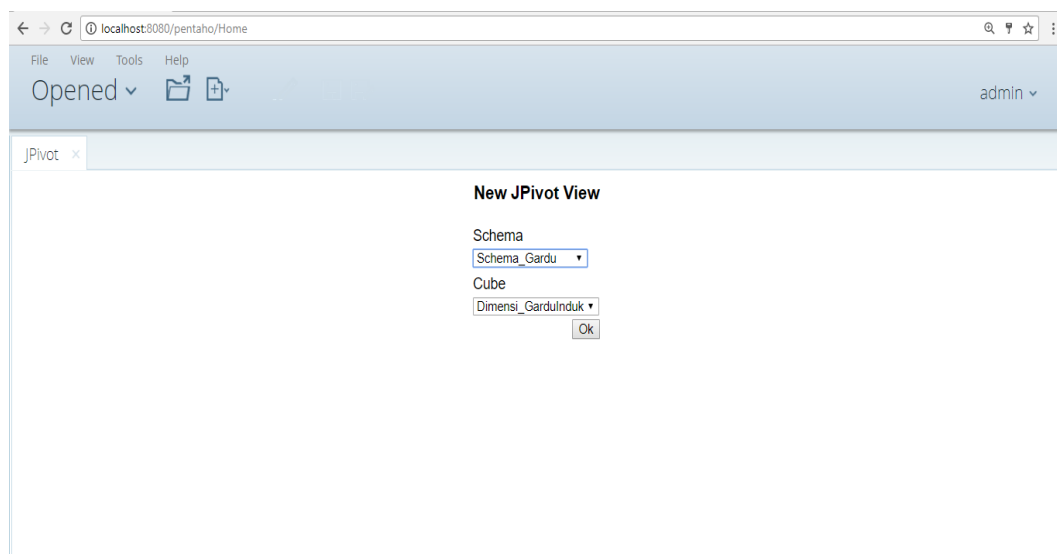
Analisa dengan aplikasi berbasis java pentaho, dimulai dengan mengoneksikan *web server* pentaho dengan *database server* yang pada aplikasi ini yakni menggunakan *postgresql* yang berbahasa pemrograman *Sql*. Koneksi ini untuk memanggil data yang tersimpan pada *databases server* yang telah tersimpan sebagai Gudang data atau *data warehouse*. *Database server* yang dari proses ETL telah di rancang *start schema* sebagai perancangan OLAP untuk dimensi analisa untuk perhitungan beban pengguna energi listrik pada *feeder*. Koneksi dengan *database server* harus diuji dengan 'test' dengan tampilan keterangan oke untuk dapat memastikan web ini dapat mengambil data pada *postgresql*. *Database connection* pada aplikasi pentaho dapat di lihat pada gambar 4.17.



Gambar 4.17 Tampilan *database connection*

4.2.3 Hasil Tampilan *Start Schema*

Database connection telah berhasil, sehingga dapat kita menganalisa dengan data di *data warehouse* sesuai dengan dimensi yang telah dibuat. Dimensi ini akan menampilkan nama *schema* dan *cube* untuk memilih dimensi yang sudah dibuat. Sebagai contoh telah dibuat dimensi dengan nama *Schema Area* dan *Cube Distribusi_Area*, sehingga tampilan *schema* pada *start schema* dengan nama Area dan dimensi yang akan di analisa berdasarkan area dapat dilihat pada Gambar 4.18



Gambar 4.18 Tampilan *schema* dan *cube* dimensi

Laporan *output* menyajikan tampilan data dan perhitungan jumlah beban distribusi pada dimensi yang dituju. Hasil yang diatur disesuaikan dengan grup bulan untuk dimensi waktu, nama area untuk dimensi area, dan seterusnya. Hasil *output* analisa terbagi menjadi dua, analisa tampilan data berdasarkan *slice* dan data berdasarkan *drill down*. Data berdasarkan *slice* merupakan tampilan yang disesusikan dengan grup data tampilan secara *cube*. Sedangkan data berdasarkan *drill down* merupakan tampilan data yang di turunkan datanya sesuai dengan subgroup yang dipilih dari grup yang akan di proses. pada gambar 4.19 hasil *output* berdasarkan jumlah data dimensi dengan analisa *slice*. Dan pada gambar 4.20 tampilan hasil *output* berdasarkan jumlah data dimensi berdasarkan Analisa *drill down*.

Area_Distribusi	Time	Measures
All Area.Area_Distribusi	All Waktu.Times	83,139,980
BANDENGAN	All Waktu.Times	9,977,095
	2016	9,977,095
	1	9,977,095
	January	9,977,095
	Friday	1,629,509
	2016-01-01 00:00:00.0	242,200
	2016-01-08 00:00:00.0	352,625
	2016-01-15 00:00:00.0	345,720
	2016-01-22 00:00:00.0	339,148
	2016-01-29 00:00:00.0	349,816
	Monday	1,331,406
	Saturday	1,502,258
	Sunday	1,358,669
	Thursday	1,394,190
	Tuesday	1,374,731
	Wednesday	1,386,332
BANTEN	All Waktu.Times	308,192
BEKASI	All Waktu.Times	626,471
BINTARO	All Waktu.Times	5,016,230

Gambar 4.19 Hasil analisa *Slice*

Tahun	Semester	Nama Bulan	Mingguan	Tanggal	Hari	Tanggal Libur	Nama Area	Nama Gardu Induk	Kode Trafo	No Trafo	Max Arus	Nama Feeder	Status Feeder	J00	J01	J02	J03	J04	J05	J06
2,016.00	1.00	February	6.00	02/01/2016	Monday		CENGGKARENG	CILEDUG	1,121.00	TRAF01	300.00	BANDO	BERBEBAN	90.00	90.00	90.00	85.00	85.00	80.00	90.00
2,016.00	1.00	February	6.00	02/01/2016	Monday		CENGGKARENG	CILEDUG	1,121.00	TRAF01	300.00	BEDAK	BERBEBAN	80.00	80.00	80.00	80.00	80.00	75.00	75.00
2,016.00	1.00	February	6.00	02/01/2016	Monday		CENGGKARENG	CILEDUG	1,121.00	TRAF01	300.00	BOING	EXPRESS	65.00	60.00	10.00	10.00	10.00	10.00	10.00
2,016.00	1.00	February	6.00	02/01/2016	Monday		CENGGKARENG	CILEDUG	1,121.00	TRAF01	300.00	BOMBER	BERBEBAN	60.00	60.00	55.00	55.00	60.00	60.00	60.00
2,016.00	1.00	February	6.00	02/01/2016	Monday		CENGGKARENG	CILEDUG	1,121.00	TRAF01	300.00	CASSA	BERBEBAN	50.00	50.00	50.00	50.00	100.00	130.00	135.00
2,016.00	1.00	February	6.00	02/01/2016	Monday		CENGGKARENG	CILEDUG	1,121.00	TRAF01	300.00	CELAK	BERBEBAN	60.00	60.00	60.00	60.00	55.00	55.00	50.00
2,016.00	1.00	February	6.00	02/01/2016	Monday		CENGGKARENG	CILEDUG	1,121.00	TRAF01	300.00	CESNA	BERBEBAN	135.00	130.00	125.00	125.00	120.00	120.00	115.00
2,016.00	1.00	February	6.00	02/01/2016	Monday		CENGGKARENG	CILEDUG	1,121.00	TRAF01	300.00	CITILINK	BERBEBAN	5.00	5.00	5.00	5.00	5.00	5.00	5.00

Gambar 4.20 Hasil analisa *drill down*

4.2.4 Hasil Analisa Dimensi Area

Analisa distribusi beban daya listrik yang pertama diproses dengan analisa dimesi Area, area distribusi *feeder* yang tersebar di DKI Jakarta dan sekitarnya ada 25 area. Sehingga area diluar Jakarta juga termasuk di dalamnya, seperti Serpong, Ciputat, Banten dan Bekasi, Karena Area distribusi tidak mencangku satu area saja. Perhitungan Analisa dimensi Area berdasarkan Total beban dengan satuan Ampere dan rata-rata beban, diperoleh Area distribusi Menteng memiliki total beban terbesar dengan Total beban pada kwh sebesar 6.247.495.919,01 kw dan dengan rata-rata terbesar pada Area Serpong sebesar 3.594,002 kw. Hasil secara data slice dapat dilihat pada gambar 4.21.

Area	Measures		
	• Beban Maksimum	Total Beban (Kwh)	• Rata-rata
MENTENG	14,964.48	6,247,495,919.46	1,411.063
BANDENGAN	34,605.36	4,340,370,386.11	1,802.884
BULUNGAN	14,340.96	3,775,189,820.01	1,722.106
CENKARENG	9,820.44	2,711,568,608.18	2,420.42
CEMPAKA PUTIH	15,588	2,483,490,223.77	1,724.98
BINTARO	45,361.08	2,289,572,579.49	2,302.215
MARUNDA	17,458.56	2,273,098,993.9	1,490.226
TANJUNG PRIOK	11,067.48	1,890,419,798.1	1,622.007
PONDOK KOPI	9,352.8	1,771,816,785.21	2,260.363
LENTENG AGUNG	9,352.8	1,613,623,027.19	2,037.264
PONDOK GEDE	9,352.8	1,479,437,439.04	2,332.13
KEBON JERUK	23,537.88	1,361,927,622.43	2,347.882
CIRACAS	9,352.8	1,332,200,361.58	2,345.451
CIPUTAT	9,352.8	1,330,562,904.65	2,234.046
JATINEGARA	38,034.72	846,496,862.39	1,624.267
KRAMAT JATI	9,352.8	787,720,718.48	2,015.668
DEPOK	8,885.16	168,599,465.04	2,524.024
BEKASI	8,885.16	77,064,453.22	1,061.514
TELUK NAGA	26,811.36	64,834,513.72	1,704.394
BANTEN	8,105.76	38,837,813.76	3,343.499
CIKOKOL	7,014.6	21,102,192.65	1,446.168
SERPONG	5,455.8	2,397,902.04	3,594.002
CILEDUG	2,494.08	56,428.56	78.376
APD	93.53	93.53	0.009

Slicer:

Gambar 4.21 Hasil Analisa Dimensi Area tampilan Slice

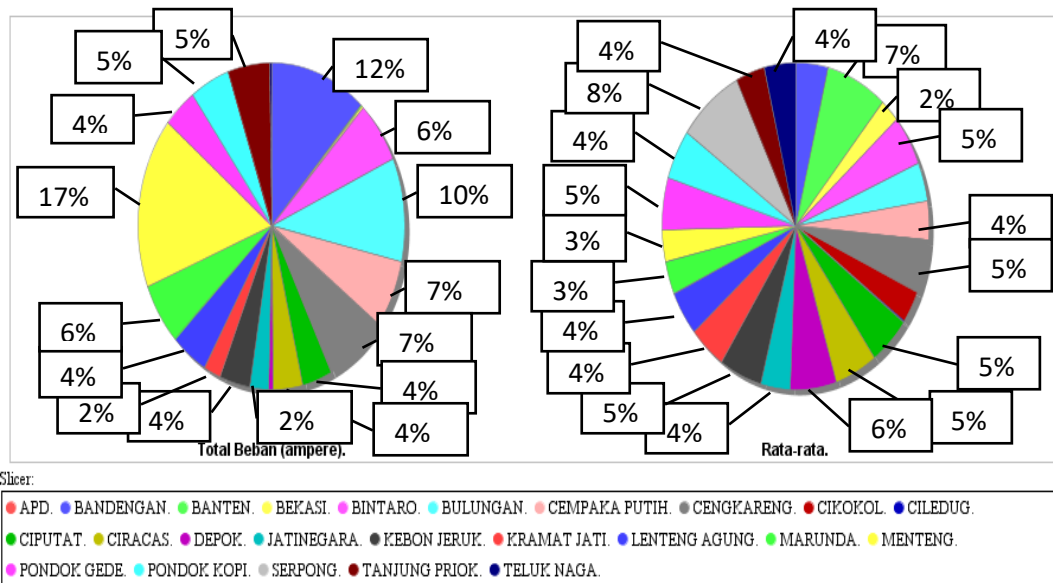
Analisa berikut tampilan data Analisa berdasarkan tampilan data secara drill down. Pada Analisa dimensi Area, sebagai contoh Area Kebon jeruk. Pada urutan dimensi Area pada Nama_Area, dengan sub gurpnya pada Kode Trafo, No. trafo dan Total beban maksimum. Hasil Analisa dice diurutkan pada Area Kebon jeruk dengan list kode trafo dan no trafo yang tertera pada gambar 4.22. tampilan hasil berikut merupakan hasil halaman pertama dengan 18 nomor rincian data dari Analisa data keseluruhan.

• Nama_Area	• Kode_Trafo	• No_Trafo	▼ Beban Maksimum
KEBON JERUK	1,611.00	TRAFO 1	23,537.88
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80
KEBON JERUK	1,611.00	TRAFO 1	9,352.80

Page 1/1,361 Goto Page 1 Rows/page 18

Gambar 4.22 Hasil Analisa dimensi area tampilan *Drill down*

Gambar 4.23 merupakan hasil grafik dari hasil dimensi Area yang di hitung berdasarkan Total Beban (Kwh) dan rata-rata beban. Sesuai dengan perhitungan data Analisa, area Menteng memiliki total beban terbesar dengan preserntase 17% dan Serpong memiliki rata-rata terbesar dengan presentase 8%. Pada rata-rata dimensi Area memiliki *range* presentase sebesar 4% - 6%.



Gambar 4.23 Grafik Hasil Analisa Dimensi Area

4.2.5 Hasil Analisa Dimensi Gardu Induk

Gardu induk untuk distribusi *feeder* beban daya arus listrik pada DKI Jakarta dan sekitarnya pada hasil Analisa sekitar ada 55 Gardu Induk yang tersebar. Sehingga satu gardu induk memiliki penyulang *feeder* berkisar 30 *feeder*. Pada gardu induk ini merupakan tempat operator dalam perhitungan manual untuk besar beban yang dikeluarkan pada setiap *feeder* dengan penginputan selama satu jam sekali. Perhitungan Analisa dimensi Gardu induk berdasarkan Total beban dengan satuan Ampere dan rata-rata beban, diperoleh pada Gardu induk Duri Kosambi memiliki total beban terbesar dengan Total beban kwh sebesar 1.346.253.847,65 kw, dengan rata-rata terbesar pada gardu induk Pondok Kelapa sebesar 2.837.195 kw, dan beban maksimum pada gardu induk cileduk sebesar 45,361 kw. Hasil secara data dari dimensi gardu induk dapat dilihat pada gambar 4.24.

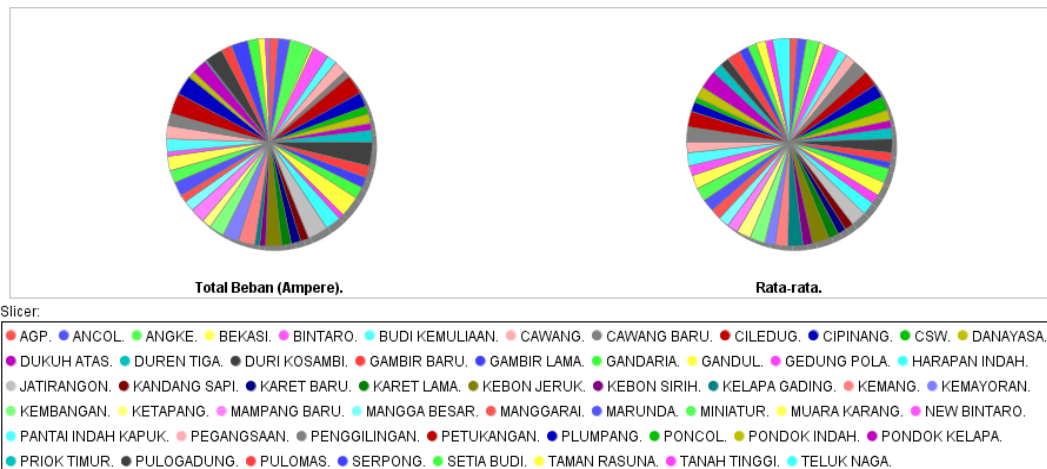
Gardu Induk	Measures		
	● Beban Maksimum	▼ Total Beban (kwh)	● Rata-rata
+ DURI KOSAMBI	9,352.8	1,346,253,847.65	2,100.957
+ ANGKE	9,352.8	1,253,948,040.5	2,110.601
+ GANDUL	9,352.8	1,211,192,681.86	2,232.2
+ JATIRANGON	9,352.8	1,205,145,940.47	2,499.324
+ PETUKANGAN	9,352.8	1,187,606,229.46	2,462.888
+ CILEDUG	45,361.08	1,135,726,341.43	2,359.099
+ PLUMPANG	11,067.48	1,108,957,193.9	1,488.492
+ PULOGADUNG	9,352.8	1,055,675,439.47	1,467.213
+ SERPONG	10,443.96	955,002,998.71	1,480.699
+ KEMANG	9,352.8	950,184,528.62	1,998.105
+ KEMAYORAN	9,352.8	913,807,291.01	1,660.039
+ BINTARO	9,352.8	913,080,568.08	2,348.478
+ HARAPAN INDAH	9,352.8	892,005,186.81	2,009.101
+ KEBON JERUK	9,352.8	887,922,003.7	2,652.895
+ KEMBANGAN	23,537.88	886,003,816.31	2,466.836
+ PONDOK KELAPA	9,352.8	883,283,046.05	2,837.195
+ MAMPANG BARU	9,352.8	855,386,106.62	1,748.353
+ CIPINANG	9,352.8	816,665,046.89	2,201.886
+ MARUNDA	17,146.8	812,934,308.51	1,984.186
+ MUARA KARANG	9,352.8	808,424,294.85	2,077.96
+ PEGANGSAAN	17,458.56	762,627,841.94	1,642.511
+ PANTAI INDAH KAPUK	34,605.36	739,686,421.24	2,056.201
+ PENGGILINGAN	9,352.8	717,994,191.61	2,354.702
+ MINIATUR	9,352.8	717,148,916.71	2,098.921
+ CAWANG	38,034.72	700,470,037.08	1,619.456
+ DUREN TIGA	9,352.8	700,372,955.26	1,684.378
+ GAMBIR BARU	14,964.48	690,090,455.88	1,455.082
+ ANCOL	9,352.8	656,955,262.32	1,379.758
+ GANDARIA	9,352.8	655,204,885.87	2,192.407
+ PULOMAS	9,352.8	621,252,943.68	2,054.117
+ MANGGA BESAR	15,588	616,001,627.05	1,689.134
+ GAMBIR LAMA	9,352.8	597,732,927.55	1,038.75
+ SETIA BUDI	9,352.8	597,471,797.33	1,348.167
+ KARET BARU	9,352.8	582,378,092.79	1,294.501
+ KARET LAMA	9,352.8	577,075,554.03	1,553.336
+ BUDI KEMULIAAN	9,352.8	572,299,297.41	1,549.798
+ DANAYASA	14,340.96	562,037,841.66	1,671.059
+ KETAPANG	9,352.8	556,413,117.55	2,235.751
+ AGP	9,352.8	532,447,608.82	1,261.371
+ CSW	9,352.8	464,797,403.49	2,145.15
+ MANGGARAI	15,588	441,438,161.97	1,651.143

+ KANDANG SAPI	9,352.8	438,982,584.32	1,376.865
+ TAMAN RASUNA	9,352.8	414,240,749.59	1,597.201
+ DUKUH ATAS	9,352.8	382,402,290.72	1,216.322
+ PONDOK INDAH	9,352.8	376,410,419.42	1,845.27
+ CAWANG BARU	9,352.8	362,479,423.86	2,367.64
+ KEBON SIRIH	9,352.8	351,989,791.27	1,546.896
+ GEDUNG POLA	9,352.8	324,229,319.71	1,494.433
+ KELAPA GADING	9,352.8	302,036,891.49	2,331.027
+ NEW BINTARO	35,696.52	250,652,701.83	1,635.149
+ TANAH TINGGI	9,352.8	241,528,546.6	1,131.607
+ BEKASI	9,352.8	200,534,475.99	714.843
+ PRIOK TIMUR	9,352.8	75,504,063.24	1,667.311
+ PONCOL	9,352.8	32,906,049.77	752.132
+ TELUK NAGA	9,352.8	12,885,352.56	2,601.488

Slicer:

Gambar 4.24 Hasil Analisa Dimensi Gardu Induk

Hasil *output* pada dimensi gardu induk berdasarkan Total Beban (Ampere) dan rata-rata beban. Dengan perhitungan data Analisa, dimensi gardu induk dengan total beban memperoleh presentase dengan kisaran 2% - 4%, pada total beban terbesar pada gardu induk Duri Kosambi memiliki 4% dari total beban keseluruhan. Dan Pada rata-rata dimensi gardu induk memiliki *range* presentase sebesar 2% dari keseluruhan rata-rata beban pada gardu induk. Hasil grafik dimensi gardu induk dapat di lihat pada gambar 4.25.



Gambar 4.25 Grafik Hasil Analisa Dimensi Gardu Induk

4.2.6 Hasil Analisa Dimensi Feeder

Jumlah *feeder* pada Gardu induk untuk distribusi *feeder* beban daya arus listrik pada DKI Jakarta dan sekitarnya pada hasil Analisa sekitar ada 1.906 *feeder* yang tersebar. Sehingga penyulang/*feeder* berkisar 30 *feeder* dimiliki oleh satu gardu induk. Perhitungan Analisa berdasarkan dimensi *feeder* dengan perhitungan Total beban satuan Ampere dan rata-rata beban, diperoleh pada *feeder* Cosmos yang memiliki total beban terbesar dengan Total beban kwh sebesar 80.658.516,01 kw, pada *feeder* pacifika beban max siang diperoleh dengan beban 440 kw, max malam nya sebesar 460 kw, dengan rata-rata terbesar pada *feeder* Budidarmas sebesar 6.323,648 kw pada feeder yang sama budidarmas memiliki beban maksimum terbesar sebesar 17.146,8 kw. Hasil secara data dari dimensi gardu induk dapat dilihat pada gambar 4.26.

Feeder	Measures		
	Rata-rata	Beban Maximum	Total Beban (kwh)
COSMOS	3,040.515	9,352.8	80,658,516.01
GURITA	3,708.062	9,352.8	69,194,539.65
BUDIDARMA6	6,323.648	17,146.8	68,349,171.24
PAJERO	5,544.024	15,588	60,077,461.39
CASABLANCA3-4	5,449.339	15,588	59,275,770.56
MENTOK	3,072.247	9,352.8	57,480,656.47
NINJA	3,514.502	9,352.8	51,971,483.15
MIDO	4,831.077	9,352.8	51,938,156.02
MAKELA	4,809.247	9,352.8	51,808,775.61
PHOTO	4,752.609	9,352.8	51,241,247.72
GETUK	3,971.156	9,352.8	51,067,628.58
PUKAT	2,705.229	9,352.8	50,569,342.57
KUTILANG	2,901.545	9,352.8	50,102,419.61
HEKTAR	4,608.617	9,352.8	49,602,574.8
GETUK	3,971.156	9,352.8	51,067,628.58
PUKAT	2,705.229	9,352.8	50,569,342.57
KUTILANG	2,901.545	9,352.8	50,102,419.61
HEKTAR	4,608.617	9,352.8	49,602,574.8
GETUK	3,971.156	9,352.8	51,067,628.58
PUKAT	2,705.229	9,352.8	50,569,342.57
KUTILANG	2,901.545	9,352.8	50,102,419.61
HEKTAR	4,608.617	9,352.8	49,602,574.8
TEKUKUR	4,565.997	9,352.8	49,183,818.77
FORMIKA	2,525.338	9,352.8	49,127,296.68
ANYAM	4,549.012	9,352.8	49,015,717.78
NOVEL	2,622.727	9,352.8	48,866,634.14
SIAGA	4,506.217	9,352.8	48,553,720.62
SASAK	4,503.12	9,352.8	48,526,815.75
PACIFIK1	4,423.598	14,340.96	48,091,068.79
DAHLIA	4,447.111	9,352.8	47,966,614.2
RAPORT	2,571.295	9,352.8	47,879,664.36
NAHKODA	4,425.02	9,352.8	47,805,995.45
INAI	4,413.637	9,352.8	47,650,427.21
TERAS	4,396.411	9,352.8	47,379,071.3

MARTIL	4,289.323	9,352.8	46,214,523
STRIKER	4,272.715	9,352.8	46,147,369.9
GAZEBO	4,275.662	9,352.8	46,067,590.51
SEIKO	4,268.63	9,352.8	46,040,560.92
MAYOR	4,265.322	9,352.8	45,940,392.43
TULANG	4,257.888	9,352.8	45,870,838.77
MADRID	4,243.85	9,352.8	45,817,901.93
BEIMO	4,242.633	9,352.8	45,664,609.54
PUPUR	4,237.584	9,352.8	45,588,664.8
SINAR	2,001.189	9,352.8	45,530,646.27
PETAI	4,218.701	9,352.8	45,513,374.76
HIAS	4,205.962	9,352.8	45,353,348.35
AMPUL	4,184.323	9,352.8	45,123,207.12
GOBI	4,183.588	9,352.8	45,092,155.82
SAKERAH	4,172.207	9,352.8	45,075,788.42
PANCA	4,171.432	9,352.8	45,020,419.85
KURAWA	2,418.496	9,352.8	44,943,913.97
BELGIA	4,173.913	9,352.8	44,927,453.02
MELINJO	4,156.518	9,352.8	44,897,835.82
SENDANG	4,163.489	9,352.8	44,835,109.7
PILAR	4,145.729	9,352.8	44,695,503.58
GURAU	4,139.692	9,352.8	44,660,555.28
RINJANI	4,150.174	9,352.8	44,620,743.53
PATIN	4,132.931	9,352.8	44,547,542.28
PELDA	4,131.788	9,352.8	44,522,695
LANTAI	4,123.925	9,352.8	44,402,916.81
AULA	4,119.766	9,352.8	44,377,165.44
CENTEX4	4,116.443	9,352.8	44,361,670.97
SUSU	4,088.854	9,352.8	44,081,523.43
NIAGARA	4,081.807	9,352.8	44,007,168.68
ROCKER	4,081.253	9,352.8	44,005,235.76

Gambar 4.26 Hasil analisa dimensi *feeder*

4.2.7 Hasil Analisa Dimensi Waktu

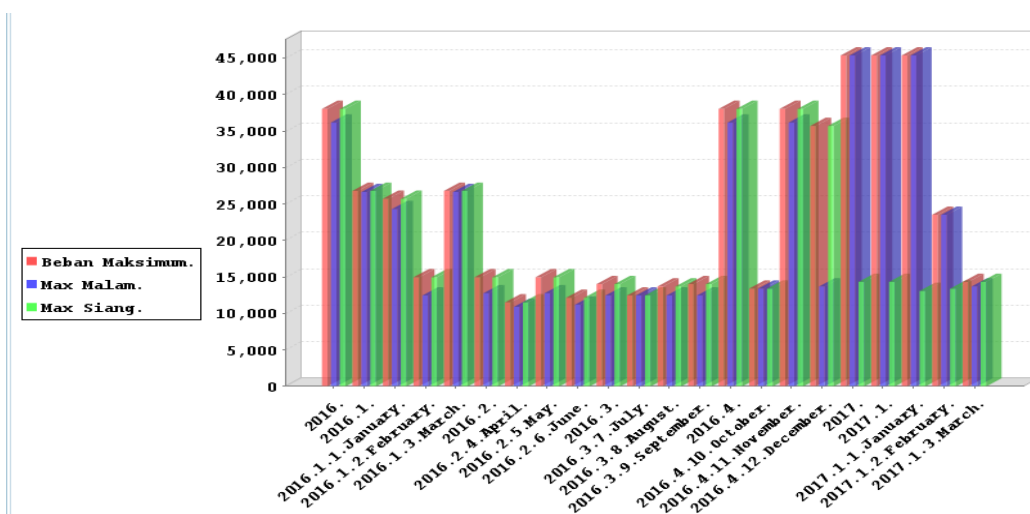
Analisa dimensi waktu distribusi beban daya listrik yang dianalisa dengan data selama 15 bulan terhitung pada Januari 2016 – Maret 2017, dan menganalisa distribusi *feeder* untuk daerah khusus DKI Jakarta dan sekitarnya. Pada sistem Analisa pentaho dapat diperoleh hasil Analisa dengan distribusi beban terbesar pada bulan April 2016 dengan total beban kwh sebesar 2.625.598.263,28 kw, dengan rata-rata pada bulan yang sama sekitar 1.916,161, dan beban maksimum terbesar pada januari 2017 sebesar 45.361,08. Hasil secara data dimensi waktu dapat dilihat pada gambar 4.27.

Waktu	Measures		
	Rata-rata	Beban Maksimum	Total Beban (kwh)
2016	1,835.151	38,034.72	29,685,938,472.1
1	1,860.555	26,811.36	7,607,962,582.58
January	1,875.978	25,720.2	2,591,972,016.59
February	1,835.163	14,964.48	2,394,040,144.42
March	1,868.976	26,811.36	2,621,950,421.57
2	1,901.774	14,964.48	7,651,682,433.46
April	1,916.161	11,535.12	2,625,598,263.28
May	1,900.408	14,964.48	2,619,204,720.3
June	1,888.322	12,158.64	2,406,879,449.88
3	1,782.808	14,029.2	7,036,225,835.75
July	1,639.911	12,470.4	2,164,060,029.94
August	1,856.727	13,717.44	2,477,217,057.54
September	1,852.492	14,029.2	2,394,948,748.27
4	1,796.611	38,034.72	7,390,067,620.31
October	1,809.348	13,405.68	2,427,918,043.64
November	1,831.21	38,034.72	2,502,565,945.24
December	1,750.335	35,696.52	2,459,583,631.43
2017	1,729.621	45,361.08	7,221,946,440.41
1	1,729.621	45,361.08	7,221,946,440.41
January	1,743.791	45,361.08	2,483,347,766.59
February	1,676.993	23,537.88	2,188,019,615.24
March	1,763.14	14,340.96	2,550,579,058.58

Slicer:

Gambar 4.27 Hasil analisa dimensi waktu

Hasil *output* pada dimensi waktu berdasarkan Total Beban (Ampere) dan rata-rata beban. Dengan perhitungan data Analisa, dimensi waktu dengan total beban dan rata-rata memperoleh presentase dengan kisaran 5% - 6%, Analisa perhitungan distribusi beban pada total beban dan rata-rata beban terbesar pada bulan April 2016 dari beban keseluruhan. Hasil grafik dimensi waktu dapat di lihat pada gambar 4.28.



Gambar 4.28 Grafik hasil analisa dimensi waktu

Analisa berikut tampilan data Analisa berdasarkan tampilan data secara drill down. Pada analisa dimensi waktu, sebagai contoh pada bulan Desember 2016. Pada urutan dimensi waktu dengan rincian tabel pada tahun, semester atau new level 1, bulan, mingguan, tanggal, nama hari, tanggal libur, max siang. Hasil analisa drill down diurutkan pada waktu Desember dengan rincian tampilan data yang tertera pada gambar 4.29. tampilan hasil berikut merupakan hasil halaman pertama dengan 18 nomor rincian data dari Analisa data keseluruhan.

Drill Through Table for Max Siang								
Tahun	New Level 1	Bulan	Nama Bulan	Mingguan	Tanggal	Nama Hari	Tanggal Libur	Max Siang
2,016.00	4.00	12.00	December	53.00	12/31/2016	Saturday		35,696.52
2,016.00	4.00	12.00	December	51.00	12/11/2016	Sunday		34,605.36
2,016.00	4.00	12.00	December	51.00	12/12/2016	Monday	12/12/2016	13,249.80
2,016.00	4.00	12.00	December	52.00	12/22/2016	Thursday		13,093.92
2,016.00	4.00	12.00	December	52.00	12/24/2016	Saturday		13,093.92
2,016.00	4.00	12.00	December	50.00	12/08/2016	Thursday		12,782.16
2,016.00	4.00	12.00	December	51.00	12/11/2016	Sunday		12,470.40
2,016.00	4.00	12.00	December	51.00	12/14/2016	Wednesday		12,470.40
2,016.00	4.00	12.00	December	50.00	12/05/2016	Monday		12,158.64
2,016.00	4.00	12.00	December	50.00	12/07/2016	Wednesday		12,158.64
2,016.00	4.00	12.00	December	50.00	12/09/2016	Friday		12,158.64
2,016.00	4.00	12.00	December	52.00	12/19/2016	Monday		12,158.64
2,016.00	4.00	12.00	December	52.00	12/20/2016	Tuesday		12,158.64
2,016.00	4.00	12.00	December	53.00	12/28/2016	Wednesday		12,158.64
2,016.00	4.00	12.00	December	53.00	12/31/2016	Saturday		12,158.64
2,016.00	4.00	12.00	December	50.00	12/04/2016	Sunday		11,846.88
2,016.00	4.00	12.00	December	51.00	12/15/2016	Thursday		11,846.88
2,016.00	4.00	12.00	December	51.00	12/16/2016	Friday		11,846.88

Page 1/3,253 Goto Page 1 Rows/page 18

Gambar 4.29 Hasil analisa dimensi waktu tampilan *Drill down*

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dengan sistem analisa menggunakan OLAP berbasis web diperoleh kesimpulan bahwa Sistem informasi analisis dapat memberikan informasi yang berkaitan dengan beban daya listrik pada satu area provinsi. sistem yang dapat melacak informasi berdasarkan multidimensi dengan dimensi area, dimensi wilayah, dimensi beban daya dan dimensi waktu. sistim infomasi yang di bangun dengan sistem OLAP dengan *tools* pentaho memberikan hasil analisa beban berupa grafik dan data. Dimensi area distribusi Menteng memiliki total beban terbesar dengan Total beban pada kwh sebesar 6.247.495.919,01 kw dengan persentase 17% rata-rata terbesar pada area Serpong sebesar 3.594,002 kw sama dengan 8%, dan kisaran rata-rata dimensi area sebesar 4% - 6%. Dimensi gardu induk Duri Kosambi memiliki total beban terbesar dengan Total beban kwh sebesar 1.346.253.847,65 kw dengan persentase 4%, rata-rata terbesar pada gardu induk Pondok Kelapa sebesar 2.837.195 kw, dan kisaran rata-rata dimensi gardu induk sebesar 2% - 4%. Dimensi *feeder* Cosmos yang memiliki total beban terbesar dengan Total beban kwh sebesar 80.658.516,01 kw. Dimensi waktu beban terbesar pada bulan April 2016 dengan total beban kwh sebesar 2.625.598.263,28 kw, rata-rata pada bulan yang sama sekitar 1.916,161, beban maksimum terbesar pada januari 2017 sebesar 45.361,08, dan kisaran rata-rata dimensi waktu sebesar 5% - 6%.

5.2 Saran

Beberapa saran perbaikan atau pengembangan pada penelitian yang akan dilakukan selanjutnya sebagai berikut:

- a. Untuk mendapatkan informasi yang lebih baik lagi sebagai bahan dalam melakukan analisa distribusi beban penyulang, sebaiknya PT. PLN (Persero)

Disjaya memberikan data lebih rinci pada keterangan gardu induk, area (seperti memberikan data alamat serta peta yang jelas dalam penyulangan). Agar dapat lebih jelas pada penyimpanan *data warehouse*nya.

- b. Terkait dengan data yang sudah ada, sebaiknya PT. PLN (Persero) Disjaya menerapkan standar dalam penerapan beberapa kode dalam database seperti kode gi, kode area, kode penyulangan dan lain-lain. Ini bermanfaat agar proses ETL (*Extract, Transform, Loading*) yang dilakukan dapat berjalan efektif dan efisien.

Daftar Pustaka

- Bouman, R., dan Jos, V. D., 2009, *Pentaho Solution Business Intelligence and Data warehouse With Pentaho and MySql*, Wiley Publishing Inc., United States of America.
- Bimonte, S., Tchounikine, A., Miquel, M., 2007, Spatial OLAP : Open Issues and a Web Based Prototype, *10th AGILE International Conference on Geographic Information Science*, Aalborg University.
- Boulekrouche, B., Jabeur, N., Alimazighi, Z., 2015, An Intelligent ETL Grid-Based Solution to Enable Spatial Data Warehouse Deployment in Cyber Physical System Context, *The 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015)*. Volume 56, 111–118.
- Bukhbinder, G., Krumeraker, M., Phillips, A., 2004, Data Marts, OLAP and Predictive Analytics, *Insurance Industry Decision Support Casualty Actuarial Society Forum*, Winter.
- Curko, K., dan Varga, M., 2008. The Review of the Role of Business Intelligence in Business Engineering, *American Conference On Applied Mathematics (Math '08)* 396-401.
- Hermawan, Y., 2005. *Konsep OLAP dan Aplikasinya Menggunakan Delphi*. Andi Yogyakarta.
- Marsudi, D., 2006. *Operasi Sistem Tenaga Listrik*, Graha Ilmu.
- Meliopoulos, S., Taylor, D., dan Singh, CH., 2005. *Comprehensive Power System Reliability Assessment*, PSRERC Publication 05-13.
- Prada, J. F., 1999, *The Value of Reliability in Power System – Pricing Operating Reserves*, Massachusetts Institute of Tecnology.
- Pradhanaa, H. W., Suryono, S., Widodo, A., 2013, Sistem Pendukung Keputusan Penentuan Lokasi dan Pemetaan Secara Simultan dengan OLAP, *Jurnal Sistem Informasi Bisnis*. volume 3 No. 2.
- Prakash, P., dan Khatod, D. K., 2016, Optimal Sizing and Siting Techniques for Distributed Generation In Distribution Systems : A Review, *Journal Renewable and Sustainable Energy Reviews*. Volume 57, 111-130.

- Ponniah, P., 2001. *Data warehouse Fundamental : A Comprehensive Guide for IT Professionals*. John Wiley & Sons, Inc., NewYork.
- Randy, O. P., 2012, Rancang Bangun Data warehouse untuk Analisis Kinerja Penjualan Pada Industri Dengan Model SPA-DW (Sales Performance Analysis – Data warehouse, *Thesis*, Master in Information System, Universitas Diponegoro.
- Reddy, G. S., Srinivasu, R., Chander Rao, M. P., & Rikkula, S. R., 2010, Data Warehouse, Data Mining, OLAP and OLTP Technologies Are Essential Element to Support Decision-Making Process in Industries. *International Journal on Computer Science and Engineering*, 2865-2873.
- Singh, A., Uphadhyay, D. C., Yadav., Hemant., 2011, The Analytical Data Warehouse : A Sustainable Approach for Empowering Institutional Decision Making, *International Journal of Engineering Science and Technology (IJEST)*. Volume. 3, No. 7.
- Solichan, A., 2012, Rancang Bangun Optimasi Keandalan Sistem Tenaga Listrik Dengan Algoritma Genetika, *Thesis*, Master in Information System, Universitas Diponegoro.
- Thorlund, J., 2010, *Business Analytics for Managers: Taking Intelligence beyond Reporting*. John Willey and Sons, Ltd., United Kingdom.
- Untoro, J., Gusmedi, H., Purwasih, N., 2014. Audit Energi dan Analisis Penghematan Konsumsi Energi pada Sistem. *Jurnal Rekayasa dan Teknologi Elektro*. Hal. 93-104.
- Wang, H. C., dan Guo, J. L., 2013, Constructing a water quality 2.0 OLAP system in Taiwan. *Journal of Cleaner Production*. Volume 40. Hal. 40-45.
- Wasterlund, P., 2008, Business Intelligence : Multidimensional Data Analysis, *Thesis*, in Computing Science 30 ECTS Credits.
- Yi Wu dan Jianjun Zhou., 2015., Design of Electric Energy Acquisition System on Hadoop., *International Journal of Grid Distribution Computing*. Volume 8. No.5. Hal. 47-54.
- Zhou, H., Liu, D., Li, D., Shao, G., Li, Q., 2013, Operating Analysis and Data Mining System for Power Grid Dispatching, *Journal Energy and Power Engineerin*. Volume 5. Hal. 616-620.

Lampiran 2 : *Execute SQL*

Transformation dan Loading

```
-- Step          : Dimension lookup/update
-- Database Connection : start_schema
-- SQL           : ALTER TABLE "public".areadistribusi_dim ADD COLUMN
date_from_KTL TIMESTAMP;
UPDATE "public".areadistribusi_dim SET date_from_KTL=date_from;
ALTER TABLE "public".areadistribusi_dim DROP COLUMN date_from;
ALTER TABLE "public".areadistribusi_dim RENAME date_from_KTL TO date_from;
;
ALTER TABLE "public".areadistribusi_dim ADD COLUMN date_to_KTL
TIMESTAMP;
UPDATE "public".areadistribusi_dim SET date_to_KTL=date_to;
ALTER TABLE "public".areadistribusi_dim DROP COLUMN date_to;
ALTER TABLE "public".areadistribusi_dim RENAME date_to_KTL TO date_to;
;
ALTER TABLE "public".areadistribusi_dim ADD COLUMN TANGGAL_KTL
TIMESTAMP;
UPDATE "public".areadistribusi_dim SET TANGGAL_KTL=TANGGAL;
ALTER TABLE "public".areadistribusi_dim DROP COLUMN TANGGAL;
ALTER TABLE "public".areadistribusi_dim RENAME TANGGAL_KTL TO
TANGGAL;
;
CREATE INDEX idx_areadistribusi_dim_lookup ON
"public".areadistribusi_dim(kode_area)
;
CREATE INDEX idx_areadistribusi_dim_tk ON
"public".areadistribusi_dim(areadistribusi_dim_id)
;

- Step          : Gardu Dimension lookup/update
-- Database Connection : start_schema
-- SQL           : ALTER TABLE "public".wilayahgardu_dim ADD COLUMN
date_from_KTL TIMESTAMP;
UPDATE "public".wilayahgardu_dim SET date_from_KTL=date_from;
ALTER TABLE "public".wilayahgardu_dim DROP COLUMN date_from;
ALTER TABLE "public".wilayahgardu_dim RENAME date_from_KTL TO date_from;
;
ALTER TABLE "public".wilayahgardu_dim ADD COLUMN date_to_KTL
TIMESTAMP;
UPDATE "public".wilayahgardu_dim SET date_to_KTL=date_to;
ALTER TABLE "public".wilayahgardu_dim DROP COLUMN date_to;
ALTER TABLE "public".wilayahgardu_dim RENAME date_to_KTL TO date_to;
```

```

;
ALTER TABLE "public".wilayahgardu_dim ADD COLUMN TANGGAL_KTL
TIMESTAMP;
UPDATE "public".wilayahgardu_dim SET TANGGAL_KTL=TANGGAL;
ALTER TABLE "public".wilayahgardu_dim DROP COLUMN TANGGAL;
ALTER TABLE "public".wilayahgardu_dim RENAME TANGGAL_KTL TO
TANGGAL;

;
ALTER TABLE "public".wilayahgardu_dim ADD COLUMN Kwh_KTL
NUMERIC(17, 2);
UPDATE "public".wilayahgardu_dim SET Kwh_KTL=Kwh;
ALTER TABLE "public".wilayahgardu_dim DROP COLUMN Kwh;
ALTER TABLE "public".wilayahgardu_dim RENAME Kwh_KTL TO Kwh;

;
CREATE INDEX idx_wilayahgardu_dim_lookup ON
"public".wilayahgardu_dim(kode_gi)
;
CREATE INDEX idx_wilayahgardu_dim_tk ON
"public".wilayahgardu_dim(wilayahgardu_dim_id)
;

-- Step          : beban daya Dimension lookup/update
-- Database Connection : start_schema
-- SQL           : ALTER TABLE "public".bebandaya_dim ADD COLUMN
date_from_KTL TIMESTAMP;
UPDATE "public".bebandaya_dim SET date_from_KTL=date_from;
ALTER TABLE "public".bebandaya_dim DROP COLUMN date_from;
ALTER TABLE "public".bebandaya_dim RENAME date_from_KTL TO date_from;

;
ALTER TABLE "public".bebandaya_dim ADD COLUMN date_to_KTL TIMESTAMP;
UPDATE "public".bebandaya_dim SET date_to_KTL=date_to;
ALTER TABLE "public".bebandaya_dim DROP COLUMN date_to;
ALTER TABLE "public".bebandaya_dim RENAME date_to_KTL TO date_to;

;
ALTER TABLE "public".bebandaya_dim ADD COLUMN TANGGAL_KTL
TIMESTAMP;
UPDATE "public".bebandaya_dim SET TANGGAL_KTL=TANGGAL;
ALTER TABLE "public".bebandaya_dim DROP COLUMN TANGGAL;
ALTER TABLE "public".bebandaya_dim RENAME TANGGAL_KTL TO TANGGAL;

;
ALTER TABLE "public".bebandaya_dim ADD COLUMN J20_KTL NUMERIC(16, 1);
UPDATE "public".bebandaya_dim SET J20_KTL=J20;
ALTER TABLE "public".bebandaya_dim DROP COLUMN J20;
ALTER TABLE "public".bebandaya_dim RENAME J20_KTL TO J20;

```

```

;
ALTER TABLE "public".bebandaya_dim ADD COLUMN J21_KTL NUMERIC(16, 1);
UPDATE "public".bebandaya_dim SET J21_KTL=J21;
ALTER TABLE "public".bebandaya_dim DROP COLUMN J21;
ALTER TABLE "public".bebandaya_dim RENAME J21_KTL TO J21;

;
ALTER TABLE "public".bebandaya_dim ADD COLUMN J22_KTL NUMERIC(16, 1);
UPDATE "public".bebandaya_dim SET J22_KTL=J22;
ALTER TABLE "public".bebandaya_dim DROP COLUMN J22;
ALTER TABLE "public".bebandaya_dim RENAME J22_KTL TO J22;

;
ALTER TABLE "public".bebandaya_dim ADD COLUMN J23_KTL NUMERIC(16, 1);
UPDATE "public".bebandaya_dim SET J23_KTL=J23;
ALTER TABLE "public".bebandaya_dim DROP COLUMN J23;
ALTER TABLE "public".bebandaya_dim RENAME J23_KTL TO J23;

;
CREATE INDEX idx_bebandaya_dim_lookup ON
"public".bebandaya_dim(kode_penyulang)
;
CREATE INDEX idx_bebandaya_dim_tk ON
"public".bebandaya_dim(bebandaya_dim_id)
;

-- Step          : Waktu Dimension lookup/update
-- Database Connection : start_schema
-- SQL           : ALTER TABLE "public".waktu_dim ADD COLUMN
date_from_KTL TIMESTAMP;
UPDATE "public".waktu_dim SET date_from_KTL=date_from;
ALTER TABLE "public".waktu_dim DROP COLUMN date_from;
ALTER TABLE "public".waktu_dim RENAME date_from_KTL TO date_from;

;
ALTER TABLE "public".waktu_dim ADD COLUMN date_to_KTL TIMESTAMP;
UPDATE "public".waktu_dim SET date_to_KTL=date_to;
ALTER TABLE "public".waktu_dim DROP COLUMN date_to;
ALTER TABLE "public".waktu_dim RENAME date_to_KTL TO date_to;

;
ALTER TABLE "public".waktu_dim ADD COLUMN tgl_beban_KTL TIMESTAMP;
UPDATE "public".waktu_dim SET tgl_beban_KTL=tgl_beban;
ALTER TABLE "public".waktu_dim DROP COLUMN tgl_beban;
ALTER TABLE "public".waktu_dim RENAME tgl_beban_KTL TO tgl_beban;

;
CREATE INDEX idx_waktu_dim_lookup ON "public".waktu_dim(tgl_beban)
;
CREATE INDEX idx_waktu_dim_tk ON "public".waktu_dim(waktu_dim_id)

```

```

;
-- Step      : fakta jan Table output
-- Database Connection : start_schema
-- SQL       : ALTER TABLE "public".faktajan_dim ADD COLUMN
KODE_TANGGAL BIGINT
;
ALTER TABLE "public".faktajan_dim ADD COLUMN TANGGAL TIMESTAMP
;
ALTER TABLE "public".faktajan_dim ADD COLUMN KODE_GI VARCHAR(15)
;
ALTER TABLE "public".faktajan_dim ADD COLUMN KODE_STATUS
VARCHAR(15)
;
ALTER TABLE "public".faktajan_dim ADD COLUMN KODE_PENYULANG
VARCHAR(15)
;
ALTER TABLE "public".faktajan_dim ADD COLUMN KODE_AREA VARCHAR(15)
;
ALTER TABLE "public".faktajan_dim ADD COLUMN KODE_TRAFO BIGINT
;
ALTER TABLE "public".faktajan_dim ADD COLUMN MAX_ARUS BIGINT
;
ALTER TABLE "public".faktajan_dim ADD COLUMN "MAX" BIGINT
;
ALTER TABLE "public".faktajan_dim ADD COLUMN Total BIGINT
;
ALTER TABLE "public".faktajan_dim ADD COLUMN Kwh NUMERIC(17, 2)
;
ALTER TABLE "public".faktajan_dim ADD COLUMN MAX_SIANG BIGINT
;
ALTER TABLE "public".faktajan_dim ADD COLUMN MAX_MALAM BIGINT
;

```

Cube Dimensi Start Schema

```

<Schema name="cube_distribusi">
  <Cube name="dim_bebanfeeder" visible="true" cache="true" enabled="true">
    <Table name="dim_fact_feeder" schema="public">
    </Table>
    <Dimension type="TimeDimension" visible="true" foreignKey="dim_waktu_id"
highCardinality="false" name="Waktu">
      <Hierarchy name="Waktu" visible="true" hasAll="true"
allMemberName="Semua_Waktu" primaryKey="dim_waktu_id">
        <Table name="dim_waktu" schema="public">
        </Table>
        <Level name="Tahun" visible="true" column="tahun" type="String"
uniqueMembers="false" levelType="TimeYears" hideMemberIf="Never">
        </Level>

```

```

        <Level name="Semester" visible="true" column="semester" type="String"
uniqueMembers="false" levelType="TimeQuarters" hideMemberIf="Never">
        </Level>
        <Level name="Nama Bulan" visible="true" column="nama_bulan"
type="String" uniqueMembers="false" levelType="TimeMonths"
hideMemberIf="Never">
        </Level>
        <Level name="Mingguan" visible="true" column="mingguan" type="String"
uniqueMembers="false" levelType="TimeWeeks" hideMemberIf="Never">
        </Level>
        <Level name="Tanggal" visible="true" column="tgl_beban" type="String"
uniqueMembers="false" levelType="TimeDays" hideMemberIf="Never">
        </Level>
        <Level name="Hari" visible="true" column="nama_hari" type="String"
uniqueMembers="false" levelType="TimeDays" hideMemberIf="Never">
        </Level>
        <Level name="Tanggal Libur" visible="true" column="tanggal_libur"
type="String" uniqueMembers="false" levelType="TimeDays" hideMemberIf="Never">
        </Level>
    </Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true" foreignKey="dim_area_id"
highCardinality="false" name="Area">
    <Hierarchy name="Area" visible="true" hasAll="true" allMemberName="Semu_
Area" primaryKey="dim_area_id">
        <Table name="dim_area" schema="public">
        </Table>
        <Level name="Nama Area" visible="true" column="area" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
    </Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true"
foreignKey="dim_garduinduk_id" highCardinality="false" name="Gardu_Induk">
    <Hierarchy name="Gardu_Induk" visible="true" hasAll="true"
allMemberName="Semua_Gardu_Induk" primaryKey="dim_garduinduk_id">
        <Table name="dm_garduinduk" schema="public">
        </Table>
        <Level name="Nama Gardu Induk" visible="true"
column="nama_gardu_induk" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="Kode Trafo" visible="true" column="kode_trafo" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="No Trafo" visible="true" column="no_trafo" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="Max Arus" visible="true" column="max_arus" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">

```



```

        </Level>
    </Hierarchy>
</Dimension>
<Dimension type="StandardDimension" visible="true"
foreignKey="dim_freeder_id" highCardinality="false" name="Feeder">
    <Hierarchy name="Feeder" visible="true" hasAll="true"
allMemberName="Semua_Freeder" primaryKey="dim_freeder_id">
        <Table name="dim_freeder" schema="public">
            </Table>
            <Level name="Nama Feeder" visible="false" column="nama_freeder"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                </Level>
            <Level name="Status Feeder" visible="true" column="status_penyulang"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                </Level>
        </Hierarchy>
        <Hierarchy name="Beban Jam" visible="true" hasAll="true"
allMemberName="Semua_beban" primaryKey="dim_freeder_id">
            <Table name="dim_freeder" schema="public">
                </Table>
                <Level name="J00" visible="true" column="j00" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
                <Level name="J01" visible="true" column="j01" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
                <Level name="J02" visible="true" column="j02" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
                <Level name="J03" visible="true" column="j03" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
                <Level name="J04" visible="true" column="j04" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
                <Level name="J05" visible="true" column="j05" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
                <Level name="J06" visible="true" column="j06" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
                <Level name="J07" visible="true" column="j07" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
                <Level name="J08" visible="true" column="j08" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
                <Level name="J09" visible="true" column="j09" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
                    </Level>
            </Hierarchy>
        </Dimension>
    </Table>

```

```

        <Level name="J10" visible="true" column="j10" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J11" visible="true" column="j11" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J12" visible="true" column="j12" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J13" visible="true" column="j13" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J14" visible="true" column="j14" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J15" visible="true" column="j15" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J16" visible="true" column="j16" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J17" visible="true" column="j17" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J18" visible="true" column="j18" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J19" visible="true" column="j19" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J20" visible="true" column="j20" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J21" visible="true" column="j21" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J22" visible="true" column="j22" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="J23" visible="true" column="j23" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
    </Hierarchy>
</Dimension>
<Measure name="Beban Maksimum" column="MAX" aggregator="max"
visible="true">
</Measure>
<Measure name="Beban Minimum" column="MAX" aggregator="min"
visible="true">
</Measure>

```

```
<Measure name="Max Siang" column="max_siang" aggregator="max"
visible="true">
  </Measure>
  <Measure name="Max Malam" column="max_malam" aggregator="max"
visible="true">
  </Measure>
  <Measure name="Rata-rata" column="rata_rata" aggregator="avg" visible="true">
  </Measure>
  <Measure name="Total Beban (Ampere)" column="total_ampere"
aggregator="sum" visible="true">
  </Measure>
  <Measure name="Total Beban (kwh)" column="kwh" aggregator="sum"
visible="true">
  </Measure>
  <Measure name="Jumlah" column="total_ampere" aggregator="count"
visible="true">
  </Measure>
  <Measure name="distinct count" column="total_ampere" aggregator="distinct-
count" visible="true">
  </Measure>
</Cube>
</Schema>
```