PERANCANGAN DATAWAREHOUSE DENGAN MENGGUNAKAN TOOLS PENTAHO DAN TABLEAU PADA DATA LAYANAN ANTAR JEMPUT IZIN BERMOTOR (AJIB) DI DINAS PM DAN PTSP PROVINSI DKI JAKARTA

Darmawan Subuh[1], Furkon[2]

Jurusan Sistem Informasi, STMIK Indonesia
Jl. Siantar No. 6, Cideng – Gambir, Jakarta 10150
Email : darmawan@stmik-indonesia.ac.id, elfurkon123@gmail.com.

Abstrak

Sistem informasi layanan AJIB (Antar Jemput Izin Bermotor / Layanan Pengiriman dan Pengantaran Lisensi Kendaraan Bermotor) dari Departemen Investasi dan Layanan Terpadu Satu Pintu Provinsi DKI Jakarta adalah satu inovasi yang dilakukan oleh Departemen Investasi dan IOSS Provinsi DKI Jakarta dalam memberikan layanan lisensi dan non-lisensi untuk masyarakat khususnya warga DKI Jakarta. Layanan AJIB ini didukung oleh perangkat lunak aplikasi berbasis android yang dapat menghubungkan petugas AJIB dan pemohon yang membutuhkan layanan AJIB. Selama implementasi sistem informasi layanan AJIB, ditemukan beberapa masalah seperti keterlambatan rekapitulasi jumlah layanan AJIB, kesulitan dalam menentukan jumlah layanan berdasarkan unit otoritas dan jenis lisensi serta kesulitan pimpinan dalam membuat kebijakan karena kurangnya data hasil analisis yang akurat. Sistem informasi layanan AJIB menyimpan data pemohon lisensi dan pesanan non-lisensi. Data-data ini perlu dikumpulkan di *Data warehouse*. Data ini meningkat secara berkala dan oleh karena itu data tersebut dapat dijalankan dari *Data warehouse* tanpa mempengaruhi kinerja operasional. Data tersebut digunakan secara luas dengan menggunakan teknik *business intelligence* untuk mengkategorikan data layanan AJIB dan karena itu mereka dapat membantu manajemen untuk membuat kebijakan yang benar berdasarkan data historis. Agar penambahan data tidak mempengaruhi bisnis, penambangan data itu sendiri harus relevan dan menjadi bagian dari bisnis proses.

Kata kunci: *Data Warehouse, business intelligence*

1 Pendahuluan

Untuk menciptakan Ibukota yang ramah investasi, Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Provinsi DKI Jakarta ("Dinas PM dan PTSP Provinsi DKI Jakarta") telah melakukan riset mengenai kendala dalam menyelenggarakan pelayanan perizinan dan non perizinan terhadap masyarakat khususnya warga DKI Jakarta. Beberapa kendala tersebut diantaranya prosedur yang berbelit, antrian yang panjang, dan maraknya praktik percaloan dan pungutan liar. Apabila diamati, ketiga permasalahan utama tersebut saling terkait satu sama lain. Disebabkan oleh prosedur pengurusan izin yang berbelit, maka berefek pada tahap verifikasi yang membutuhkan waktu yang cukup lama, sehingga antrian pengajuan izin pun memanjang. Lebih lanjut, hal tersebut pun menyebabkan terbukanya peluang praktik percaloan, juga pungutan liar yang dilakukan oleh oknum tidak bertanggung jawab [1].

Maka menjadi penting bagi Dinas PM dan PTSP Provinsi DKI Jakarta untuk memutuskan rantai permasalahan tersebut dari pangkalnya. Untuk itu Dinas PM dan PTSP Provinsi DKI Jakarta pun berupaya mengeluarkan beragam inovasi dan salah satunya inovasi andalannya yaitu layanan AJIB (Antar Jemput Izin Bermotor). Layanan AJIB ini didukung dengan *software* aplikasi berbasis android, yang dapat menghubungkan petugas AJIB dengan pemohon yang membutuhkan jasa layanan AJIB. Aplikasi ini memberi kemudahan bagi pengguna jasa layanan AJIB untuk mendapatkan layanan perizinan dan non perizinan yang cepat dan efisien.

Dengan menggunakan aplikasi ini, warga DKI Jakarta tidak perlu mengurus perizinan dengan datang langsung ke unit-unit PTSP di kantor Kelurahan, Kecamatan, Walikota atau Dinas Provinsi, cukup dengan order petugas AJIB melalui *smartphone*. Kemudian petugas AJIB yang menerima order akan menjemput berkas perizinan ke lokasi yang telah ditentukan pada saat memesan layanan, kemudian petugas AJIB yang akan mengurus berkas pemohonan izin ke unit PTSP sesuai kewenangan izin berdasarkan Peraturan Gubernur Provinsi DKI Jakarta No. 47 Tahun 2017 Tentang Petunjuk Pelaksanaan Pelayanan Terpadu Satu Pintu.

Kemudahan proses pemesanan layanan AJIB berdampak langsung pada lonjakan jumlah pelayanan hingga mencapai puncaknya pada bulan November 2016 mencapai 12.422 pelayanan. Berarti rasio jumlah pelayanan terhadap jumlah petugas meningkat menjadi 1:6, satu petugas melayani 6 perizinan per hari dengan total 120 petugas AJIB menangani lebih dari 620 layanan setiap hari [2]. Rasio tersebut dapat terus bertambah bila permintaan masyarakat semakin banyak. Permasalahannya peningkatan rasio antara permintaan dengan kemampuan pelayanan AJIB belum tentu berjalan secara linier ini karena keterbatasan SDM dan keuangan. Selain itu pada penggunaan aplikasi AJIB masih terdapat proses-proses yang dilakukan secara manual sehingga dalam perekapan data order sering terjadi kesalahan. Hal ini menyebabkan keterlambatan waktu pelaporan terhadap pimpinan. Dalam menangani keterbatasan-keterbatasan yang ada Dinas PM dan PTSP Provinsi DKI Jakarta khususnya pada layanan AJIB memerlukan suatu informasi yang akurat untuk membuat keputusan-keputusan yang strategis untuk pengembangan sistem informasi selanjutnya.

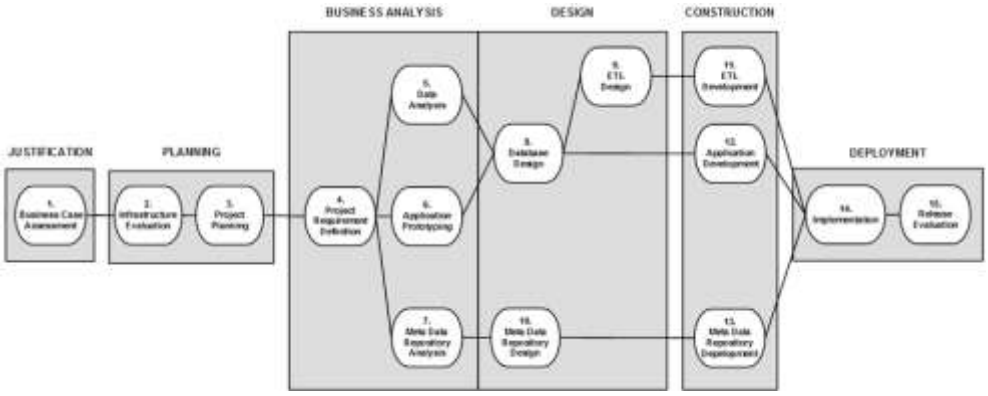
Berdasarkan kendala-kendala dan fakta yang terjadi, maka diperlukan suatu perancangan sistem yang mampu melakukan analisis data dan memberikan laporan (*reporting*) secara otomatis dan dapat memberikan informasi berupa pengetahuan (*knowledge*) yang berfungsi untuk menanggulangi keterlambatan laporan, monitoring petugas AJIB, penempatan posisi petugas AJIB dan penilaian kinerja petugas AJIB. Untuk itu diperlukan suatu aplikasi *business intelligence* untuk melengkapi sistem informasi tersebut. Fungsi dari aplikasi ini adalah menampilkan *report* dalam format yang mudah dipahami, dengan penggunaan visualisasi berbagai *chart* atau diagram dalam satu halaman presentasi. Dengan menambahkan proses analisis data multi dimensi user pengguna dapat dengan mudah dan selektif memilih dan melihat data dari berbagai sudut pandang yang berbeda-beda.

2 Metodelogi Penelitian

Metode Penelitian adalah cara ilmiah untuk mendapatkan data yang valid dengan tujuan dapat ditemukan, dibuktikan, dan dikembangkan suatu pengetahuan, sehingga pada gilirannya dapat digunakan untuk memahami, memecahkan dan mengantisipasi masalah dalam bidang bisnis. Metodologi penelitian ini terdiri dari metodologi pengembangan sistem dan metode pengumpulan data

2.1 Metodologi Pengembangan business intelligence

Dalam merancang dan mengimplementasikan *business intelligence*, dapat digunakan beberapa metode yang ada. Dalam penulisan ini, metode pengembangan sistem yang dibahas adalah menggunakan pendekatan *business intelligence roadmap* (Moss dan Atre, 2003) seperti yang tergambar pada Gambar 1.1. di bawah ini.



Gambar 2. 1. *Business Intelligence Project Life Cycle* (Moss dan Atre, 2003) [3].

3 Landasan Teori

3.1 Data Warehouse

Menurut Willian Harvey Inmon (W.H. Inmon, 1970) atau lebih dikenal dengan sebutan Bill Inmon, merupakan salah satu tokoh penting di dalam perkembangan dunia teknologi informasi, khususnya dibidang *data warehouse. Data warehouse didefinisikan* sebagai sekumpulan data yang dimiliki enam buah sifat atau karakteristik berupa berorientasi subjek (*subject oriented*), terintegrasi (*integrated*), berorientasi pada proses (*process oriented*), time variant, dapat diakses dengan mudah (*accesible*), dan bersifat *non volatile* [4].

3.2 Karakteristik *Data Warehouse*

Menurut Bill Inmon (1970) terdapat enam buah karekteristik pada *data warehouse*, yang meliputi *Subject Oriented*, *Integrated*, *Process Oriented*, *Time Variant*, *Accesible* dan *Non Volatil*e [4].

1) *Subject Oriented*

Data yang ditampilkan dan data yang disusun hanyalah data menurut subjek yang diperlukan untuk proses pengambilan keputusan. Data-data ini dirangkum ke dalam bentuk dimensi, yang meliputi periode waktu (*time*), riwayat (*history*), wilayah (*region*), dan lain-lain.

2) Integrated

Data warehouse dibangun dari proses integrasi berbagai sumber data (terutamanya sumber data beruba *database*) yang berasal dari berbagai aplikasi (*software*), menjadi satu kesatuan yang utuh. Karakteristik *integrated* muncul sebagai bentuk lain dari karakteriktik *subject oriented.*

3) Time Variant

Data warehouse (yang dikumpulkan dari berbagai sumber data dari berbagai aplikasi) diidentifikasikan dari periode waktu penyimpanannya. Atau dengan kata lain, proses penyimpanan data-data yang dikumpulkan tersebut, berdasarkan kepada waktu. Adapun informasi yang disajikan oleh data-data pada *data warehouse* tersebut, dilihat sudut pandang riwayat penyimpanan (*historical point of view*).

4) Non Volatile

Data-data dari berbagai sumber data yang dikumpulkan ke dalam *data warehouse*, tidak boleh mengalami manipulasi data dalam bentuk *edit*, *update* dan *delete.* Dengan kata lain, *data*

warehouse mementingkan adanya *historical data* (data asli), guna menunjang proses analisis data yang dilakukan kelak.

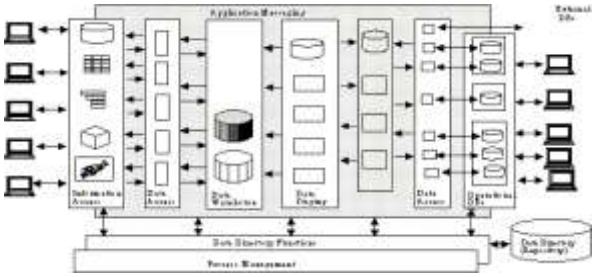
5) Process Oriented

Data warehouse dipandang sebagai sebuah proses berkesinambungan di dalam pengolahan data menjadi informasi serta pengiriman informasi tersebut. Proses menjadi orientasi dari *data warehouse* di dalam operasionalnya.

6) Accessible

Data warehouse beserta dengan data-data di dalamnya, harus dapat diakses dengan mudah oleh pengguna. Pengguna dapat memperoleh data yang mana saja yang mereka butuhkan, baik keseluruhan maupun sebagian (parsial), sesuai dengan hak akses (*privillege*) yang diartika oleh sistem atau pemilik sistem.

Menurut Ken Orr (1999), "*Data Warehouse Architecture* (DWA) atau arsitektur *data warehouse* merupakan sebuah cara yang dilakukan oleh arsitek *data warehouse* atau SDM yang berkaitan dengan perancangan dan desain sistem *data warehouse* di dalam sebuah organisasi, untuk mempresentasikannya ke dalam bentuk bagan/gambar/desain, yang memuat segala hal mengenai struktur data, komunikasi, pemrosesan, dan presentasi dari komputasi *end to end* antar komputer di dalam sistem *data warehouse*" [4].



Gambar 3.1. *Data Warehouse Architecture* versi Ken Orr

3.3 OLAP (*On Line Analytical Processing*)

OLAP merupakan proses komputer yang memungkinkan pengguna dapat dengan mudah dan selektif memilih dan melihat data dari sudut pandang yang berbeda-beda. Data pada OLAP disimpan dalam basis data multidimensi. Jika pada basis data relasional terdiri dari dua dimensi, maka pada basis data multidimensi terdiri dari banyak dimensi yang dapat dipisahkan oleh OLAP menjadi beberapa *sub atribut* [4].

3.4 ETL (*Extract, Transform, and Loading*)

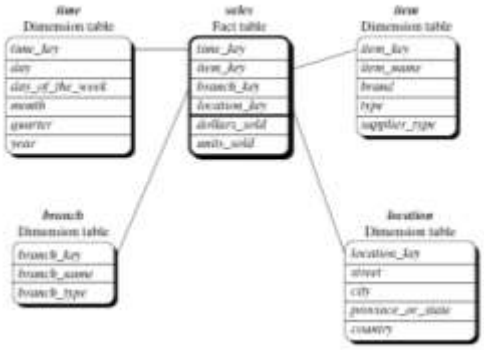
Menurut Ralph Kimball dan Joe Caserta, ETL (*Extract, Transform, and Loading)* merupakan urutan langkah di dalam pemrosesan data pada *database* (khususnya *data warehouse*), yang melibatkan proses pengekstrasian (*extraction*) data-data dari sumber-sumber datanya, mempertahankan kualitas data, menerapkan standarisasi untuk data, menyajikannya ke dalam berbagai bentuk (*transformation*), untuk kemudian di alirkan atau diteruskan (*loading*) ke *data warehouse* untuk digudangkan, dalam rangka kebutuan untuk analisis data maupun informasi [4]

3.5 *Dimensional Modeling*

Model data yang populer untuk *data warehouse* adalah model multidimensi. Beberapa konsep pemodelan *data warehouse* pada model multidimensi yang dikenal pada umumnya adalah *star schema, snowflake* dan *fact constellation schema* [4].

1) *Star Schema* (Skema Bintang)

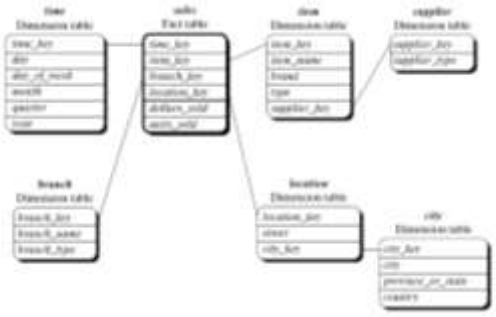
Paradigma pemodelan yang paling umum adalah skema bintang, di mana gudang data berisi : tabel pusat (tabel fakta) yang berisi sebagian besar data tanpa redundansi, dan dimensi tabel, satu untuk setiap dimensi. Grafik skema menyerupai s*tarburst*, dengan tabel dimensi ditampilkan dalam pola radial di sekitar tabel fakta pusat.



Gambar 3.2. *Star Schema*

2) Snowflake Schema

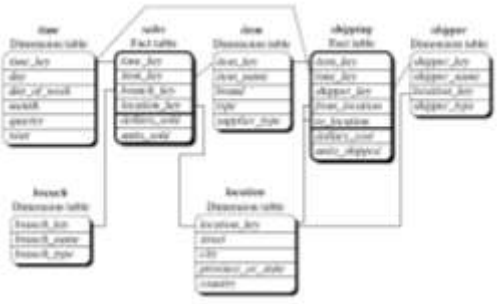
Skema snowflake adalah varian dari model skema bintang, di mana beberapa tabel dimensi dinormalkan, sehingga lebih memecah data menjadi tabel tambahan. Grafik skema yang dihasilkan membentuk bentuk yang mirip dengan kepingan salju.



Gambar 3.3. *Snowflake Schema*

3) Fact Constellation Schema

Aplikasi yang canggih mungkin memerlukan beberapa tabel fakta untuk dibagikan tabel dimensi. Skema semacam ini dapat dilihat sebagai kumpulan bintang, dan karenanya disebut skema galaksi atau konstelasi fakta.



Gambar 3.4. *Fact Constellation Schema*

3.6 Layanan Antar Jemput Izin Bermotor (AJIB)

AJIB adalah salah satu inovasi andalan Dinas PM dan PTSP Provinsi DKI Jakarta. Dimana semua layanan AJIB diberikan secara gratis untuk melayani warga DKI Jakarta yang akan mengurus perizinan dan non perizinan. Layanan ini menjadi solusi yang mampu mengatasi berbagai permasalahan yang kerap kali ditemui pada praktik pelayanan publik. Tugas utama AJIB secara garis besar adalah:

1. Menjemput berkas pengajuan izin;
2. Melakukan verifikasi berkas; dan
3. Mengantarkan berkas izin yang sudah dirilis.

4 Hasil dan Pembahasan

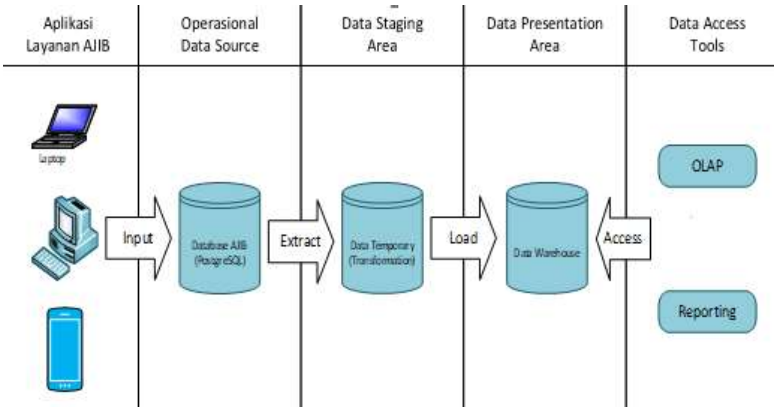
4.1 Perancangan *Data Warehouse*

4.1.1 Perancangan Arsitektur Layanan AJIB

Perancangan arsitektur layanan AJIB meliputi perancangan arsitektur *logical* dan arsitektur fisik. Arsitektur *logical* berupa rancangan tahapan alur data dari sumber data sampai menjadi data pada *datawarehouse*, sedangkan arsitektur *fisik* berupa gambaran konfigurasi teknis dari data *warehouse* tersebut. Perancangan arsitektur tersebut tentunya sedapat mungkin di desain sesuai dengan kondisi yang ada pada sistem layanan AJIB.

1) Arsitektur *Logical*

Arsitektur *logical* pada perancangan *data warehouse* layanan AJIB dapat dilihat pada gambar di bawah ini.



Gambar 4.1. Arsitektur *Logical*

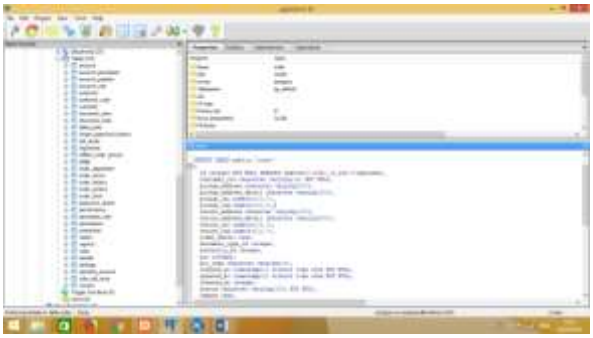
2) Arsitektur Fisik

Arsitektur fisik pada perancangan *data warehouse* layanan AJIB dapat dilihat pada gambar di bawah ini.

Gambar 4.2. Arsitektur Fisik

3) Sumber Data

Berikut sumber data penelitian pada basis data operasional layanan AJIB. Jumlah tabel operasional adalah 33 tabel dengan total kolom 258 dan ukuran data saat dilakukan penelitian adalah 143 megabytes. Dari 33 tabel tersebut tidak semua tabel akan disimpan dalam *data warehouse* dan dari masing-masing tabel yang digunakan, tidak semua kolom akan disimpan. Tabel-tabel yang akan diproses dan disimpan dalam *data warehouse* adalah tabel account, authority, authority_type, customer, document_type, order, order_driver, dan reports.



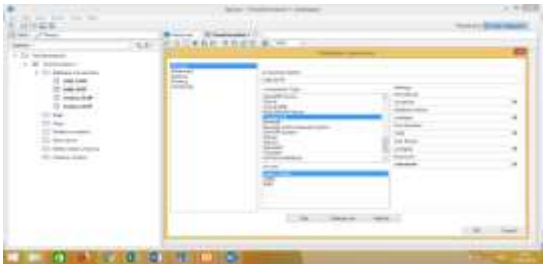
Gambar 4.3. Database Sistem Layanan AJIB

4.1.2 Proses ETL (*Extract*, *Transformation*, dan *Loading*)

Proses ETL (*Extract*, *Transformation*, dan *Loading*) merupakan urutan langkah di dalam pemrosesan data pada *data warehouse* yang melibatkan proses pengekstrasian (*extraction*) data-data dari sumber-sumber datanya, mempertahankan kualitas data, menerapkan standarisasi untuk data, menyajikannya ke dalam berbagai bentuk (*transformation*), untuk kemudian di alirkan atau diteruskan (*loading*) ke *data warehouse* untuk digudangkan.

1) Proses *Extract*

Pada proses *extract,* dilakukan ekstraksi data yang berasal dari berbagai sumber data. Sumber data tersebut dapat beragam database yang menggunakan sistem *On Line Transaction Processing* (OLTP). Sumber data yang digunakan pada penelitian ini adalah database AJIB dengan platform PostgreSQL.Data yang diambil kemudian dipilih dan disimpan ke temporary database. Pelaksanaan ETL akan diproses pada aplikasi Pentaho, maka perlu dilakukan koneksi dari Pentaho ke database yang telah disimpan.



Gambar 4.4. Koneksi Pentaho ke Database

2) Proses *Transformation*

Setelah melewati proses extract dan cleansing, proses selanjutnya adalah melakukan transformasi dari sumber database ke dalam bentuk tabel dimensi dan tabel fakta yang akan membentuk star schema.

a. Transformasi Dimensi *Customer*
Table input *customer* di drag untuk memasukkan data yang berasal dari *data source*, kemudian dihubungkan ke *add sequence* untuk membuat *surogate key*. Selanjutnya melakukan *mapping* pada *select value* dari sumber database *customer* ke dimensi *customer* untuk membentuk *star schema*

Gambar 4.5. Transformasi Dimensi *Customer*

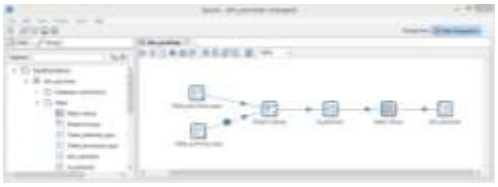
b. Transformasi Dimensi *Officer*
Table input *account* dan *order_driver* di drag untuk memasukkan data yang berasal dari *data source*, kedua tabel input tersebut di gabungkan dengan sebuah *stream lookup* untuk menghasilkan output yang kemudian dihubungkan ke *add sequence* untuk membuat *surogate key*. Selanjutnya melakukan *mapping* pada *select value* dari sumber database *account dan order_driver* ke dimensi *customer* untuk membentuk *star schema.*

Gambar 4.6. Transformasi Dimensi *Officer*

c. Transformasi Dimensi Kewenangan
Table input *authority*, *authority_type* dan *account*di drag untuk memasukkan data yang berasal dari *data source*, ketiga tabel input tersebut di gabungkan dengan dua buah *stream lookup* untuk menghasilkan *output*yang kemudian dihubungkan ke *add sequence* untuk membuat *surogate key*. Selanjutnya melakukan *mapping* pada *select value* dari sumber database *account dan order_driver* ke dimensi *customer* untuk membentuk *star schema.*

Gambar 4.7. Transformasi Dimensi Kewenangan

d. Transformasi Dimensi Perizinan
Table input *document_type* dan *dim_kewenangan* di drag untuk memasukkan data yang berasal dari *data source*, kedua tabel input tersebut di gabungkan dengan sebuah *stream lookup* untuk menghasilkan output yang kemudian dihubungkan ke *add sequence* untuk membuat *surogate key*. Selanjutnya melakukan *mapping* pada *select value* dari sumber database *account dan order_driver* ke dimensi *customer* untuk membentuk *star schema.*

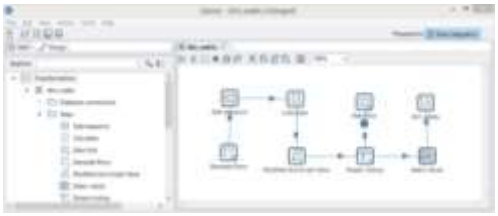


Gambar 4.8. Transformasi Dimensi Perizinan

e. Transformasi Dimensi Waktu

Dimensi waktu merupakan dimensi yang datanya digenerate sendiri. Setelah proses *add sequence*, *calculator* berfungsi untuk memasukkan kalkulasi dari field pada dimensi waktu. *Java Script* digunakan untuk memasukkan fungsi nilai pada waktu. *Data grid* berisi metadata yang memuat informasi nama dan urutan bulan, kemudian digabungkan dengan *stream lookup* dan menghasilkan *output* yang akan membentuk *star schema*

.



Gambar 4.9. Transformasi Dimensi Waktu

f. Transformasi Dimensi *Order Online*

Table input *order_online* di drag untuk memasukkan data yang berasal dari *data source*, kemudian dihubungkan ke *add sequence* untuk membuat *surogate key*. Selanjutnya melakukan *mapping* pada *select value* dari sumber database *order* ke dimensi *order_online* untuk membentuk *star schema*.

Gambar 4.10. Transformasi Dimensi *Order Online*

g. Transformasi Dimensi *Order Offline*

Table input *order_offline* di drag untuk memasukkan data yang berasal dari *data source*, kemudian dihubungkan ke *add sequence* untuk membuat *surogate key*. Selanjutnya melakukan *mapping* pada *select value* dari sumber database *report* ke dimensi *order_offline* untuk membentuk *star schema*.

Gambar 4.11. Transformasi Dimensi *Order Offline*

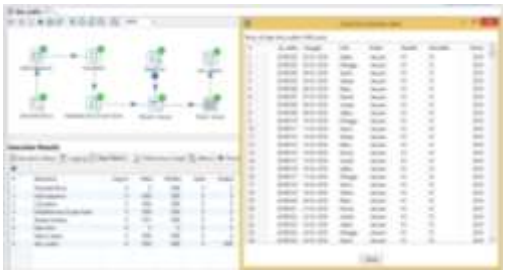
h. Transformasi Fakta Order

Dimensi-dimensi yang telah dibuat kemudian digabungkan dengan menggunakan *stream lookup*. Setelah digabungkan selanjutnya melakukan *mapping* pada *select value* dan menghasilkan *output fakta order* yang merupakan pusat star schema.

Gambar 4.12. Transformasi Dimensi Fakta Order

3) Proses *Loading*

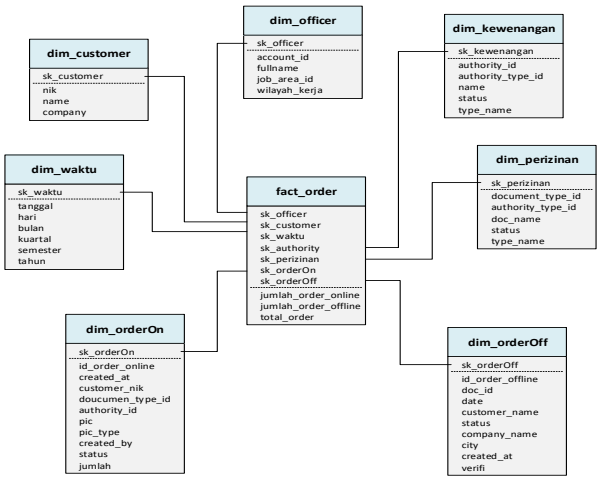
Proses yang dilakukan pada tahap terakhir ini adalah proses pemuatan data (*loading*). Pada tahap ini secara garis besar data telah diseragamkan ke dalam format data yang didapatkan dari hasil transformasi ke dalam data warehouse untuk dapat diteruskan ke tatap muka aplikasi dan layanan, dimana pengguna dapat mengakses keluaran dari tahapan *loading* ini dalam bentuk data maupun informasi. Penyajian dapat dilakukan melalui laporan maupun sekumpulan data untuk kebutuhan analisis dan pengambilan keputusan. Proses *loading* dapat di lihat pada gambar. di bawah ini.



Gambar 4.13. Proses *Loading preview data*

4.1.3 Skema Bintang (*Star Schema*)

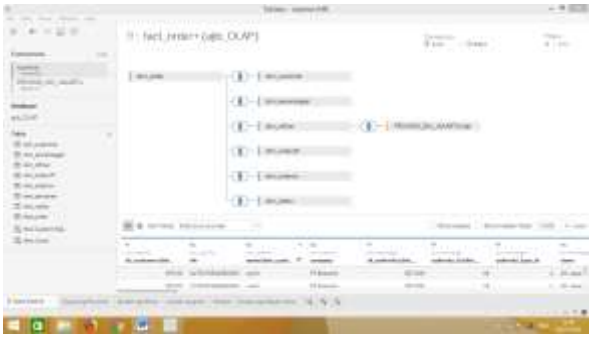
Berdasarkan hasil transformasi tabel-tabel dimensi yang dibuat pada perancangan data warehouse, maka dibentuk skema bintang sebagaimana terlihat pada gambar berikut:



Gambar 4.14. *Star Schema* Sistem Layanan AJIB

4.1.4 Informasi Layanan

Setelah proses data warehouse dilakukan maka proses terakhir adalah menampilkan semua Informasi yang diperlukan dalam bentuk visualisasi *dashboard* menggunakan aplikasi Tableau. Untuk menghasilkan informasi dalam bentuk visualisasi *dashboard* perlu dibentuk Cube OLAP yang terdiri dari fact order, dimensi waktu, dimensi order online, dimensi order offline, dimensi perizinan, dimensi customer, dimensi officer dan dimensi kewenangan.



Gambar 4.15. *Cube OLAP* Layanan AJIB

4.1.5 Visualisasi Dashborad

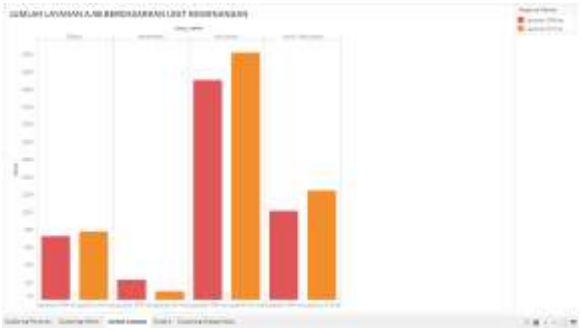
Cube OLAP dari layanan AJIB dapat menghasilkan informasi dalam bentuk charts dan grafik sehingga akan mudah di fahami oleh orang yang melihat informasi tersebut.

Gambar 4.16. Tampilan *Dashboard* Layanan AJIB

Berikut adalah informasi clustering wilayah kerja layanan AJIB berdasarkan jumlah layanan yang ditunjukan dengan visualisasi data dalam bentuk filled map.

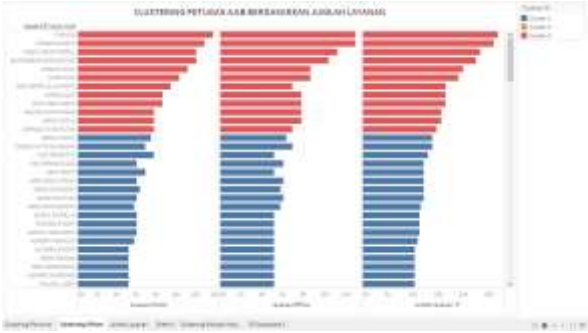
Gambar 4.17. Informasi Clustering Wilayah Kerja Berdasarkan Jumlah Layanan

Berikut adalah informasi jumlah layanan AJIB berdasarkan unit kewenangan yang ditunjukan dengan visualisasi data dalam bentuk *side-by-side bars*.



Gambar 4.158. Informasi Jumlah Layanan AJIB Berdasarkan Unit Kewenangan

Berikut adalah informasi clustering petugas AJIB berdasarkan jumlah layanan yang ditunjukan dengan visualisasi data dalam bentuk *horizontal bars*.



Gambar 4.19. Informasi Clustering Petugas AJIB Berdasarkan Jumlah Layanan

Berikut adalah informasi clustering jenis perizinan berdasarkan jumlah layanan yang ditunjukan dengan visualisasi data dalam bentuk *horizontal bars*.

Gambar 4.20. Informasi Clustering Jenis Perizinan Berdasarkan Jumlah Layanan

5. Kesimpulan dan Saran

5.1 Kesimpulan

Pada Layanan AJIB Dinas Penanaman Modal Dan Pelayanan Terpadu Satu Pintu Provinsi DKI Jakarta di mulai dari tahap analisis, perancangan, serta desain sistem maka penulis pada kesempatan ini mengambil beberapa kesimpulan bahwa:

1. Perancangan data warehouse pada sistem pelaporan jumlah layanan *online* dan *offline* layanan AJIB Dinas PM dan PTSP Provinsi DKI Jakarta sangatlah diperlukan, dengan harapan dapat mencapai sasaran ketepatan, kecepatan, serta kemudahan monitoring dalam pelayanan perizinan dan non perizinan, dimana data warehouse yang dibangun menjadi sumber data yang dapat diambil dan digunakan untuk kepentingan-kepentingan *stakehoulder* dalam membuat kebijakan-kebijakan dan pengembangan sistem selanjutnya.
2. Pembuatan tampilan informasi dalam bentuk grafik atau *dashborad* penting dilakukan agar informasi mudah dibaca dan difahami oleh berbagai pihak. Karena informasi merupakan hasil akhir dari suatu sistem, ideal atau tidaknya suatu sistem di lihat dari informasi yang dihasilkan oleh sistem tersebut. Pada sistem yang diusulkan diharapkan mampu menghasilkan informasi yang penting yang diperlukan oleh pimpinan untuk membuat kebijakan dalam hal pengembangan sistem layanan AJIB selanjutnya.

5.2 Saran

Setelah berusaha semaksimal mungkin untuk menyelesaikan tugas akhir ini, maka perkenankanlah penulis untuk memberikan saran yang pernah diketahui selama dalam melakukan penelitian, diantaranya:

1. Penggunaan aplikasi layanan AJIB lebih dimaksimalkan lagi. Karena sumber data yang digunakan untuk perancangan data warehouse adalah data pada sistem layanan AJIB, sehingga hasil dari penelitian ini akan baik jika sumber datanya valid.
2. Perlu adanya pengembangan aplikasi layanan AJIB kedepan terutama terkait wilayah kerja petugas AJIB supaya dimasukan perkecamatan sesuai dengan penempatan masing-masing petugas.
3. Diharapkan perancangan *data warehouse* yang sudah dilakukan dikembangkan kembali karena penulis menyadari masih banyak kekurangan.
4. Perancangan data warehouse diterapkan pada bagian lain pada Dinas PM Dan PTSP Provinsi DKI Jakarta

DAFTAR PUSTAKA

[1] KPPOD, “Badan Pelayanan Terpadu Satu Pintu (BPTSP) di Provinsi DKI Jakarta: Perspektif Kewenangan Dan Kelembagaan,” Jakarta, 2013.

[2] M. Hasstriansyah, “*Revolusi Pelayanan Publik di BPTSP DKI Jakarta*”. Jakarta: DPMPTSP Provinsi DKI Jakarta, 2017.

[3] Atre, Moss. [2003]. Proses ETL dan Konsep Businness Intelligence Roadmap.

[4] I. Putu Agus Eka Pratama, *Handbook Data Warehouse, Teori Dan Praktik Berbasiskan Open Source*”. Bandung: Informatika Bandung, 201