PEMBANGUNAN DATA WAREHOUSE DAN REPORTING TOOLS PADA DINAS BKKBN KABUPATEN CIANJUR

Hamdan Muhammad[1]

[1] Program Studi Teknik Informatika
Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia
Jl. Dipatiukur No 112-116 Bandung - Indonesia
E-mail: hamdanmuhammad@icloud.com[1]

ABSTRAK

Data yang terdapat pada setiap divisi yang ada di BKKBN Kabupaten Cianjur masih perlu di olah dan dianalisis untuk mendapatkan informasi sesuai dengan kebutuhan setiap divisinya. Selain itu data yang ada di setiap divisi tidak tersusun sesuai kurun waktu perbulan atau pertahunnya, hal ini berdampak kepada kegiatan petugas di setiap divisisnya dalam pencarian data yang dibutuhkan untuk proses analisis dan pembuatan laporan hasil analisis menjadi sangat lambat dan dapat menghambat proses pengambilan keputusan.

Dari permasalahan yang ada dibutuhkan suatu aplikasi *data warehouse* di BKKBN Kabupaten Cianjur yang membantu setiap divisinya dalam menyajikan informasi yang multidimensi dan ringkas. Sumber daya pengetahuan dapat diakses dengan mudah dan cepat dapat membantu meningkatkan performa dan kinerja dalam melakukan analisa data di setiap divisinya dan mendukung dalam pembuatan laporan yang memiliki periode waktu berupa tabel dan grafik. Pembangunan *data warehouse* menggunakan skema *constellation*.

Dengan dibangunnya aplikasi *data warehouse* ini dapat diambil kesimpulan yaitu, proses penganalisaan terhadap data mengenai Pil KB dan Keluarga sejahtera lebih mudah, dengan ditampilkannya tampilan berupa tabel dan grafik yang mudah untuk dipahami yang berguna untuk memaksimalkan pengambilan keputusan, serta aplikasi *data warehouse* ini dapat menyajikan data dalam bentuk laporan yang memiliki interval waktu tertentu sesuai yang dibutuhkan.

Kata kunci : *Data Warehouse*, Skema *Constellation*, Multidimensi.

1. PENDAHULUAN

Badan Kependudukan dan Keluarga Berencana Nasional (BKKBN) Kabupaten Cianjur adalah badan yang bertugas untuk melaksanakan tugas pemerintah dibidang keluarga berencana dan keluarga sejahtera sesuai dengan peraturan perundang-undangan yang berlaku.

Dari hasil wawancara dengan Kepala BKKBN dan para staf dari ke 3 divisi yang ada di BKKBN Kabupaten Cianjur, di dapat bahwa data yang ada di setiap divisi yang dikumpulkan dari setiap kecamatan masih perlu di olah dan dianalisis untuk mendapatkan informasi sesuai dengan kebutuhan setiap divisinya. Selain itu data yang ada di setiap divisi tidak tersusun sesuai kurun waktu perbulan atau pertahunnya, hal ini berdampak kepada kegiatan petugas di setiap divisisnya dalam pencarian data yang dibutuhkan untuk proses analisis dan pembuatan laporan hasil analisis menjadi sangat lambat dan dapat menghambat proses pengambilan keputusan. Selain itu dalam pembuatan laporan hasil analisis di setiap divisinya yang memiliki periode waktu menjadi kurang efektif dan efisien.

Dari permasalahan yang telah dijelaskan, maka dibutuhkan suatu aplikasi *data warehouse* di BKKBN Kabupaten Cianjur yang membantu setiap divisinya dalam menyajikan informasi yang multidimensi dan ringkas sehingga dapat memaksimalkan suatu keputusan yang dibuat, sehingga program-program yang akan di canangkan selanjutnya oleh pihak BKKBN Kabupaten Cianjur dapat berjalan secara efektif dan tepat sasaran, serta mendukung dalam pembuatan laporan yang multidimensi atau memiliki periode waktu berupa tabel dan grafik.

Tujuan dari pembangunan *data warehouse* pada BKKBN Kabupaten Cianjur ini adalah sebagai berikut:

1. Menyajikan informasi yang multidimensi dan ringkas sehingga dapat mempercepat proses penganalisaan data dan memaksimalkan suatu keputusan yang dibuat oleh *top management* untuk menentukan program-program seperti program KB, penyedian stok alat kontrasepsi, pembinaan keluarga, penyuluhan kesehatan reproduksi remaja, dan kesejahteraan keluarga yang akan berjalan selanjutnya di setiap kecamatan yang ada di Kabupaten Cianjur agar lebih efektif dan tepat sasaran.

2. Menyajikan laporan hasil analisis yang memiliki interval waktu seperti perbulan dan pertahun untuk pihak BKKBN Kabupaten Cianjur di setiap divisinya.

Batasan masalah dalam pembangunan *data warehouse* pada BKKBN Kabupaten Cianjur ini adalah sebagai berikut:

1. Data yang diambil adalah data yang dimiliki oleh BKKBN Kabupaten Cianjur yaitu data pembinaan keluarga, pik-krr, peserta kb, stok alat kontrasepsi, uppks, sasaran peserta kb baru (unmetneed) dan tempat pelayanan.
2. Data yang diolah dari tahun 2011-2015.
3. Sistem di bangun berbasis *Client Server*.
4. Pengujian *data warehouse* menggunakan OLAP (*On-Line Analytical Processing*).
5. DBMS menggunakan SQL Server 2012.
6. Penyajian laporan hasil analisis disajikan dalam bentuk tabel dan grafik.
7. Analisis dan perancangan perangkat lunak menggunakan analisis berorientasi objek.

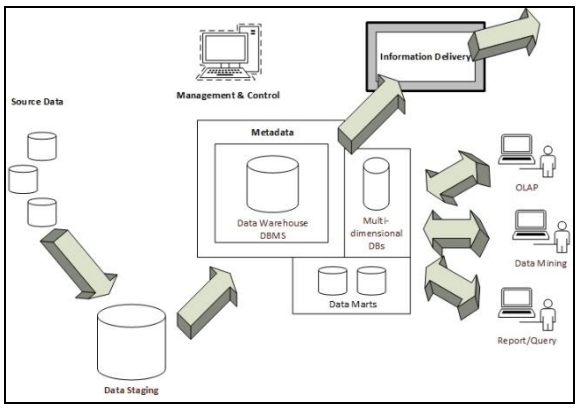
1.1 Data Warehouse

Data warehouse adalah sebuah tempat penyimpanan data yang lengkap dan konsisten dimana data yang disimpan itu saling berelasi dan dirancang berdasarkan query dan analisis daripada proses transaksinya [1]. Data tersebut ditransformasikan menjadi sebuah informasi yang dapat diakses kapan saja dan selalu *up to date*. Informasi ini kemudian akan digunakan untuk dianalisis untuk menghasilkan informasi baru [8].

Data warehouse juga bisa dikatakan sebagai kumpulan data yang berorientasi subjek, terintegrasi, tidak dapat di *update*, memiliki dimensi waktu yang digunakan untuk mendukung proses manajemen pengambilan keputusan dan kecerdasan bisnis. Tujuan dari *data warehouse* adalah sebagai berikut [3] :

1. Memberikan kemudahan untuk mengakses informasi yang ada.
2. Kemudahan disini berbicara tentang efisiensi. *Data warehouse* harus efisien sehingga dengan mudah dipahami oleh *user* bukan hanya oleh *devoleper* saja. Selain itu, pengguna juga dapat mengkombinasikan data dalam *data warehouse* dengan berbagai cara (*slicing and dicing*). Untuk mengakses *data warehouse* disarankan sebaiknya dapat dilakukan dengan sederhana dan mudah dioperasikan.
3. Menyediakan informasi yang konsisten.
4. *Data warehouse* hanya berisi informasi-informasi yang relevan bagi kebutuhan *user* untuk pengambil keputusan. Oleh karena itu, kredibilitas data yang terdapat dalam *data warehouse* harus dapat dipertanggungjawabkan.
5. Mampu beradaptasi dan tahan terhadap perubahan.
6. Perubahan-perubahan yang terjadi harus dapat diatasi oleh *data warehouse*. Dengan kata lain, *data warehouse* harus dirancang agar mampu menghadapi setiap perubahan dengan terencana. Hal ini berarti perubahan yang terjadi tidak boleh merusak atau mengganggu data dan aplikasi yang telah ada sebelumnya.
7. Mampu mengamankan informasi.
8. Informasi yang tersimpan dalam *data warehouse* harus tersimpan dengan aman. Dengan kata lain, informasi tersebut tidak boleh sampai jatuh ke tangan yang salah. Oleh karena itu, *data warehouse* harus mampu mengendalikan setiap akses dari informasi yang ada.
9. Mampu memberikan dukungan dalam pengambilan keputusan.
10. Ini merupakan tujuan yang paling penting dan harus ada dalam setiap pembuatan *data warehouse*. Data warehouse bisa digambarkan sebagai kumpulan teknologi pendukung keputusan, dimaksudkan agar setiap pekerjaan yang berhubungan dengan informasi, dapat membuat keputusan dengan cepat dan tepat.
11. *User friendly*.
12. Seperti pada tujuan *data warehouse* pertama, *data warehouse* harus dirancang agar dapat dioperasikan dengan mudah oleh *user*. Tidak seperti sistem operasional dimana seringkali *user* tidak memiliki pilihan yang lain kecuali menggunakan sistem baru, akan tetapi *user data warehouse* biasanya merupakan pilihan. Oleh karena itu, proses penentuan *user data warehouse* merupakan faktor yang sangat penting.

Arsitektur pada data *warehouse* dikelompokkan menjadi 5 bagian seperti Gambar 1 [4] :



Gambar 1 Arsitektur Data Warehouse

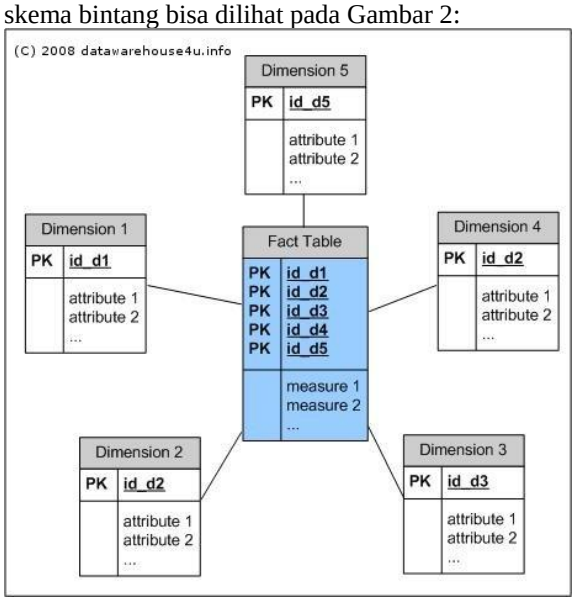
1.2 Model Dimensional Data Warehouse

Model dimensional pada *data warehouse* terdiri dari tabel fakta dan tabel dimensi. Tabel fakta merupakan tabel yang berisi kumpulan *foreign key* dari *primary* key yang terdapat pada masing-masing tabel dimensi, sedangkan tabel dimensi merupakan

tabel yang berisi data detail yang menjelaskan *foreign key* yang terdapat pada tabel fakta. Ada beberapa model skema yang terdapat pada pemodelan *data warehouse*, yaitu skema *star*, skema *snowflake*, dan skema *constellation*. Penjelasan dari masing-masing model skema adalah sebagai berikut :

1. Skema Bintang (*Star Schema*)

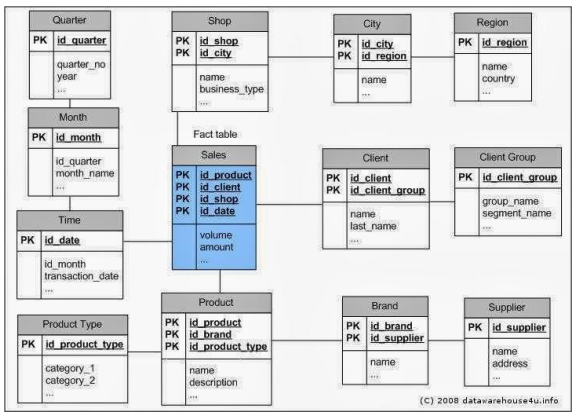
Skema ini mengikuti bentuk bintang, dimana terdapat satu tabel fakta di pusat bintang dengan beberapa tabel dimensi yang mengelilinginya. Semua tabel dimensi berhubungan dengan tabel fakta. Tabel fakta memiliki beberapa *primary key* dalam tabel dimensi. Berikut ini adalah contoh dari skema bintang bisa dilihat pada Gambar 2:



Gambar 2 Skema Bintang

2. Skema Bola Salju (*Snowflake Schema*)

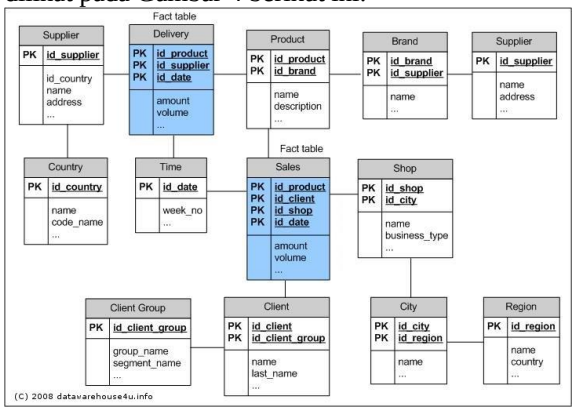
Skema bola salju merupakan perluasan dari skema bintang dengan tambahan beberapa tabel dimensi yang tidak berhubungan secara langsung dengan tabel fakta. Tabel dimensi tersebut berhubungan dengan tabel dimensi yang lain. Contoh skema bola salju dapat dilihat pada Gambar 3 berikut ini:



Gambar 3 Skema Bola Salju

3. Skema *Constellation*

Suatu skema dikatakan sebagai skema *constellation* jika ada satu atau beberapa tabel dimensi yang dipakai bersamaan oleh satu atau lebih tabel fakta [5]. Pada skema ini terdapat beberapa tabel fakta yang menggunakan satu atau beberapa tabel dimensi. Contoh skema c*onstellation* dapat dilihat pada Gambar 4 berikut ini:



Gambar 4 Skema Constellation

1.3 Proses ETL Dalam Data Warehouse

Proses ETL atau biasa disebut *Extract*, *Transform*, dan *Load* merupakan proses pengubahan data dari OLTP *database* menjadi *data warehouse*. Jika dilihat dari arstitektur *data warehouse*, proses ETL ini merupakan proses yang berada di *data staging*.

Proses ETL merupakan proses untuk mengubah, memformat ulang serta mengintegrasikan data yang berasal dari satu atau beberapa OLTP *systems* [6].

1. Extraction

   Extraction merupakan sebuah proses dimana proses tersebut mencari sumber data kemudian dengan menggunakan beberapa kriteria yang sudah diberikan untuk memilah data dan juga untuk mencari data yang berkualitas, kemudian data tersebut diangkut ke file lain atau database [6].

2. Transformation

   Data transformation merupakan suatu fase yang terjadi ketika data sudah menjadi raw data (hasil extraction) diubah menjadi bentuk yang sudah ditetapkan dimana bentuk tersebut harus bisa digunakan dalam data warehouse [4]. Berikut ini adalah beberapa proses dasar yang harus ada dalam data *transformation* :

   a) Selection
      Memilih atau memilah data hasil dari *extraction*.
   b) Splitting/Joining
      Splitting/joining meliputi tipe-tipe manipulasi data yang perlu dilakukan pada proses *selection*.
   c) Conversion
      Proses ini merupakan tahapan paling penting. Pada tahap *conversion*, data hasil

selection kemudian akan diubah menjadi data yang layak digunakan pada *data warehouse*.

d) Summarization
Tahap ini merupakan tahap pembentukan model yang akan ditampilkan kepada user.

e) Enrichment
Tahap ini merupakan tahap pembentukan kembali serta penyederhanaan *field* yang ada untuk membuat *field* tersebut lebih berguna pada *data warehouse*.

3. Loading
Loading adalah suatu proses pemindahan data secara fisik dari OLTP *systems* ke dalam data *destination* atau *data warehouse*. Operasi *loading* terdiri dari memasukkan *record* ke dalam bermacam-macam dimensi dan *fact* tabel yang ada pada data destination atau *data warehouse* [3].

1.4 OLAP (*On-Line Analytical Processing*)

OLAP (*On-Line Analytical Processing*) adalah teknologi yang memproses data ke dalam struktur multidimensi, menyediakan jawaban yang cepat untuk *query* analisis yang kompleks dengan tujuan untuk mengorganisir sejumlah data yang besar, agar bisa dianalisa dan dievaluasi dengan cepat serta menyediakan kecepatan dan fleksibelitas untuk mendukung analisa dalam waktu nyata [2].

Ada beberapa karakteristik OLAP, yaitu :

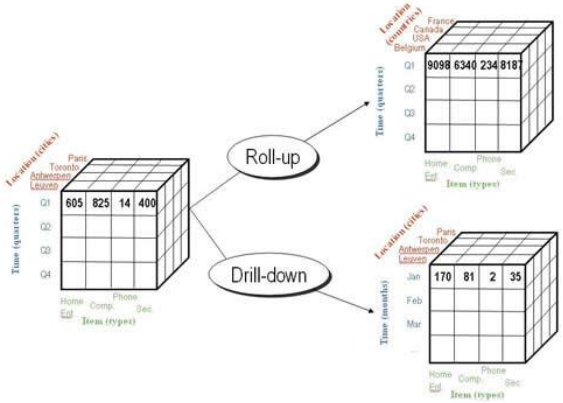
1. Mengijinkan pelaku bisnis melihat data dari sudut pandang *logical* dan multidimensional pada *data warehouse*.
2. Memfasilitasi *query* yang kompleks dan analisa bagi pengguna.
3. Mengijinkan pengguna melakukan *drill-down* untuk menampilkan data yang lebih detail atau *roll-up* untuk agregasi dari suatu dimensi atau beberapa dimensi.
4. Menyediakan proses kalkulasi dan perbandingan data.
5. Menampilkan hasil dalam bentuk tabel atau grafik.

Keuntungan dari OLAP, yaitu :

1. Mengingkatkan produktivitas pemakai akhir bisnis, pengembang IT, dan keseluruhan organisasi.
2. Pengawasan yang lebih dan akses tepat waktu terhadap informasi strategis dapat membuat pengambilan keputusan lebih cepat.
3. Mengurangi pengembangan aplikasi bagi staf IT dengan membuat pemakaian akhir dapat mengubah skema dan membuat model sendiri.
4. Penyimpanan pengawasan organisasi melalui integritas data koorporasi sebagai aplikasi OLAP tergantung pada *data warehouse* dan sistem OLTP untuk memperbaharui sumber tingkatan data.

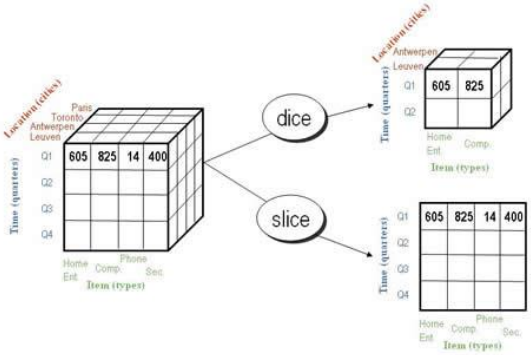
OLAP dapat digunakan untuk melakukan seperti [2]:

1. Konsolidasi (*roll-up*)
Konsolidasi melibatkan pengelompokan data.
2. *Drill-down*
Suatu bentuk yang merupakan kebalikan dari konsolidasi untuk menjabarkan data yang ringkas menjadi data yang lebih detail. Gambaran untuk *roll-up* dan *drill-down* dapat dilihat pada Gambar 6Gambar 5 berikut ini:



Gambar 5 Roll-up dan Drill-down

3. *Slicing* dan *dicing*
Menjabarkan pada kemampuan untuk melihat data dari sudut pandang. Gambaran untuk *slicing* dan *dicing* dapat dilihat pada Gambar 6 berikut ini:



Gambar 6 *Slicing* dan *Dicing*

1.5 SSIS (*SQL Server Integration Service*)

SSIS (*SQL Server Integration Services*) adalah suatu platform untuk membangun sistem yang handal untuk integrasi data, *extraction, transformation*, dan *loading* yang digunakan pada *data warehousing* [7]. SSIS menawarkan solusi dalam menghadapi permasalahan data integrasi. Selain itu, *tools* ini membatu untuk menigkatkan efisiensi waktu pembuatan.

Arsitektur *SQL Server Integration Services* secara umum berisi berbagai macam komponen, diantaranya yaitu:

1. *SSIS Deginer*. Merupakan *tools* yang digunakan untuk membuat dan mengatur paket *integration service*. Pada *SQL Server 2012*, *tools* ini sudah terintegrasi dengan *Visual Studio 2010* yang merupakan bagian *project Bussiness Intelegence*.

2. *Runtime Engine*. Komponen ini berguna untuk menjalankan semua paket SSIS yang sudah dibuat.
3. *Task* dan *executable binary*.
4. *Data Flow Engine* dan *Data Flow*. Komponen *data flow* merupakan enkapsulasi *data flow engine* yang menyediakan *buffer* didalam memori dan bertugas memindahkan data dari sumber data ke tujuan data. Sedangkan *data flow* merupakan sumber data, tujuan data, dan transformasi.
5. *Integration Services service*. Memungkinkan *SQL Server Management Studio* dapat digunakan untuk melakukan monitoring paket SSIS dan mengatur SSIS *storage* yang digunakan.
6. *SQL Server Import* dan *Export Wizard*. *Tools* ini digunakan untuk melakukan *copy* data dari sumber ke tujuan data.

2. ISI PENELITIAN

Terdapat beberapa tahapan analisis dalam pembangunan perangkat lunak *data warehouse* ini, yaitu:

2.1 Analisis Kebutuhan Informasi

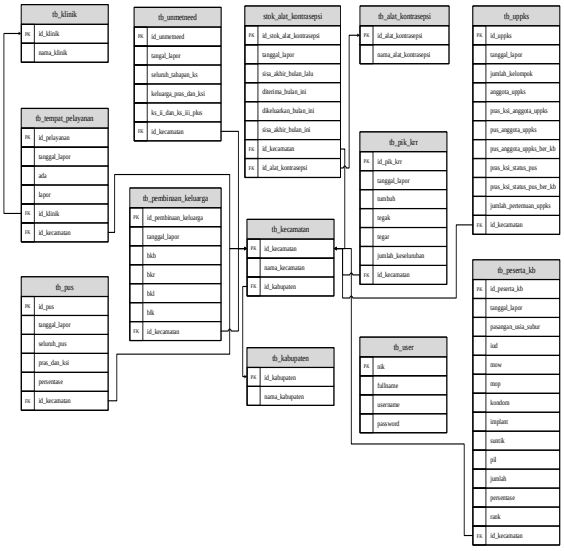
Analisis kebutuhan informasi merupakan tahap untuk menganalisis apa saja yang dibutuhkan oleh BKKBN Kabupaten Cianjur untuk *data warehouse* yang akan dibangun.

Berdasarkan hasil wawancara langsung dengan Kepala BKKBN dan para staf dari ke 3 divisi yang ada di BKKBN Kabupaten Cianjur didapat kebutuhan informasi strategis sebagai berikut:

1. Informasi jumlah ada dan lapor dari klinik kb pemerintah dan swasta di setiap kecamatan setiap bulan setiap tahunnya.
2. Informasi jumlah ada dan lapor dari dokter praktik swasta di setiap kecamatan setiap bulan setiap tahunnya.
3. Informasi jumlah ada dan lapor dari bidan praktik swasta di setiap kecamatan setiap bulan setiap tahunnya.
4. Informasi jumlah PUS pertahapan KS kategori PRA S dan KSI di setiap kecamatan setiap bulan setiap tahunnya.
5. Informasi jumlah peserta KB per mix kotrasepsi di setiap kecamatan setiap bulan setiap tahunnya.
6. Informasi jumlah Un Met Need pertahapan KS di setiap kecamatan setiap bulan setiap tahunnya.
7. Informasi jumlah persediaan alat kontrasepsi per alat kontrasepsi di setiap kecamatan setiap bulan setiap tahunnya.
8. Informasi jumlah bina keluarga balita di setiap kecamatan setiap bulan setiap tahunnya.
9. Informasi jumlah bina keluarga remaja di setiap kecamatan setiap bulan setiap tahunnya.
10. Informasi jumlah bina keluarga lansia di setiap kecamatan setiap bulan setiap tahunnya.
11. Informasi jumlah bina latihan ketenagakerjaan setiap bulan setiap tahunnya.
12. Informasi jumlah kelompok, anggota, pertemua UPPKS pertahapan KS kategori PRA S dan KSI di setiap kecamatan setiap bulan setiap tahun nya.
13. Informasi jumlah PIK-KRR kategori tegak, tumbuh, tegar di setiap kecamatan setiap bulan setiap tahun nya.

2.2 Sumber Data

Database OLTP yang sekarang berada di BKKBN Kabupaten Cianjur menjadi *data source* untuk membangun *data warehouse*. Skema relasi pada *database* yang ada dapat di lihat pada Gambar 7:



Gambar 7 Skema Relasi Tabel OLTP BKKBN Kabupaten Cianjur

2.3 Analisis Arsitektur Data Warehouse

Arsitektur yang digunakan untuk pembangunan *data warehouse* adalah menggunakan model *two-layer architecture*. Analisis arsitektur ini terbagi dalam empat *layer*, yaitu analisis *source layer*, analisis *data stagging*, analisis *data warehouse layer*, dan analisis menggunakan OLAP.

1. Analisis Source Layer
   Pada lapisan ini, data masih berupa data operasional. Sumber data yang digunakan dalam pembangunan *data warehouse* ini sudah berupa data *logic* yang ada di *database*. Sumber data yang digunakan yaitu berasal dari *database* OLTP BKKBN Kabupaten Cianjur.
2. Analisis Data Stagging
   Pada lapisan ini, data operasional akan diekstrak (melalui proses ETL) ke dalam *data warehouse*.

a) Extract
   Proses *extract* dilakukan untuk mengekstraksi data atau menjabarkan data apa saja yang ada pada suber data yang akan dijadikan *data*

warehouse. Proses ini merupakan pemilihan data dari sumber data untuk pembuatan *data warehouse*, yaitu tabel kabupaten, tabel kecamatan, tabel alat kontrasepsi, tabel stok alat kontrasepsi, tabel klinik, tabel klinik tempat pelayanan, tabel pembinaan keluarga, tabel pus, tabel peserta kb, tabel pik-krr, tabel unmetneed dan tabel uppks. Tabel User tidak di ekstrak karena tidak di butuhkan. Atribut-atribut yang ada pada tabel yang akan diekstrak tidak ada perubahan menambah atau mengurangi atribut-atributnya, masih tetap sama dengan sumber data. Proses ekstraksi data dari sumber data ke dalam *data warehouse* adalah sebagai berikut:

Tabel 1 Tabel Extract

No	Nama Tabel	Field
1	Tabel Kabupaten	id_kabupaten
		nama_kabupaten
2	Tabel Kecamatan	id_kecamatan
		nama_kecamatan
		id_kabupaten
3	Tabel Alat Kontrasepsi	id_alat_kontasepsi
		nama_alat_kontrasepsi
4	Tabel Stok Alat Kontrasepsi	id_stok_alat_kontasepsi
		tanggal_lapor
		sisa_akhir_bulan_lalu
		diterima_bulan_ini
		dikeluarkan_bulan_ini
		sisa_akhir_bulan_ini
		id_alat_kontasepsi
		id_kecamatan
5	Tabel Klinik	id_klinik
		nama_klinik
6	Tabel Tempat Pelayanan	id_tempat_pelayanan
		tanggal_lapor
		id_klinik
		ada
		lapor
		id_kecamatan
7	Tabel Pembinaan Keluarga	id_pembinaan_keluarga
		tanggal_lapor
		bkb
		bkr
		bkl
		blk
		id_kecamatan
8	Tabel PUS	id_pus
		tanggal_lapor
		seluruh_pus
		pras_dan_ksi
		persentase
		id_kecamatan
9	Tabel Peserta KB	id_peserta_kb
		tanggal_lapor
		pasangan_usia_subur
		iud
		mow
		mop
		kondom
		implant
		suntik
		pil
		jumlah
		persentase
		rank
		id_kecamatan
10	Tabel PIK-KRR	id_pik_krr
		tanggal_lapor
		tumbuh
		tegak
		tegar
		jumlah_keseluruhan
		id_kecamatan
11	Tabel Unmetneed	id_unmetneed
		tanggal_lapor
		bulan
		seluruh_tahapan_ks
		keluarga_pras_dan_ksi
		ks_ii_dan_ks_iii_plus
		id_kecamatan
12	Tabel Uppks	id_uppks
		tanggal_lapor
		jumlah_kelompok
		anggota_uppks
		pras_ksi_anggota_uppks
		pus_anggota_uppks
		pus_anggota_uppks_ber_kb
		pras_ksi_status_pus
		pras_ksi_status_pus_ber_kb
		jumlah_pertemuan_uppks
		id_kecamatan

b) Transform

Proses *transform* yang dilakukan terdiri dari 2 proses, yaitu:

1. Cleaning

   Proses *cleaning* dilakukan untuk membersihkan data yang tidak digunakan dari tabel yang sudah diekstrak, yaitu menghilangkan *field* yang tidak terpakai dalam *data warehouse*. Berikut merupakan nama *field* yang dihilangkan dalam proses *cleaning*:

   a. Pada tabel stok alat kontrasepsi tidak memerlukan *field* id_stok_alat_kontrasepsi, sisa_akhir_bulan_lalu, diterima_bulan_ini, dikeluarkan bulan ini, id_kecamatan dan tanggal_lapor.
   b. Pada tabel tempat pelayanan tidak memerlukan *field* id_tempat_pelayanan, id_kecamatan dan tanggal_lapor.
   c. Pada tabel pembinaan keluarga tidak memerlukan *field* id_pembinaan_keluarga, id_kecamatan dan tanggal_lapor.
   d. Pada tabel pus tidak memerlukan *field* id_pus, persentase, id_kecamatan dan tanggal_lapor.
   e. Pada tabel peserta kb tidak memerlukan *field* id_peserta_kb, persentase, rank, id_kecamatan dan tanggal_lapor.
   f. Pada tabel pik-krr tidak memerlukan *field* id_pik_krr, id_kecamatan dan tanggal_lapor.
   g. Pada tabel unmetneed tidak memerlukan *field* id_unmetneed, id_kecamatan dan tanggal_lapor.
   h. Pada tabel uppks tidak memerlukan *field* id_uppks, id_kecamatan dan tanggal_lapor.

2. Conditioning

Proses *conditioning* dilakukan untuk mengubah format dari data oprasional menjadi format *data warehouse*. Tabel-tabel yang di *conditioning* yaitu tabel stok alat kontrasepsi, tempat pelayanan, pembinaan keluarga, pus, pik-krr, peserta kb, unmetneed dan uppks. Proses *conditioning* dilakukan untuk memastikan tidak adanya redudansi data, sehingga pada suatu tabel fakta bisa memiliki lebih dari satu tabel dimensi. Pada **Error! Reference source not found.** akan dijelaskan contoh dari tahapan *conditioning* pada tabel stok alat kontrasepsi.

Tabel Error! No text of specified style in document.**.2 Conditioning Tabel Stok Alat Kontrasepsi**

Tabel Stok Alat Kontrasepsi	
No	**Field**
1	tanggal_lapor
2	sisa_akhir_bulan_ini
3	id_alat_kontasepsi
4	id_kecamatan

fact_stok_alat_kontrasepsi	
No	**Field**
1	id_alat_kontasepsi
2	id_wilayah
3	id_waktu
4	sisa_akhir_bulan_ini

Tabel Stok Alat Kontrasepsi	
tanggal_lapor	date

dim_waktu	
id_waktu	int
tanggal	int
bulan	varchar
tahun	Int
fulldate	date

Tabel Stok Alat Kontrasepsi	
id_kecamatan	int

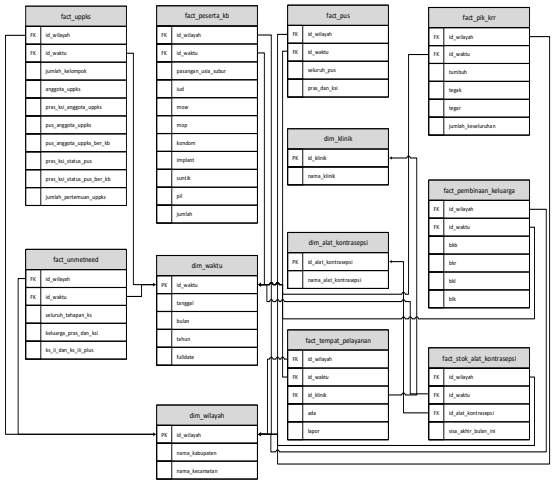
dim_wilayah	
id_wilayah	int
nama_kabupaten	varchar
nama_kecamatan	varchar

c. Loading

Setelah data di *extract* dan di *transform*, selanjutnya data tersebut dimasukkan ke dalam *data warehouse*. Proses *loading* pada aplikasi *data warehouse* akan dilakukan secara otomatis setelah proses *transform* selesai. Teknik yang digunakan adalah *update*, dimana proses ini akan langsung meng-*update data warehouse* tanpa merubah data yang sudah ada.

3. Analisis Data *Warehouse Layer*

Pada lapisan ini, data yang sudah melalui proses ETL akan disimpan pada sebuah penyimpanan logic yang tersentralisasi, yaitu data *warehouse*. Tabel-tabel yang akan dibutuhkan dalam pembuatan perancangan data *warehouse*, yaitu tabel fakta stok alat kontrasepsi, tabel fakta tempat pelayanan, tabel fakta pembinaan keluarga, table fakta pus, table fakta peserta kb, table fakta pik_kkr, table fakta unmetneed dan table fakta uppks. Selain itu terdapat tabel dimensi yang akan digunakan bersama dalam beberapa tabel fakta, yaitu tabel dimensi wilayah, tabel dimensi waktu, table dimensi alat kontrasepsi, dan tabel dimensi klinik. Skema yang digunakan dalam pembangunan data *warehouse* adalah skema constellation. Berikut skema constellation dalam pembangunan data *warehouse* dapat dilihat pada Gambar 8:



Gambar 8 Skema constellation

Tabel 3 menunjukkan jenis dari masing-masing tabel yang ada pada Gambar 8

Tabel 3 Penjelasan Skema *Data Warehouse*

No	Nama Tabel	Jenis Tabel
1	fact_stok_alat_kontrasepsi	Fakta
2	fact_pelayanan	Fakta
3	fact_pembinaan_keluarga	Fakta
4	fact_pus	Fakta
5	fact_peserta_kb	Fakta
6	fact_pik_krr	Fakta
7	fact_unmetneed	Fakta
8	fact_uppks	Fakta
9	dim_wilayah	Dimensi
10	dim_waktu	Dimensi
12	dim_alat_kontrasepsi	Dimensi
13	dim_klinik	Dimensi

2.4 Spesifikasi Kebutuhan Perangkat Lunak

Analisis spesifikasi kebutuhan perangkat lunak berisi deskripsi dari kebutuhan perangkat lunak yang akan dibangun baik kebutuhan fungsional maupun kebutuhan non fungsional. Tabel spesifikasi kebutuhan perangkat lunak dapat dilihat pada Tabel 4 dan Tabel 5.

Tabel 4 Spesifikasi Kebutuhan Perangkat Lunak Fungsional

Kode Kebutuhan	Kebutuhan
SKPL-F-1	User dapat melakukan login kedalam sistem
SKPL-F-2	User dapat melakukan proses etl dalam sistem
SKPL-F-3	User dapat melakukan proses analisis data
SKPL-F-4	User dapat menyimpan data kedalam format excel dan pdf
SKPL-F-5	User dapat menampilkan data dimensi dan fakta
SKPL-F-6	User dapat melihat informasi data di dalam chart

Tabel 5 Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional

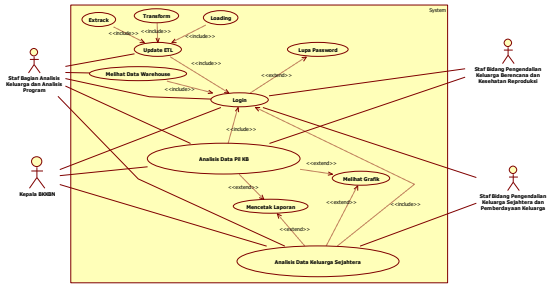
Kode Kebutuhan	Kebutuhan
SKPL-NF-1	Sistem dapat berjalan pada *platfrom windows*.
SKPL-NF-2	Sistem dapat berjalan pada *hardware* berspesifikasi menengah.
SKPL-NF-3	Bahasa pemrograman yang digunakan adalah C#.

2.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk menguraikan proses kegiatan dan kebutuhan yang akan diterapkan pada sistem yang akan dibangun.

2.5.1 Use Case Diagram

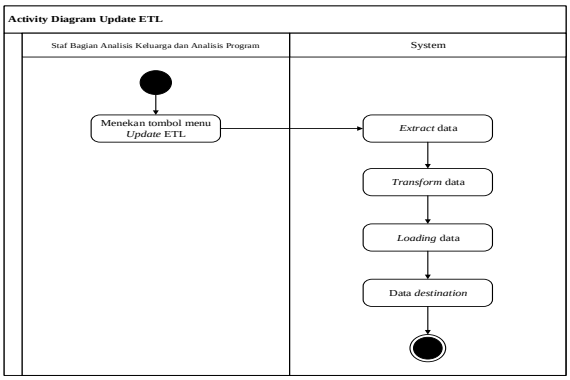
Use case diagram merupakan gambaran fungsionalitas atau perilaku dari sistem. Berikut *use case diagram* pada aplikasi dapat dilihat pada Gambar 9.



Gambar 9 Use Case Diagram

2.5.2 Activity Diagram

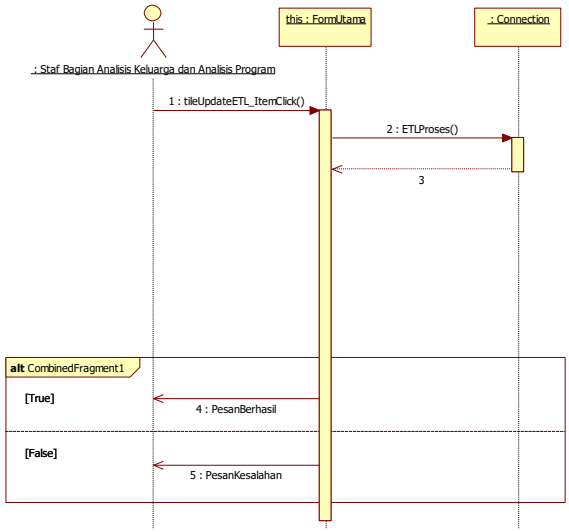
Activity diagram memodelkan aliran kerja dari urutan aktivitas dalam suatu proses yang mengacu pada *use case diagram* yang ada. Berikut ini adalah contoh *activity diagram* dari *Update* ETL dapat dilihat pada Gambar 10.



Gambar 10 *Activity Diagram Update* ETL

2.5.3 Sequence Diagram

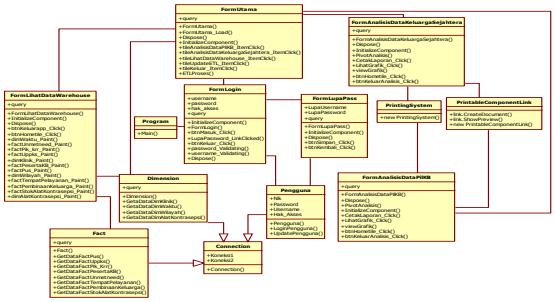
Sequence diagram menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek yang saling berinteraksi. Berikut ini adalah contoh *sequence diagram* dari *update* ETL dapat dilihat pada Gambar 11.



Gambar 11 *Sequence Diagram Update* ETL

2.5.4 Class Diagram

Class diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur ini meliputi atribut-atribut dan metode-metode yang ada pada masing-masing kelas. *Class diagram* dari perangkat lunak *data warehouse*yang akan dibangun dapat dilihat pada Gambar 12.



Gambar 12 *Class Diagram*

2.6 Implementasi Basis Data

Implementasi basis data menjelaskan seluruh *query* yang digunakan pada saat pembuatan data *warehouse*. Berikut ini contoh implementasi tabel dim_wilayah dapat dilihat pada Tabel 6.

Tabel 6 Implementasi tabel dim_wilayah

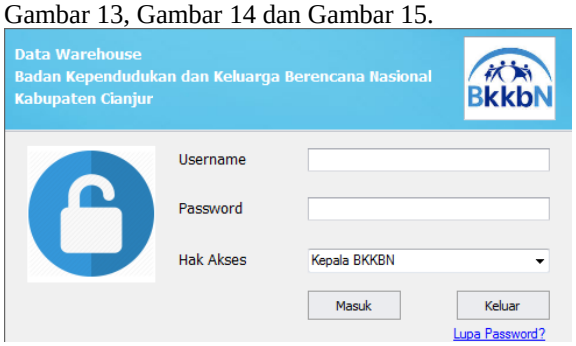
Create dim_wilayah

```
CREATE TABLE [dbo].[dim_wilayah](
	[id_wilayah] [int] NOT NULL,
	[nama_kabupaten] [varchar](50) NOT
NULL,[nama_kecamatan] [varchar](50) NOT NULL,


CONSTRAINT [PK_dim_wilayah] PRIMARY KEY
CLUSTERED
(
	[id_wilayah] ASC
)WITH (PAD_INDEX = OFF,
STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
```

2.7 Implementasi Antar Muka

Implementasi antar muka dilakukan dengan menampilkan setiap tampilan sistem yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah contoh implementasi antar muka dari aplikasi data warehouse dapata dilihat pada Gambar 13, Gambar 14 dan Gambar 15.



Gambar 13 Halaman *Login*



Gambar 14 Halaman Lupa *Password*



Gambar 15 Halaman *Update* ETL

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil analisis serta pengujian yang dilakukan maka dapat disimpulkan bahwa:

1. Aplikasi *data warehouse* yang dibangun dapat mempercepat proses penganalisaan terhadap data mengenai Pil KB dan Keluarga sejahtera lebih mudah, dengan ditampilkannya tampilan berupa tabel dan grafik yang mudah untuk dipahami yang berguna untuk memaksimalkan pengambilan keputusan untuk menentukan program apa saja yang berjalan selanjutnya.
2. Aplikasi *data warehouse* dapat menyajikan data dalam bentuk laporan yang memiliki interval waktu tertentu sesuai yang dibutuhkan sehingga memudahkan pihak BKKBN Kabupaten Cianjur untuk penganalisisan data lebih lanjut.

3.2 Saran

Berikut ini adalah saran yang dapat diberikan untuk pengembangan sistem selanjutnya, yaitu:

1. Sistem tidak hanya menganalisis data Pil KB dan Keluarga Sejahtera saja, tetapi juga menganalisis data kependudukan yang ada di wilayah Kabupaten Cianjur..
2. Sistem dapat secara otomatis melakukan proses *update* ETL pada waktu tertentu, missal setiap tanggal 4 setiap bulannya.

DAFTAR PUSTAKA

[1] P. Lane, Oracle9i Data Warehousing Guide Release 2 (9.2), Redwood: Oracle Corporation, 2002.

[2] M. Y. Pusadan, Rancang Bangun Data Warehouse, Yogyakarta: Graha Ilmu, 2013.

[3] M. Y. Pusadan, Rancang Bangun Data Warehouse, Yogyakarta: Graha Ilmu, 2013.

[4] P. Ponniah, Data Warehousing Fundamentals. A Comprehensif Guide For IT Professionals, John Wiley & Sons, Inc., 2001.

[5] M. Golfarelli dan S. Rizzi, Data Warehouse Design: Modern Principles and Methodologies, The McGraw-Hill Companies, 2009.

[6] W. H. Inmon, Building the Data Warehouse Third Edition, Canada: John Wiley & Sons, Inc., 2002.

[7] B. Knight, D. Knight, M. Davis dan W. Snyder, Microsoft SQL Server 2012 Integration Services, Indianapolis: John Wiley & Sons, Inc, 2013.

[8] D. Browning dan J. Mundy, "Data Warehouse Design Considerations," *MicrosoftSQL2000 TechnicalArticles - MSDN.com*, 2001.