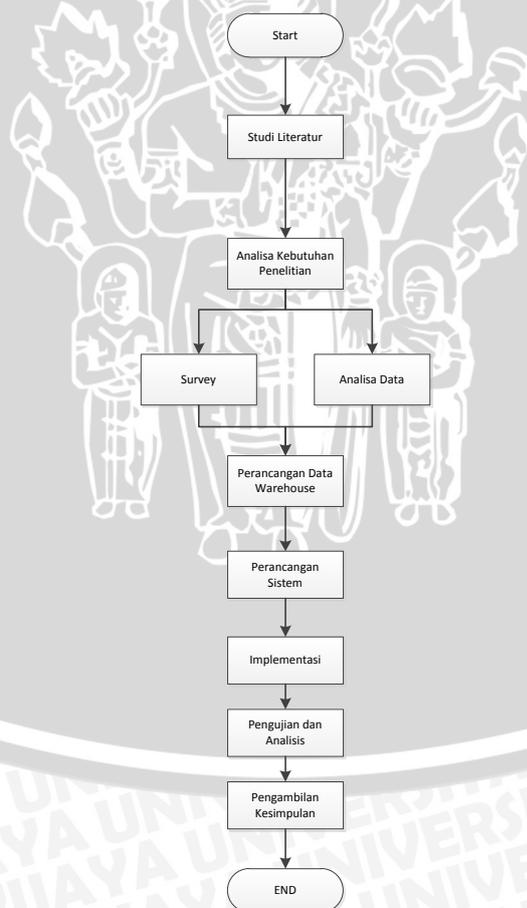


BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN

3.1 Metode Penelitian

Pada bab ini akan dijelaskan langkah – langkah yang akan ditempuh dalam menyusun skripsi ini, yaitu metodologi penelitian dan perancangan pada sistem yang akan dibuat. Rancangan *Data Warehouse* ini dibuat adalah untuk menerapkan optimasi *Online Analytical Processing* dengan pendekatan *Materialized Query Table*. Dimana dengan diterapkannya rancangan *Data Warehouse* untuk dapat memanfaatkan OLAP ini akan dapat mempercepat *request user* untuk evaluasi dari suatu universitas terkait penerimaan jumlah mahasiswa baru dilihat dari akademiknya. Metodologi penelitian yang dilakukan adalah sebagai berikut :



Gambar 3.1 Diagram Blok Penelitian

3.2 Studi Literatur

Langkah awal ini dilakukan untuk mendalami konsep – konsep dan teori – teori yang akan diterapkan pada judul skripsi yang diangkat oleh penulis. Langkah ini sangat penting karena dalam perancangan dan implementasi yang dilakukan akan menggunakan pengetahuan dasar dari beberapa dasar teori yang diperlukan. Beberapa konsep yang perlu dipelajari dalam studi literatur adalah :

- *Database*
- *Data Warehouse*
- *ETL (Extract, Transform, Load)*
- *OLAP (Online Analytical Processing)*
- *OLTP (Online Transaction Processing)*
- *MQT (Materialized Query Table)*

3.3 Analisa Kebutuhan Penelitian

Pada bagian ini akan dilakukan analisa terhadap kebutuhan – kebutuhan yang dibutuhkan dalam pengimplementasian rancangan terhadap sistem yang akan dibuat. Pada tahap ini kita akan mendapatkan data – data yang dibutuhkan untuk menjalankan penelitian ini, Pada tahap analisa kebutuhan penelitian ini terbagi menjadi 2 tahap yaitu tahap *survey* dan analisa data.

3.3.1 Survey

Survey yang dilakukan adalah untuk mengetahui kondisi lapangan yang sesungguhnya seperti bisnis proses apa yang diperlukan. Dari informasi yang didapatkan akan dianalisis untuk menjadi data yang digunakan dalam proses penelitian yang akan dilakukan. Dalam *survey* ini, akan dapat pula kita mengetahui bagaimana skema relasional yang telah ada. Karena penelitian ini adalah untuk menawarkan metode optimasi OLAP dengan pendekatan *Materialized Query Table*.

Untuk kebutuhan informasi, kebutuhan informasi yang akan dibangun adalah :

1. Mengetahui Sebaran Indeks Prestasi Kumulatif dari mahasiswa – mahasiswa yang berasal dari jalur masuk tertentu.
2. Mengetahui sebaran IPK dari mahasiswa – mahasiswa yang berasal dari daerah tertentu.

3.3.2 Analisa

Pada tahap analisa, akan dibagi menjadi 2 macam proses yaitu analisa data dan analisa kebutuhan. Berikut adalah penjelasan masing – masing proses yang ada pada tahap analisa.

3.3.2.1 Analisa Data

Analisa Data adalah proses untuk mengelompokkan, membuat urutan, memanipulasi dan menyingkat data – data sehingga mudah untuk dipahami [ZAA-10]. Pada fase ini, kita akan mendefinisikan kebutuhan informasi yang harus dihasilkan oleh *data warehouse* melalui analisis reporting. Pada penelitian ini, penulis menggunakan basis data akademik (SIKAD) Universitas Brawijaya sebagai studi kasus penelitian ini. *Data Warehouse* yang dirancang sebagai bahan uji coba OLAP adalah berdasarkan *physical diagram Source*. Jumlah data yang digunakan sebagai bahan uji coba akan disesuaikan dengan keadaan pada *database* SIKAD pada waktu tertentu.

3.3.2.2 Analisa Kebutuhan

Untuk proses analisa kebutuhan disini, akan dijelaskan mengenai kebutuhan fungsional dan kebutuhan non-fungsional.

A. Kebutuhan Fungsional

Kebutuhan fungsional yang akan didefinisikan dalam penelitian ini adalah :

1. *Data Warehouse* harus mampu menggambarkan analisa terhadap peningkatan mahasiswa berdasarkan seleksi masuknya.

2. *Data Warehouse* harus mampu menggambarkan analisa terhadap peningkatan mahasiswa berdasarkan daerah asalnya.

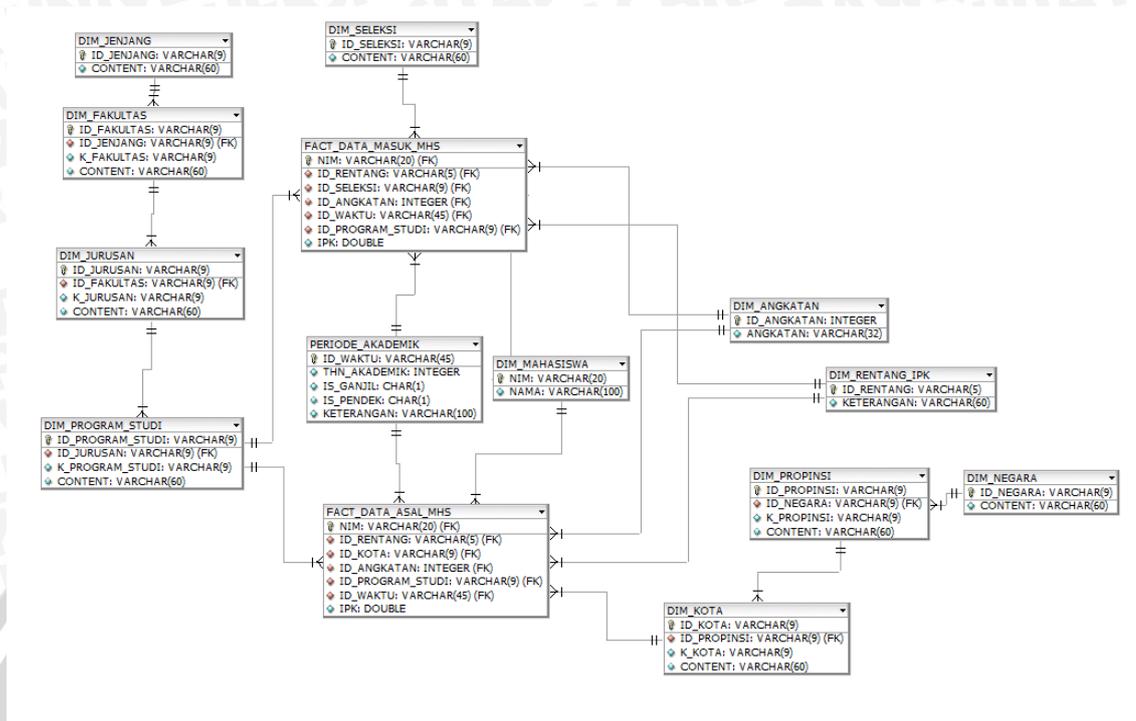
B. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah suatu kebutuhan yang tidak secara langsung terkait pada suatu *fitur* tertentu [ASM-12]. Kebutuhan non-fungsional memberikan batasan pada kebutuhan fungsional. Berikut adalah kebutuhan non-fungsional yang dapat didefinisikan :

1. Untuk pengujian, *data warehouse* dapat diakses dari sisi *client* dengan *IBM Data Studio*. *Log in* cukup dilakukan saat *log in* untuk *windows* dan *log in database* untuk akses data.
2. Spesifikasi minimum yang digunakan adalah *windows 7 service pack 1* dengan *processor Intel Core™ I3* dengan *memory RAM 4 GB*.

3.4 Perancangan Data Warehouse

Pada tahap ini merupakan tahap perancangan *Data Warehouse* yang akan digunakan untuk uji coba OLAP, perancangan ini terlepas dari pengembangan *Data Warehouse* di SIAKAD yang sesungguhnya. Pada penelitian ini metode yang digunakan dalam merancang *Data Warehouse* adalah *Kimball Nine-Step Methodology* yang dikemukakan oleh *Ralph Kimball*. Sembilan langkah ini akan diterapkan pada perancangan *Data Warehouse* untuk menguji optimasi OLAP dengan menggunakan *database SIAKAD*. Penjelasan detail mengenai perancangan *Data Warehouse* ini terdapat pada lampiran, berikut adalah *physical diagram* hasil perancangan *Data Warehouse* yang digunakan untuk *testing OLAP* (Gambar 3.2)



Gambar 3.2 Rancangan Data Warehouse untuk Testing

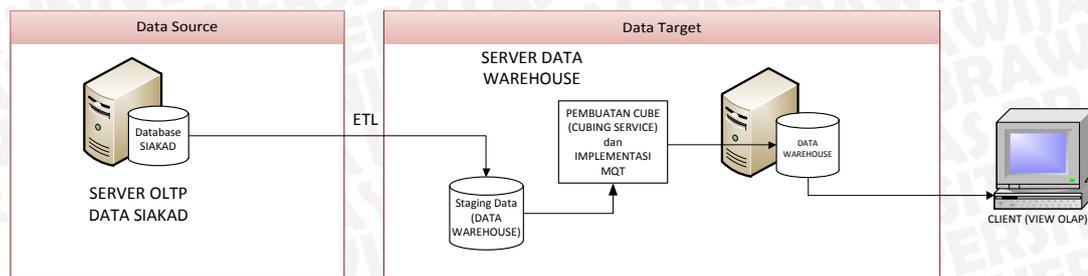
3.5 Perancangan Sistem

Setelah selesai pada tahap perancangan *Data Warehouse*, maka kita akan masuk pada tahap perancangan Sistem. Pada tahap ini dibagi menjadi dua, yaitu perancangan arsitektur fisik dari *Data Warehouse*, dan perancangan arsitektur logis dari *Data Warehouse*. Berikut akan dijelaskan dari kedua perancangan tersebut.

3.5.1 Rancangan Arsitektur Fisik dari Data Warehouse

Setelah didapat analisa kebutuhan untuk merancang *data warehouse*, maka sekarang masuk pada perancangan arsitektur sesuai kebutuhan yang ada. Rancangan arsitektur fisiknya adalah seperti ditunjukkan pada gambar 3.9.





Gambar 3.3 Rancangan Arsitektur Fisik Data Warehouse

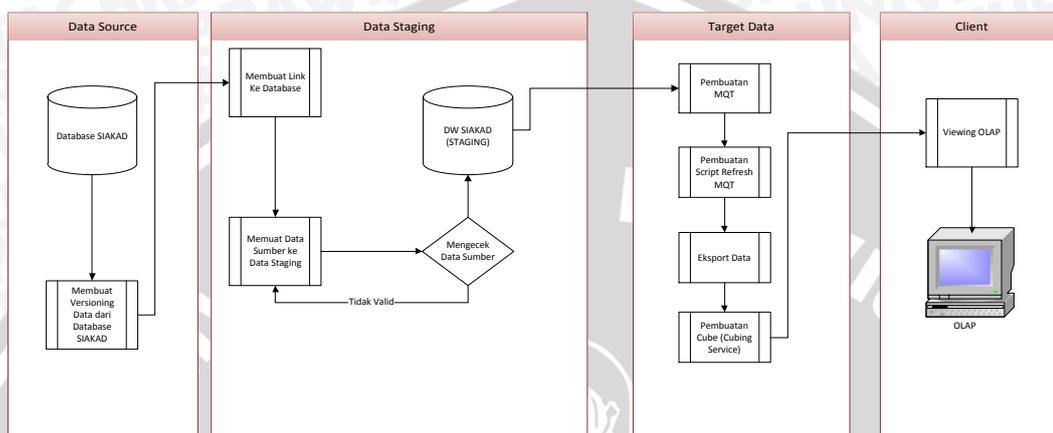
Sesuai dengan gambar rancangan arsitektur pada gambar 3.9, kita akan membuat *versioning data* terlebih dahulu dari *database SIAKAD* yang digunakan. Data diambil dari server OLTP (*Online Analytical Processing*), kemudian setelah didapatkan *database SIAKAD*, maka kita akan membuat *versioning Data* pada tiap tabel yang digunakan pada OLTP. Hal ini dimaksudkan untuk mengetahui apabila terjadi perubahan data atau *delete* data pada *database SIAKAD* OLTP. Sehingga proses ETL (*Extract, Transform, Load*) ke *staging data* adalah untuk data – data yang berubah saja, baik itu *insert, update, delete*, sehingga tidak perlu deploy keseluruhan data lagi. Kemudian dari *staging data* tersebut, kita lakukan *cubing service* dan implementasi MQT untuk proses pembuatan *cube* yang dimaksudkan untuk membuat rancangan OLAP. Setelah semuanya selesai, maka *client* mengakses *cube* yang telah ada untuk melihat hasil OLAP.

Singkatnya, tahapan yang terjadi pada perancangan arsitektur fisik *data warehouse* ini adalah sebagai berikut :

1. Data SIAKAD diambil dari server OLTP yang ada sebagai source untuk database SIAKAD.
2. Melakukan Proses ETL ke Staging Data
3. Melakukan proses pembuatan *cube* (*cubing service*) dan implementasi MQT.
4. Uji Performansi OLAP melalui sisi Client.

3.5.2 Rancangan Arsitektur Logis Data Warehouse

Setelah didapatkan arsitektur fisik kemudian dirancang arsitektur logis untuk melihat proses logis *data warehouse*, dibawah ini terdapat 2 rancangan arsitektur logis dari *data warehouse* untuk penelitian ini, seperti pada gambar 3.10 dan 3.11.



Gambar 3.4 Rancangan Arsitektur Logis Data Warehouse

Sesuai dengan gambar rancangan arsitektur logis *data warehouse* di gambar 3.10, terdapat 4 tahapan dalam proses *Data Warehouse* ini, berikut penjelasannya adalah sebagai berikut :

1. Data Source

Pada tahap ini, merupakan *data source* yaitu pemrosesan pada sumber data yang akan digunakan dalam implementasi penelitian ini. *Data source* berasal dari *Database* Siakad dimana digabung dengan pembuatan *Versioning Data* dari *Database* Siakad tersebut. Oleh sebab itu, terdapat proses pembuatan *versioning* data terhadap *database* SIAKAD. Hal ini dimaksudkan untuk dapat mengetahui apabila terjadi proses perubahan maupun penghapusan data pada data sumber. Sehingga bila dilakukan proses ETL yang kedua kalinya, yang di Load tidak perlu keseluruhan data lagi, namun cukup data yang berubah saja berdasarkan kurun waktu tertentu, tergantung proses *maintenance* terhadap *Data Warehouse* yang ada.

2. Data Staging

Setelah proses pada *data source* selesai, maka pada tahapan disini dimulai dari proses membuat *link* ke *database* sumber yang telah disiapkan sebelumnya.

Setelah *link* berhasil dibuat ke data sumber maka proses yang dilakukan adalah proses memasukkan data ke *data staging*. Proses pengecekan ini dapat dilakukan dengan menggunakan proses ETL yang disiapkan. Setelah data berhasil dimuat pada tahap *staging*, maka kita buat script MQT dan *script refresh* MQT. Setelah semua selesai dibuat maka dari sisi client membuat *cube* dari hasil eksport data yang dilakukan dengan menggunakan MQT tadi, kemudian hasil OLAP dapat dilihat dari berbagai bentuk grafik.

Terdapat 2 *fact table* yang disediakan pada target data, berikut merupakan perancangan *query* melalui aljabar relasionalnya dari kedua *table fact* tersebut.

a. Aljabar relasional sebagai dasar untuk FACT_DATA_MASUK_MHS

Berikut adalah rancangan query melalui aljabar relasional untuk dasar pembuatan table fakta pada FACT_DATA_MASUK_MHS

```
JEN_FK<=<math>\sigma</math> FAKULTAS.K_JENJANG = JENJANG.K_JENJANG (FAKULTAS X JENJANG)

FK_JUR <=<math>\sigma</math> FAKULTAS.K_FAKULTAS = JURUSAN.K_FAKULTAS ^ FAKULTAS.K_JENJANG =
JURUSAN.K_JENJANG (JEN_FK X JURUSAN)

FK_JUR_PR <=<math>\sigma</math> JURUSAN.ID_FAKULTAS = PROGRAM_STUDI.K_FAKULTAS ^
JURUSAN.K_JURUSAN = PROGRAM_STUDI.K_JURUSAN ^ JURUSAN.K_JENJANG =
PROGRAM_STUDI.K_JENJANG (PROGRAM_STUDI X FK_JUR)

MHS_STS <=<math>\sigma</math> MAHASISWA.K_PROGRAM_STUDI = PROGRAM_STUDI.K_PROGRAM_STUDI
^
MAHASISWA.K_JURUSAN = PROGRAM_STUDI.K_JURUSAN ^
MAHASISWA.K_FAKULTAS = PROGRAM_STUDI.K_FAKULTAS
^ MAHASISWA.K_JENJANG = PROGRAM_STUDI.K_JENJANG (FK_JUR_PR X MAHASISWA)

MHS_STS_SELEKSI <=<math>\sigma</math> MAHASISWA.K_SELEKSI = SELEKSI.K_SELEKSI (MHS_STS X SELEKSI)

MHS_TB_MHS_AKADEMIK <=<math>\sigma</math> MHS_AKADEMIK.NIM = MAHASISWA.NIM
(MHS_STS_SELEKSI X MHS_AKADEMIK)

MHS_RESULTS (NIM, ANGKATAN, FAKULTAS, JURUSAN, PROGRAM_STUDI, JENJANG,
SELEKSI, IPK, IS_GANJIL, IS_PENDEK) <=<math>\pi</math> MAHASISWA.NIM, MAHASISWA.ANGKATAN,
FAKULTAS.CONTENT, JURUSAN.CONTENT, PROGRAM_STUDI.CONTENT, JENJANG.CONTENT,
SELEKSI.CONTENT, MHS_AKADEMIK.IPK_LULUS, MHS_AKADEMIK.IS_GANJIL,
MHS_AKADEMIK.IS_PENDEK (MHS_TB_MHS_AKADEMIK)
```

Gambar 3.5 Rancangan Aljabar Relasional Seleksi Masuk Mahasiswa

Aljabar Relasional diatas merupakan rancangan query untuk dapat menampilkan nim mahasiswa, angkatan mahasiswa, fakultas, jurusan, program studi, jenjang, seleksi, is_ganjil, is_pendek beserta IPK nya. dari dasar rancangan query tersebut akan bisa kita dapatkan nantinya untuk memenuhi proses di *Data*

Warehouse, yaitu untuk dapat menampilkan IPK terendah, tertinggi, dan rata – rata berdasarkan angkatan dan periode tertentu, serta menampilkan jumlah mahasiswa pada rentang IPK tertentu.

b. Aljabar relasional sebagai dasar untuk FACT_DATA_ASAL_MHS

Berikut adalah rancangan query melalui aljabar relasional untuk dasar pembuatan tabel fakta pada FACT_DATA_ASAL_MHS :

```
JEN_FK<=<σ FAKULTAS.K_JENJANG = JENJANG.K_JENJANG (FAKULTAS X JENJANG)

FK_JUR <=<σ FAKULTAS.K_FAKULTAS = JURUSAN.K_FAKULTAS ^ FAKULTAS.K_JENJANG =
JURUSAN.K_JENJANG (JEN_FK X JURUSAN)

FK_JUR_PR <=<σ JURUSAN.ID_FAKULTAS = PROGRAM_STUDI.K_FAKULTAS ^
JURUSAN.K_JURUSAN = PROGRAM_STUDI.K_JURUSAN ^ JURUSAN.K_JENJANG =
PROGRAM_STUDI.K_JENJANG (PROGRAM_STUDI X FK_JUR)

MHS_STS <=<σ MAHASISWA.K_PROGRAM_STUDI = PROGRAM_STUDI.K_PROGRAM_STUDI
^
MAHASISWA.K_JURUSAN = PROGRAM_STUDI.K_JURUSAN ^
MAHASISWA.K_FAKULTAS = PROGRAM_STUDI.K_FAKULTAS
^ MAHASISWA.K_JENJANG = PROGRAM_STUDI.K_JENJANG (FK_JUR_PR X MAHASISWA)

MHS_MHS_BIODATA <=<σ MAHASISWA.NIM = MAHASISWA_BIODATA.NIM (MHS_STS X
MAHASISWA_BIODATA)

MHS_KOTA <=<σ MAHASISWA_BIODATA.K_KOTA_ASAL = KOTA.K_KOTA ^
MAHASISWA_BIODATA.K_PROPINSI_ASAL = KOTA.K_PROPINSI ^
MAHASISWA_BIODATA.K_NEGARA_ASAL = KOTA.K_NEGARA (MHS_MHS_BIODATA X KOTA)

MHS_KT_PROP <=<σ KOTA.K_PROPINSI = PROPINSI.K_PROPINSI ^
KOTA.K_NEGARA = PROPINSI.K_NEGARA (MHS_KOTA X PROPINSI)

MHS_KT_PROP_NEGARA <=<σ PROPINSI.K_NEGARA = NEGARA.K_NEGARA (MHS_KT_PROP
X NEGARA)

MHS_TB_MHS_AKADEMIK <=<σ MHS_AKADEMIK.NIM = MAHASISWA.NIM (MHS_KT_PROP
_NEGARA X MHS_AKADEMIK)

MHS_RESULTS (NIM, ANGKATAN, NEGARA, PROPINSI, KOTA, IPK, IS_GANJIL, IS_PENDEK,
JENJANG, FAKULTAS, JURUSAN, PROGRAM_STUDI) <=<π MAHASISWA.NIM,
MAHASISWA.ANGKATAN, NEGARA.CONTENT, PROPINSI.CONTENT, KOTA.CONTENT,
MHS_AKADEMIK.IPK_LULUS, MHS_AKADEMIK.IS_GANJIL, MHS_AKADEMIK.IS_PENDEK,
JENJANG.CONTENT, FAKULTAS.CONTENT, JURUSAN.CONTENT, PROGRAM_STUDI.CONTENT
(MHS_TB_MHS_AKADEMIK)
```

Gambar 3.6 Rancangan Aljabar Relasional Asal Mahasiswa

Aljabar Relasional diatas merupakan rancangan *query* untuk dapat menampilkan nim mahasiswa, angkatan mahasiswa, negara asal, propinsi asal, kota asal, is_ganjil, is_pendek beserta IPK nya. dari dasar rancangan *query* tersebut akan bisa kita dapatkan nantinya untuk memenuhi proses di *Data*

Warehouse, yaitu untuk dapat menampilkan IPK terendah, tertinggi, dan rata – rata berdasarkan angkatan dan periode tertentu, serta menampilkan jumlah mahasiswa pada rentang IPK tertentu.

3. Target Data

Pada proses ini merupakan proses persiapan pada target data, karena tujuan penelitian ini adalah untuk mengoptimasi OLAP dengan pendekatan *materialized view* atau *materialized query table*, maka penulis merancang pembuatan *fact table* dengan menggunakan MQT, yang nanti pada analisa di client akan dibandingkan hasilnya dengan tanpa menggunakan MQT. OLAP akan ditampilkan dengan tools *IBM Cognos Insight*. Jika kita menggunakan tools *IBM Cognos Insight*, maka MQT dibuat sebelum proses *Cubing*, karena MQT digunakan untuk *ekspor* data ke format *csv* yang nantinya akan digunakan untuk OLAP pada *IBM Cognos Insight*.

4. Client

Pada tahap ini merupakan uji performansi OLAP dari *Data Warehouse* pada sisi client yang telah selesai dibuat pada tahapan sebelumnya. Uji dan analisis akan dibahas pada bab selanjutnya.

3.6 Implementasi

Implementasi yang dilakukan dalam penelitian ini adalah proses pengimplementasian dari sisi OLTP hingga pengujian OLAP. Dimulai dari implementasi tabel – tabel yang ada pada sisi OLTP, implementasi tabel – tabel yang ada pada sisi *Data Warehouse*, kemudian *Store Procedure*, *ETL*, *data flow*, *control flow*, MQT, hingga implementasi pada OLAP.

3.7 Pengujian dan Analisis

Pengujian yang dilakukan adalah pengujian performansi pada MQT yang diimplementasikan dalam penelitian ini dengan cara membandingkan waktu eksekusi export data untuk OLAP dengan menggunakan MQT dan query biasa.

3.8 Pengambilan Kesimpulan

Kesimpulan yang diambil dalam penelitian ini adalah berdasarkan hasil pengujian, analisis, dan implementasi yang telah diterapkan.

