



**PENGEMBANGAN APLIKASI DATA WAREHOUSE PRESTASI  
MAHASISWA (STUDI PADA: FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Sausan Zahrah  
NIM: 165150700111023



**PROGRAM STUDI TEKNOLOGI INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA**

**MALANG  
2020**



# PENGESAHAN

PENGEMBANGAN APLIKASI DATA WAREHOUSE PRESTASI MAHASISWA (STUDI PADA: FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Sausan Zahrah

NIM: 165150700111023

Skripsi ini telah diuji dan dinyatakan lulus pada  
14 Mei 2020

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Issa Arwani, S.Kom., M.Sc.  
NIP: 198309222012121003

Satrio Agung Wicaksono, S.Kom., M.Kom.  
NIP: 198605212012121001

Mengetahui

Ketua Jurusan Sistem Informasi



Dr. Eng. Herman Tolle, S.T., M.T.

NIP: 197408232000121001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 22 April 2020



Sausan Zahrah

NIM: 165150700111023



## ABSTRAK

**Sausan Zahrah, Pengembangan Aplikasi *Data Warehouse* Prestasi Mahasiswa (Studi Pada: Fakultas Ilmu Komputer Universitas Brawijaya)**

**Pembimbing: Issa Arwani, S.Kom., M.Sc., dan Satrio Agung Wicaksono, S.Kom., M.Kom.**

Akreditasi program studi dilakukan setiap lima tahun sekali. Dalam proses akreditasi, terdapat beberapa parameter yang digunakan untuk penilaian akreditasi, di antaranya adalah pencapaian mahasiswa yang salah satunya dilihat dari aspek prestasi. Pada Fakultas Ilmu Komputer, data prestasi yang diperoleh mahasiswa disimpan oleh Bagian Kemahasiswaan. Sedangkan data akademik mahasiswa disimpan oleh Bagian Akademik. Dalam melakukan rekapitulasi data prestasi dan akademik, pihak program studi pada Fakultas Ilmu Komputer mendapatkan data dari dua sumber yang berbeda, yaitu Bagian Kemahasiswaan dan Bagian Akademik. Untuk mempermudah proses rekapitulasi data dari dua sumber tersebut, diperlukan sebuah *data warehouse* sehingga data dari sumber yang berbeda dapat diakses pada waktu yang bersamaan.

Dalam membangun *data warehouse*, tahap perancangan yang dilakukan mengacu pada *Nine Step Methodology*. Dari perancangan dan implementasi yang dilakukan, dihasilkan 6 tabel dimensi dengan 3 tabel fakta, yaitu tabel fakta Prestasi, Lulusan dan Mahasiswa Aktif. Kemudian dilakukan pengujian validasi yang menghasilkan status valid pada seluruh kasus uji, lalu dilakukan pengujian performa ETL pada segi waktu dengan hasil waktu rata-rata yang dibutuhkan untuk eksekusi proses ETL adalah  $\pm 11$  detik dengan total jumlah data sebanyak 32.746 data.

Kata kunci: *data warehouse*, ETL (*Extract, Transform, Load*), OLAP (*Online Analytical Processing*), prestasi mahasiswa



## ABSTRACT

**Sausan Zahrah, *The Development of Student Achievement Data Warehouse (Case on: Faculty of Computer Science Universitas Brawijaya)***

**Supervisors: Issa Arwani, S.Kom., M.Sc., and Satrio Agung Wicaksono, S.Kom., M.Kom.**

*Study program accreditation is done every five years. In the accreditation process, there are several parameters used for accreditation assessment, including student achievement, one of which is seen from the aspect of achievement. At the Faculty of Computer Science, student achievement data is stored by the Student Affairs Section. Whereas student academic data is stored by the Academic Section. In recapitulating achievement and academic data, the study program at the Faculty of Computer Science obtains data from two different sources, they are the Student Affairs Section and Academic Section. To simplify the process of recapitulating data from the two sources, a data warehouse is needed so that data from different sources can be accessed at the same time.*

*In building a data warehouse, the design phase that has been done refers to the Nine Step Methodology. From the design and implementation phase, it produced 6 dimension tables with 3 fact tables, they are the fact table Achievement, Graduates and Active Students. Then, a validation test is performed which results in a valid status in all test cases, and then the ETL performance testing in terms of time is done with the average time required for the execution of the ETL process is  $\pm 11$  seconds with a total amount of 32,746 data.*

**Keywords:** *data warehouse, ETL (Extract, Transform, Load), OLAP (Online Analytical Processing), student achievement*



**DAFTAR ISI**

PENGESAHAN..... ii

PERNYATAAN ORISINALITAS..... iii

PRAKATA..... iv

ABSTRAK..... v

ABSTRACT..... vi

DAFTAR ISI..... vii

DAFTAR TABEL..... x

DAFTAR GAMBAR..... xi

DAFTAR LAMPIRAN.....xiv

BAB 1 PENDAHULUAN..... 1

    1.1 Latar Belakang..... 1

    1.2 Rumusan Masalah..... 2

    1.3 Tujuan..... 2

    1.4 Manfaat..... 3

    1.5 Batasan Masalah..... 3

    1.6 Sistematika Pembahasan..... 3

BAB 2 LANDASAN KEPUSTAKAAN..... 5

    2.1 Kajian Pustaka..... 5

    2.2 Landasan Teori..... 6

        2.2.1 Fakultas Ilmu Komputer Universitas Brawijaya..... 6

        2.2.2 *Online Analytical Processing (OLAP)*..... 6

        2.2.3 Data..... 7

        2.2.4 Informasi..... 7

        2.2.5 *Database (Basis Data)*..... 7

        2.2.6 *Database Management System (DBMS)*..... 8

        2.2.7 *Data Warehouse*..... 8

        2.2.8 *Extract, Transform, Load (ETL)*..... 16

        2.2.9 *Talend Open Studio for Data Integration*..... 18

        2.2.10 *Jedox*..... 18

        2.2.11 *Pengujian*..... 19

BAB 3 METODOLOGI..... 20

3.1 Observasi dan Wawancara.....	20
3.2 Studi Literatur.....	21
3.3 Analisis Data.....	21
3.4 Perancangan.....	21
3.4.1 Arsitektur <i>Data Warehouse</i> .....	21
3.4.2 Perancangan <i>Worksheet</i> .....	22
3.4.3 Perancangan Data Penelitian.....	22
3.4.4 Perancangan <i>Data Warehouse</i> .....	22
3.4.5 Perancangan Proses ETL ( <i>Extract, Transform, Load</i> ).....	23
3.5 Implementasi.....	24
3.5.1 Implementasi <i>Data Warehouse</i> .....	24
3.5.2 Proses ETL ( <i>Extract, Transform, Load</i> ).....	24
3.5.3 Implementasi OLAP ( <i>Online Analytical Processing</i> ).....	24
3.6 Pengujian.....	24
3.6.1 Pengujian Validasi.....	24
3.6.2 Pengujian Performa Proses ETL ( <i>Extract, Transform, Load</i> ).....	24
3.7 Kesimpulan dan Saran.....	25
BAB 4 HASIL DAN PEMBAHASAN.....	26
4.1 Hasil Analisis Data.....	26
4.2 Perancangan.....	26
4.2.1 Arsitektur <i>Data Warehouse</i> .....	26
4.2.2 Perancangan <i>Worksheet</i> .....	27
4.2.3 Perancangan Data Penelitian.....	28
4.2.4 Perancangan <i>Data Warehouse</i> .....	28
4.2.5 Perancangan Proses ETL.....	34
4.3 Implementasi.....	41
4.3.1 Implementasi <i>Data Warehouse</i> .....	41
4.3.2 Proses ETL ( <i>Extract, Transform, Load</i> ).....	49
4.3.3 Implementasi OLAP.....	68
4.4 Pengujian.....	79
4.4.1 Pengujian Validasi.....	79
4.4.2 Pengujian Performa Proses ETL ( <i>Extract, Transform, Load</i> ).....	80



BAB 5 PENUTUP.....	82
5.1 Kesimpulan.....	82
5.2 Saran.....	82
LAMPIRAN A HASIL WAWANCARA.....	83
LAMPIRAN B HASIL PENGUJIAN.....	85
DAFTAR REFERENSI.....	88





## DAFTAR TABEL

Tabel 4.1 Diagram <i>information package</i> Prestasi Mahasiswa.....	29
Tabel 4.2 Diagram <i>information package</i> Mahasiswa Lulus.....	29
Tabel 4.3 Diagram <i>information package</i> Mahasiswa Aktif.....	29
Tabel 4.4 Fakta yang dipilih.....	30
Tabel 4.5 Dimensi Program Studi.....	31
Tabel 4.6 Dimensi Angkatan.....	32
Tabel 4.7 Dimensi Tahun.....	32
Tabel 4.8 Dimensi Juara.....	32
Tabel 4.9 Dimensi Tingkat Lomba.....	33
Tabel 4.10 Dimensi Tahun Akademik.....	33
Tabel 4.11 Tabel kebutuhan <i>backup</i> .....	34
Tabel 4.12 Implementasi DDL untuk <i>buffer pool</i> dan <i>table space</i> .....	42
Tabel 4.13 Implementasi DDL untuk dimensi Juara.....	42
Tabel 4.14 Implementasi DDL untuk dimensi Tingkat Lomba.....	43
Tabel 4.15 Implementasi DDL untuk dimensi Tahun.....	43
Tabel 4.16 Implementasi DDL untuk dimensi Program Studi.....	44
Tabel 4.17 Implementasi DDL untuk dimensi Angkatan.....	45
Tabel 4.18 Implementasi DDL untuk dimensi Tahun Akademik.....	45
Tabel 4.19 Implementasi DDL untuk fakta Prestasi Mahasiswa.....	46
Tabel 4.20 Implementasi DDL untuk fakta Lulusan.....	47
Tabel 4.21 Implementasi DDL untuk fakta Mahasiswa Aktif.....	48
Tabel 4.22 Komponen pada implementasi proses ETL fakta Prestasi.....	55
Tabel 4.23 Komponen pada implementasi proses ETL fakta Lulusan.....	58
Tabel 4.24 Komponen pada implementasi proses ETL fakta Mahasiswa Aktif.....	62
Tabel 4.25 Pengaturan pada komponen tPaloCube.....	69
Tabel 4.26 Komponen untuk proses penulisan data pada cube.....	71
Tabel 4.27 Hasil pengujian validasi.....	79
Tabel 4.28 Hasil pengujian performa ETL.....	80
Tabel 4.29 Hasil pengujian performa proses ETL secara keseluruhan.....	81



## DAFTAR GAMBAR

Gambar 2.1 Topologi <i>data warehouse</i> .....	8
Gambar 2.2 Aspek orientasi subyek dari <i>data warehouse</i> .....	9
Gambar 2.3 Aspek integrasi data pada <i>data warehouse</i> .....	10
Gambar 2.4 Aspek <i>nonvolatile</i> pada <i>data warehouse</i> .....	10
Gambar 2.5 Aspek <i>time variant</i> pada <i>data warehouse</i> .....	11
Gambar 2.6 <i>Star Schema</i> .....	12
Gambar 2.7 Arsitektur dasar <i>data warehouse</i> .....	14
Gambar 2.8 Notasi pada pemodelan konseptual ETL.....	17
Gambar 2.9 Contoh diagram pemodelan konseptual.....	18
Gambar 3.1 Metodologi penelitian.....	20
Gambar 4.1 Arsitektur <i>data warehouse</i> .....	26
Gambar 4.2 <i>Worksheet</i> Prestasi Mahasiswa.....	27
Gambar 4.3 <i>Worksheet</i> Akademik Mahasiswa (Lulusan dan Mahasiswa Aktif) ..	28
Gambar 4.4 <i>Star Schema</i> Prestasi Mahasiswa .....	30
Gambar 4.5 <i>Star Schema</i> Lulusan.....	31
Gambar 4.6 <i>Star Schema</i> Mahasiswa Aktif.....	31
Gambar 4.7 Pemetaan data dimensi Program Studi.....	35
Gambar 4.8 Pemetaan data dimensi Angkatan.....	35
Gambar 4.9 Pemetaan data dimensi Tahun.....	36
Gambar 4.10 Pemetaan data dimensi Juara.....	36
Gambar 4.11 Pemetaan data dimensi Tingkat Lomba.....	37
Gambar 4.12 Pemetaan data dimensi Tahun Akademik.....	37
Gambar 4.13 Pemetaan data fakta Prestasi Mahasiswa.....	38
Gambar 4.14 Pemetaan data fakta Lulusan.....	40
Gambar 4.15 Pemetaan data fakta Mahasiswa Aktif.....	41
Gambar 4.16 Implementasi proses ETL tabel DIM_PROD.....	50
Gambar 4.17 Implementasi proses ETL tabel DIM_TINGKAT.....	51
Gambar 4.18 Implementasi proses ETL tabel DIM_TAHUN .....	52
Gambar 4.19 Implementasi proses ETL tabel DIM_ANGKATAN .....	53
Gambar 4.20 Implementasi proses ETL tabel DIM_TAHUN_AJAR.....	54
Gambar 4.21 Implementasi proses ETL tabel DIM_JUARA.....	55



Gambar 4.22 Implementasi proses ETL fakta Prestasi .....	57
Gambar 4.23 Pemetaan data pertama pada fakta Prestasi .....	57
Gambar 4.24 Agregasi pada fakta Prestasi .....	58
Gambar 4.25 Pemetaan data kedua pada fakta Prestasi .....	58
Gambar 4.26 Implementasi proses ETL fakta Lulusan .....	61
Gambar 4.27 Pemetaan data pertama pada fakta Lulusan.....	61
Gambar 4.28 Agregasi pada fakta Lulusan.....	61
Gambar 4.29 Pemetaan data kedua pada fakta Lulusan.....	62
Gambar 4.30 Implementasi proses ETL fakta Mahasiswa Aktif.....	64
Gambar 4.31 Pemetaan data pertama pada fakta Mahasiswa Aktif.....	64
Gambar 4.32 Agregasi pada fakta Mahasiswa Aktif.....	65
Gambar 4.33 Pemetaan data kedua pada fakta Mahasiswa Aktif .....	65
Gambar 4.34 Waktu eksekusi yang dipilih untuk penjadwalan proses ETL fakta Prestasi .....	66
Gambar 4.35 Waktu eksekusi yang dipilih untuk penjadwalan proses ETL fakta Mahasiswa Aktif .....	67
Gambar 4.36 Waktu eksekusi yang dipilih untuk penjadwalan proses ETL fakta Lulusan .....	68
Gambar 4.37 Pembuatan dimensi pada OLAP untuk fakta Prestasi .....	69
Gambar 4.38 Pembuatan dimensi pada OLAP untuk fakta Lulusan dan Mahasiswa Aktif.....	69
Gambar 4.39 Komponen untuk membuat <i>cube</i> .....	70
Gambar 4.40 Pengaturan untuk membuat <i>cube</i> Prestasi .....	70
Gambar 4.41 Pengaturan untuk membuat <i>cube</i> Lulusan .....	71
Gambar 4.42 Pengaturan untuk membuat <i>cube</i> Mahasiswa.....	71
Gambar 4.43 Penulisan data pada <i>cube</i> Prestasi.....	73
Gambar 4.44 Pemetaan data dari <i>data warehouse</i> menuju OLAP .....	73
Gambar 4.45 Konversi tipe data sebelum data ditulis pada <i>cube</i> Prestasi.....	74
Gambar 4.46 Pemeriksaan elemen yang terdapat pada tabel fakta Prestasi .....	74
Gambar 4.47 Penulisan data pada <i>cube</i> Lulusan .....	74
Gambar 4.48 Pemetaan data dari <i>data warehouse</i> menuju OLAP .....	75
Gambar 4.49 Konversi tipe data sebelum data ditulis pada <i>cube</i> Lulusan.....	75
Gambar 4.50 Pemeriksaan elemen yang terdapat pada tabel fakta Lulusan.....	75



Gambar 4.51 Penulisan data pada *cube* Mahasiswa..... 76

Gambar 4.52 Pemetaan data dari *data warehouse* menuju OLAP..... 76

Gambar 4.53 Konversi tipe data sebelum data ditulis pada *cube* Mahasiswa..... 76

Gambar 4.54 Pemeriksaan elemen yang terdapat pada tabel fakta Mahasiswa Aktif ..... 77

Gambar 4.55 *User report* Lulusan ..... 78

Gambar 4.56 *User report* Prestasi Mahasiswa..... 78

Gambar 4.57 *User report* Mahasiswa Aktif..... 79



### DAFTAR LAMPIRAN

Lampiran 1 Naskah wawancara .....	84
Lampiran 2 Pengujian validasi fakta Prestasi .....	85
Lampiran 3 Pengujian validasi fakta Lulusan .....	86
Lampiran 4 Pengujian validasi fakta Mahasiswa Aktif .....	87



## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Teknologi informasi dapat membantu organisasi dalam memperoleh informasi penting yang dibutuhkan dengan lebih cepat dan tepat. Sumber informasi yang tepat dan cepat dapat membantu organisasi dalam mengambil keputusan (Mujiono & Musdholifah, 2016). Teknologi yang diusulkan untuk mendukung pengambilan keputusan adalah *data warehouse* (Connolly & Begg, 2015). Dengan *data warehouse*, data rekapitulasi dapat disimpan dan ditampilkan sesuai kebutuhan.

Fakultas Ilmu Komputer merupakan salah satu fakultas di Universitas Brawijaya yang menjalankan proses akreditasi pada setiap program studi yang dinaunginya. Pada Fakultas Ilmu Komputer terdapat dua sub organisasi yang memiliki peran sebagai unit pelaksana administrasi fakultas yang menyelenggarakan pelayanan administrasi akademik dan kemahasiswaan, yaitu Bagian Akademik dan Bagian Kemahasiswaan. Dua sub organisasi tersebut memiliki tugas pokok serta fungsi yang berbeda. Bagian Kemahasiswaan memiliki tugas untuk melakukan administrasi kemahasiswaan dan memiliki salah satu fungsi, yaitu menghimpun dan mengolah data di bidang kemahasiswaan. Sedangkan Bagian Akademik memiliki tugas untuk melaksanakan administrasi akademik dimana salah satu fungsinya adalah menghimpun dan mengklasifikasikan data di bidang kurikulum, silabus dan mahasiswa beserta prestasi belajarnya.

Akreditasi program studi pada perguruan tinggi negeri dilakukan secara rutin setiap lima tahun sekali. Dalam melakukan akreditasi, terdapat beberapa parameter penilaian yang harus diisi. Salah satu parameter penilaian dalam akreditasi tersebut adalah pencapaian mahasiswanya, baik prestasi akademik, non-akademik, serta publikasi ilmiah dan penelitian yang dihasilkan oleh mahasiswa. Pencapaian mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya dapat dikatakan cukup banyak. Setiap prestasi yang ditorehkan oleh mahasiswa didata dan disimpan pada Bagian Kemahasiswaan Fakultas Ilmu Komputer. Dalam melakukan rekapitulasi data yang akan dilaporkan pada saat akreditasi program studi, pihak program studi membutuhkan data dari Bagian Kemahasiswaan dan Bagian Akademik dimana data prestasi dan akademik mahasiswa diolah dari sumber data yang berbeda sehingga menyebabkan tidak adanya sinkronisasi antara data kemahasiswaan dengan data akademik. Berdasarkan hasil wawancara dengan Bagian Kemahasiswaan Fakultas Ilmu Komputer, hal ini dinilai kurang efisien karena proses pengambilan dan rekapitulasi data dilakukan dari sumber data yang berbeda. Dalam menangani masalah tersebut, *data warehouse* dapat digunakan untuk menyimpan data dari berbagai sumber yang berbeda sehingga data yang bersumber dari Bagian Kemahasiswaan dan Bagian Akademik dapat saling terintegrasi. Data dari kedua sumber tersebut merupakan data selama



empat tahun terakhir yang dimuat ke dalam *data warehouse* secara periodik sesuai dengan jadwal yang ditentukan. Sehingga ketika pihak program studi membutuhkan data prestasi dan akademik mahasiswa, pihak tersebut dapat mengakses satu sumber data saja dimana nantinya digunakan juga teknologi OLAP yang dapat membantu pengguna dalam menganalisis data prestasi dan akademik mahasiswa dengan menampilkan informasi dalam bentuk *cube* dan informasi nilai akhir pada parameter prestasi mahasiswa dalam proses penilaian akreditasi.

*Data warehouse* merupakan kumpulan data berorientasi subyek, terintegrasi dan bersifat *time variant* serta *non volatile* (Connolly & Begg, 2015). *Data warehouse* yang dipadukan dengan teknologi OLAP dapat mendukung analisis dan pengelolaan data sehingga memungkinkan pengguna untuk mengakses informasi secara cepat.

Dalam merancang dan membangun *data warehouse*, diperlukan struktur data yang baik agar *data warehouse* yang dibangun dapat menghasilkan informasi yang sesuai dengan kebutuhan. Untuk itu, dalam proses perancangannya digunakan metodologi perancangan *data warehouse* yang dirumuskan oleh Kimball, yaitu *Nine Step Design Methodology*. Metode tersebut digunakan karena berorientasi kepada proses bisnis (Kimball & Ross, 2002) sehingga diharapkan dapat menghasilkan informasi yang sesuai dengan kebutuhan akreditasi.

Dari permasalahan yang telah dijabarkan, diangkat penelitian dengan judul "Pengembangan Aplikasi *Data Warehouse* Prestasi Mahasiswa (Studi Pada: Fakultas Ilmu Komputer Universitas Brawijaya)". Diharapkan pada penelitian ini dapat menghasilkan sebuah *data warehouse* yang dapat digunakan untuk merekapitulasi data prestasi mahasiswa pada Bagian Kemahasiswaan dan data akademik mahasiswa pada Bagian Akademik.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang dikemukakan, didapatkan rumusan masalah sebagai berikut:

1. Bagaimana hasil perancangan *data warehouse* prestasi mahasiswa?
2. Bagaimana hasil validasi terkait tingkat akurasi data akademik dan data kemahasiswaan Fakultas Ilmu Komputer Universitas Brawijaya yang dihasilkan?
3. Bagaimana hasil performa proses ETL (*Extract, Transform, Load*) pada *data warehouse* prestasi mahasiswa?
4. Bagaimana hasil pengolahan informasi pada OLAP (*Online Analytical Processing*) yang diolah sesuai dengan standar akreditasi?

## 1.3 Tujuan

Pelaksanaan penelitian ini bertujuan untuk:

1. Melakukan perancangan *data warehouse* prestasi mahasiswa dengan menggunakan *Nine Step Design Methodology*.
2. Mengukur validitas data yang dihasilkan oleh *data warehouse*.



3. Mengukur performa proses ETL (*Extract, Transform, Load*) pada *data warehouse* prestasi mahasiswa.
4. Melakukan implementasi OLAP (*Online Analytical Processing*) untuk menampilkan informasi hasil rekapitulasi data prestasi dan akademik mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya sesuai dengan standar akreditasi.

#### 1.4 Manfaat

Manfaat yang dapat diperoleh dari penelitian ini adalah:

1. Menyediakan data rekapitulasi dari sumber data akademik mahasiswa dan kemahasiswaan.
2. Menghasilkan informasi yang sesuai dengan kebutuhan penilaian akreditasi pada parameter prestasi mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya.

#### 1.5 Batasan Masalah

Agar masalah yang dirumuskan tidak keluar melebihi konteks, maka penelitian ini dibatasi dalam hal:

1. Data yang digunakan dalam penelitian ini adalah data prestasi dan akademik mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya selama empat tahun terakhir.
2. Penelitian dibatasi dan difokuskan pada Bagian Kemahasiswaan dan Bagian Akademik Fakultas Ilmu Komputer.
3. *Grain* yang dipresentasikan pada tabel fakta adalah jumlah prestasi, jumlah mahasiswa dan jumlah lulusan.
4. Data mahasiswa dibatasi hanya data angkatan mahasiswa.

#### 1.6 Sistematika Pembahasan

Untuk memudahkan pemahaman pembaca dalam mengikuti alur pembahasan yang terdapat pada penelitian ini, maka sistematika pembahasan dibagi menjadi lima bab sesuai dengan permasalahan yang dibahas, yaitu:

##### BAB 1 PENDAHULUAN

Bab ini berisi latar belakang penelitian, rumusan masalah, tujuan penelitian, dan manfaat penelitian.

##### BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas mengenai ringkasan dan sintesis dari penelitian sejenis yang pernah dilakukan dan landasan teori yang mendukung penelitian dalam melakukan analisis dan perancangan *data warehouse*.

##### BAB 3 METODE PENELITIAN





Pada bab ini akan dijabarkan mengenai metodologi yang digunakan dalam penelitian ini.

#### BAB 4 HASIL DAN PEMBAHASAN

Pada bab ini dijabarkan mengenai hasil serta pembahasan dari penelitian yang dilakukan, yang meliputi hasil analisis data, arsitektur *data warehouse*, rancangan worksheet, data dan *data warehouse* prestasi mahasiswa, rancangan proses ETL (*Extract, Transform, Load*), implementasi *data warehouse*, implementasi proses ETL (*Extract, Transform, Load*) dan OLAP (*Online Analytical Processing*), serta skenario dan hasil pengujian pada *data warehouse* yang telah diimplementasikan.

#### BAB 5 PENUTUP

Bab ini berisi kesimpulan yang didapatkan dari hasil perancangan, implementasi dan pengujian *data warehouse* prestasi mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya serta saran yang dapat membantu untuk pengembangan selanjutnya.



## BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi ringkasan dan sintesis dari penelitian sejenis yang telah dilakukan sebelumnya serta uraian dan pembahasan mengenai teori tentang data, informasi, basis data, *data warehouse* dan *Online Analytical Processing* (OLAP).

### 2.1 Kajian Pustaka

Penelitian sejenis yang pernah dilakukan oleh peneliti lain digunakan sebagai bahan referensi dan diharapkan dapat membantu dalam penelitian ini.

Penelitian pertama yang dilakukan oleh Ardista, Purbandini dan Taufik pada tahun 2017 dengan judul Rancang Bangun *Data Warehouse* Untuk Pembuatan Laporan dan Analisis pada Data Kunjungan Pasien Rawat Jalan Rumah Sakit Universitas Airlangga Berbasis *Online Analytical Processing* (OLAP). Penelitian ini bertujuan untuk merancang dan membangun *data warehouse* berbasis OLAP agar dapat digunakan oleh bagian rekam medis RSUD dalam pembuatan laporan. Dari penelitian yang dilakukan, didapat 10 tabel dimensi dan 2 tabel fakta. Data yang digunakan untuk *data warehouse* adalah data bulan Juli 2014 hingga Juni 2015 dengan jumlah transaksi sebanyak 41.243 sehingga diperlukan pembaharuan pada data untuk menghasilkan informasi terbaru.

Penelitian kedua dilakukan oleh Prasetyo, Soedijono dan Amborowati pada 2017 memiliki judul Perancangan *Data Warehouse* untuk Mendukung Perencanaan Pemasaran Perguruan Tinggi. Penelitian ini dilakukan untuk membangun *data warehouse* pemasaran di STMIK Amikom Purwokerto untuk penelusuran informasi yang berkaitan dengan penerimaan mahasiswa baru. Metode yang digunakan adalah *applied research*. Perangkat yang digunakan adalah Pentaho Data Integration/Kettle sebagai perangkat ETL (*Extract, Transform, Load*) dan Pentaho Workbench sebagai pengolah basis data.

Penelitian ketiga dengan judul Perancangan Model *Data Warehouse* Dalam Mendukung Perusahaan Jasa Pengiriman dilakukan oleh Oktavia pada tahun 2011. Objek penelitian ini adalah PT. Atlas Transindo Raya sebagai perusahaan yang bergerak dalam bidang jasa pengiriman. Aplikasi *data warehouse* yang dihasilkan selain berfungsi sebagai alat penunjang pelaporan juga berfungsi sebagai alat untuk menganalisis kecenderungan yang saat ini berlangsung dengan mengimplementasikan penggunaan *dashboard*.

Penelitian keempat dengan judul Pengembangan Sistem Analisis Akademis Menggunakan OLAP dan *Data Clustering* Studi Kasus : Akademik Universitas Sebelas Maret Surakarta ditulis oleh Yusuf Bakhtiar, Bima Murti Wijaya dan Dwi Cahyono pada tahun 2015. Penelitian ini berhasil mengembangkan sistem OLAP dan *clustering* untuk memenuhi fungsionalitas sistem yaitu sistem dapat melakukan *drill up* dan *drill down* untuk membantu merekap data jumlah mahasiswa, jumlah calon mahasiswa baru, jumlah lulusan, mencari nilai maksimum, minimum, rata-rata dan presentase profil IPK lulusan, dan lain



sebagainya. Data yang dikumpulkan untuk penelitian terdiri atas data primer berupa data yang diperoleh melalui wawancara dengan civitas akademik UNS dan data sekunder berupa data yang diambil melalui studi literatur dan telaah dokumen. Proses ETL pada *data warehouse* perlu dikembangkan agar data dapat diperbaharui secara otomatis jika terdapat perubahan pada data sumber.

Penelitian kelima dengan judul Pembuatan *Data Warehouse* Penjualan Produk dan Penerapan Dalam Studi Kasus Divisi Greenscope Energy - PT. Tustika Nagata Surabaya ditulis oleh F. Azizah, Agus Hendrawan dan Aulia Vinarti pada tahun 2012. Metode *association rule mining* digunakan untuk melakukan analisa keterkaitan antar produk yang dibeli oleh pelanggan. Pada penelitian ini data reguler transaksi pembelian dan pengklasifikasian pelanggan masih belum ada.

## 2.2 Landasan Teori

### 2.2.1 Fakultas Ilmu Komputer Universitas Brawijaya

Fakultas Ilmu Komputer merupakan salah satu fakultas baru di Universitas Brawijaya. Fakultas ini sudah berdiri sejak tahun 2011, dimana sebelumnya fakultas ini bernama Program Teknologi Informasi dan Ilmu Komputer (PTIIK) yang merupakan gabungan dari dua program studi yang telah berada di Universitas Brawijaya. Saat ini Fakultas Ilmu Komputer Universitas Brawijaya memiliki dua jurusan dan enam program studi yang dinaunginya. Jurusan-jurusan tersebut, yaitu Teknik Informatika dan Sistem Informasi dengan enam program studinya, yaitu Magister Ilmu Komputer, Teknik Informatika, Teknik Komputer, Sistem Informasi, Pendidikan Teknologi Informasi dan Teknologi Informasi.

Sebagaimana fakultas pada umumnya, proses akreditasi program studi pada Fakultas Ilmu Komputer Universitas Brawijaya diselenggarakan setiap lima tahun sekali. Proses akreditasi tentunya memerlukan data yang digunakan pada proses penilaian. Beberapa parameter penilaian berkaitan dengan mahasiswa, seperti IPK lulusan, prestasi mahasiswa, publikasi mahasiswa, dan lain sebagainya. Dalam melakukan penilaian tersebut, data yang digunakan bersumber dari sub bagian akademik dan kemahasiswaan. Data yang diambil dari sumber data yang berbeda tersebut dinilai kurang efektif karena rekapitulasi data harus dilakukan secara berulang. Untuk menangani masalah tersebut, *data warehouse* dapat digunakan untuk menampung dan merekapitulasi data akademik dan kemahasiswaan.

### 2.2.2 Online Analytical Processing (OLAP)

*Online Analytical Processing* (OLAP) adalah teknologi yang digunakan untuk mengakses informasi yang terdapat pada *data warehouse* secara efektif untuk proses analisis secara *online* (Aho, 2000). Penggabungan dan pengelompokan data sering diperlukan untuk melakukan analisis pada data. Dengan menggunakan teknologi OLAP, data dapat dihitung dan diintegrasikan terlebih dahulu sehingga mempercepat proses analisis.



### 2.2.3 Data

Pada penelitian yang dilakukan oleh Zins (2007) dengan judul *Conceptual Approaches for Defining Data, Information, and Knowledge* terdapat beberapa definisi mengenai data, antara lain:

Data merujuk pada catatan yang dikodekan untuk digunakan di komputer, namun lebih banyak digunakan untuk pengamatan statistik dan catatan lain atau kumpulan bukti. Data adalah seperangkat simbol yang mewakili fakta mentah (Raya Fidel). Data merupakan fakta dan statistik yang dapat dinilai, dihitung, diukur dan disimpan (Donald Hawkins).

### 2.2.4 Informasi

Menurut Humdiana dan Indrayani (2008), informasi adalah data yang berguna yang diolah sehingga dapat dijadikan dasar dalam proses pengambilan keputusan secara tepat. Informasi dapat diandalkan jika bebas dari kesalahan. Informasi bersifat relevan jika dapat mendukung proses pengambilan keputusan. Informasi dikatakan lengkap jika mengandung semua data yang relevan. Informasi dapat dipahami jika tersusun dalam bentuk yang dapat dimanfaatkan.

Konsep informasi dapat dikategorikan sebagai berikut (McCreadie & Rice, 1999):

1. Informasi sebagai representasi pengetahuan diartikan sebagai pengetahuan yang disimpan.
2. Informasi sebagai data di lingkungan. Informasi dapat diperoleh dari fenomena di lingkungan ketika diinterpretasikan dengan tepat.
3. Informasi sebagai bagian dari proses komunikasi. Faktor sosial dan waktu berperan penting dalam pemrosesan dan interpretasi sebuah informasi.
4. Informasi sebagai sumber daya atau komoditas. Informasi ditransmisikan dalam pesan dari pengirim ke penerima sehingga akan ada nilai tambah ketika informasi disebarluaskan.

### 2.2.5 Database (Basis Data)

Basis data merupakan sekumpulan data tersebar yang berhubungan secara logis, dan penjelasan dari data ini dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi (Connolly & Begg, 2015). Basis data juga dapat dikatakan sebagai kumpulan seluruh sumber data berbasis komputer milik organisasi (McLeod dan Scshell, 2007).

Dari pengertian tersebut dapat disimpulkan bahwa basis data merupakan sekumpulan data yang tersebar yang berhubungan secara logis dan dikelola secara terpusat sehingga dapat diakses oleh banyak pengguna dan sistem.

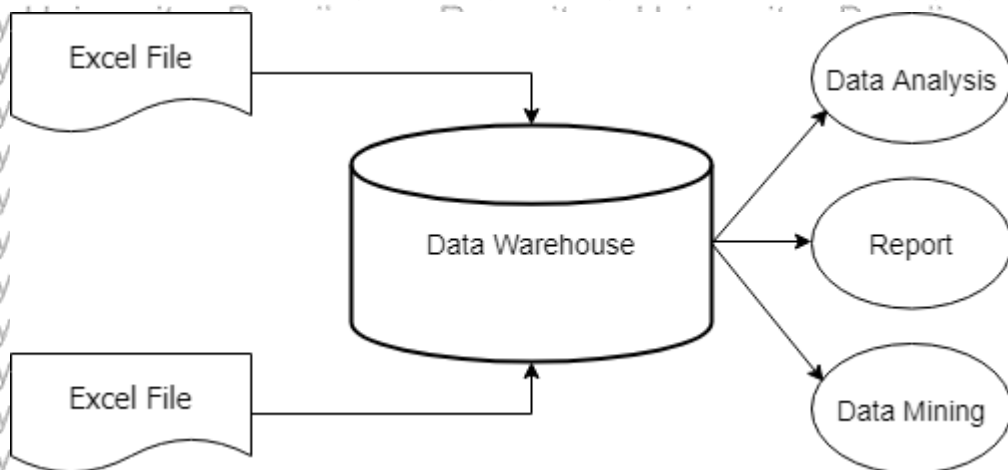


## 2.2.6 Database Management System (DBMS)

Database Management System (DBMS) adalah sebuah sistem perangkat lunak yang memungkinkan pengguna untuk membuat, memelihara serta melakukan kontrol akses ke basis data (Connolly & Begg, 2015).

## 2.2.7 Data Warehouse

Menurut Connolly dan Begg (2015), *data warehouse* diartikan sebagai kumpulan data yang berorientasi subyek, terintegrasi, dan bersifat *time variant* serta *nonvolatile* yang digunakan untuk mendukung proses pengambilan keputusan manajemen. Adapun topologi *data warehouse* dapat dilihat pada Gambar 2.1. Hal pertama yang perlu dilakukan adalah menentukan isi *data warehouse* yang bersumber dari berbagai sumber data eksternal, seperti *file Excel*, *flat file*, basis data transaksional, dan lain sebagainya. Setelah itu, struktur *data warehouse* ditetapkan untuk kemudian dianalisis dan dibangun. Setelah *data warehouse* dibangun, *front-end* aplikasi dapat dibuat untuk melakukan analisis data mendalam dari informasi yang disimpan pada *data warehouse*.



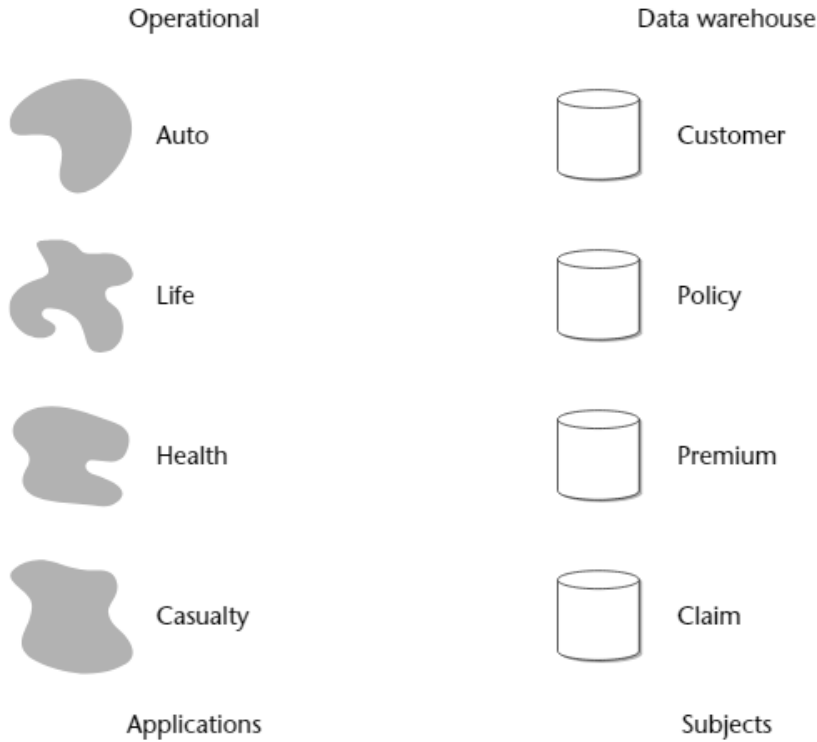
Gambar 2.1 Topologi *data warehouse*

Terdapat empat karakteristik *data warehouse* menurut Inmon (2005), yaitu :

### 1. *Subject-oriented*

*Data warehouse* berorientasi pada subyek suatu organisasi. Berbeda dengan *Online Transactional Processing (OLTP)* yang berfokus pada proses atau fungsi. Contohnya, subyek utama pada perusahaan asuransi antara lain, pelanggan, klaim, kebijakan yang dapat dilihat pada Gambar 2.2.

**SUBJECT ORIENTATION**



**Gambar 2.2 Aspek orientasi subyek dari data warehouse**

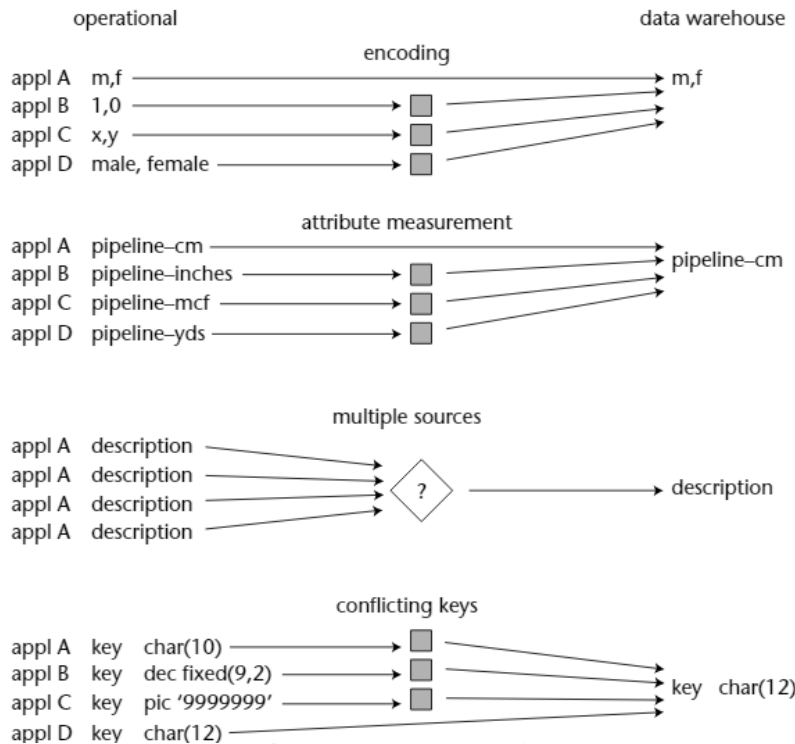
Sumber: Inmon (2005)

**2. Integration**

Data dari banyak sumber yang berbeda dan terpisah dipindahkan untuk kemudian dimuat ke dalam *data warehouse*. Selama proses pemindahan tersebut, data dikonversi, diformat, dirangkum dan lain sebagainya. Integrasi terjadi saat data yang berasal dari lingkungan operasional dimuat ke dalam *data warehouse*. Untuk menciptakan subyek area yang berguna, sumber data yang berasal dari beberapa sistem yang berbeda harus terintegrasi sebelum dimuat ke dalam *data warehouse*. Proses integrasi data dapat dilihat pada Gambar 2.3.



**INTEGRATION**



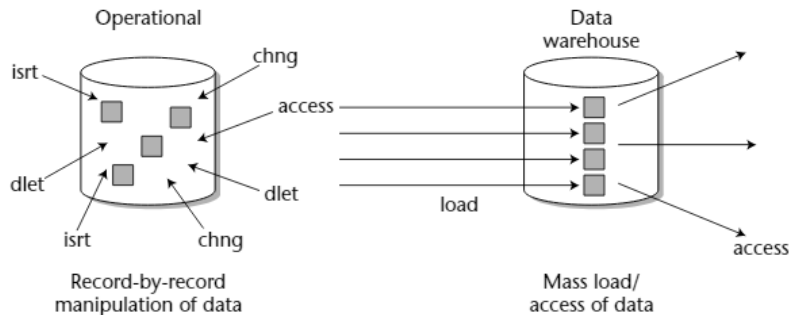
**Gambar 2.3 Aspek integrasi data pada data warehouse**

Sumber: Inmon (2005)

**3. Nonvolatile**

Data pada *data warehouse* hanya dapat dimuat dan diakses, tidak dapat diperbarui. Berbeda dengan basis data operasional (OLTP) dimana proses *insert, update, delete* dapat dilakukan dengan basis *record*. Pada *data warehouse*, data dimuat dalam keadaan statis sehingga ketika perubahan berikutnya terjadi, maka data baru akan disatukan dengan data yang sebelumnya dan catatan historis perubahan data disimpan di *data warehouse* seperti yang dapat dilihat pada Gambar 2.4.

**NONVOLATILITY**



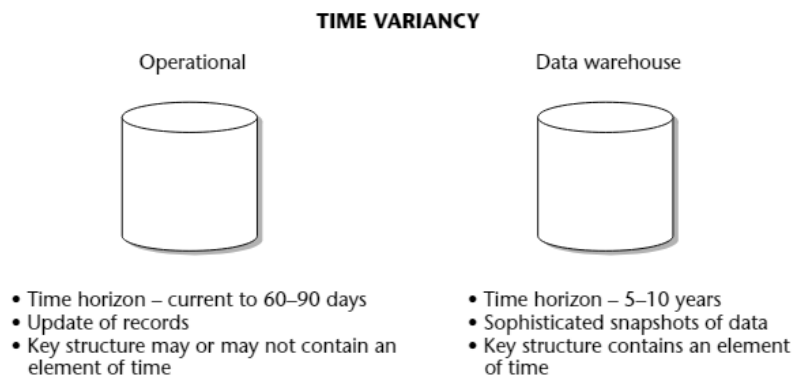
**Gambar 2.4 Aspek nonvolatile pada data warehouse**

Sumber: Inmon (2005)



#### 4. Time Variant

Setiap unit data di *data warehouse* bersifat akurat pada waktu tertentu. Pada beberapa kasus, catatan diberi *record* dalam format *time stamp*. Pada kasus lain, catatan memiliki tanggal transaksi. Namun dalam setiap kasus terdapat beberapa bentuk penandaan waktu untuk menunjukkan momen dalam waktu di mana catatan itu akurat. Data yang terdapat dalam *data warehouse* bersifat akurat selama periode tertentu, hal ini yang membedakan *data warehouse* dengan OLTP, dimana data yang tersimpan pada OLTP bersifat akurat hanya pada waktu sesaat setelah data diakses. Horizon waktu yang dimiliki oleh lingkungan yang berbeda bersifat berbeda pula. Horizon waktu dapat diartikan sebagai parameter dari waktu yang direpresentasikan dalam suatu lingkungan. Pada OLTP, horizon waktu yang normal berkisar antara 60 - 90 hari, sedangkan untuk *data warehouse* adalah sekitar 5 - 10 tahun. Oleh karena perbedaan dalam horizon waktu inilah, *data warehouse* mengandung histori lebih banyak dibandingkan dengan basis data operasional. Ilustrasi mengenai aspek *time variant* pada *data warehouse* dapat dilihat pada Gambar 2.5.



**Gambar 2.5 Aspek *time variant* pada *data warehouse***

Sumber: Inmon (2005)

#### 2.2.7.1 Information Package

*Information package* merupakan suatu konsep pengumpulan kebutuhan informasi pada *data warehouse*. *Information package* mendefinisikan hubungan antara subyek dengan *measure*. *Information package* dibuat dengan tujuan untuk menyediakan ruang lingkup yang terfokus pada kebutuhan pengguna sehingga komunikasi yang efektif antara staf teknis dan pengguna dapat terfasilitasi (Hammergren, 1989).

#### 2.2.7.2 Dimensional Modeling

*Dimensional modeling* adalah sebuah teknik desain logis untuk menghadirkan data dalam bentuk yang standar dan intuitif yang memungkinkan pengaksesan basis data dengan performa yang tinggi (Connolly & Begg, 2015).

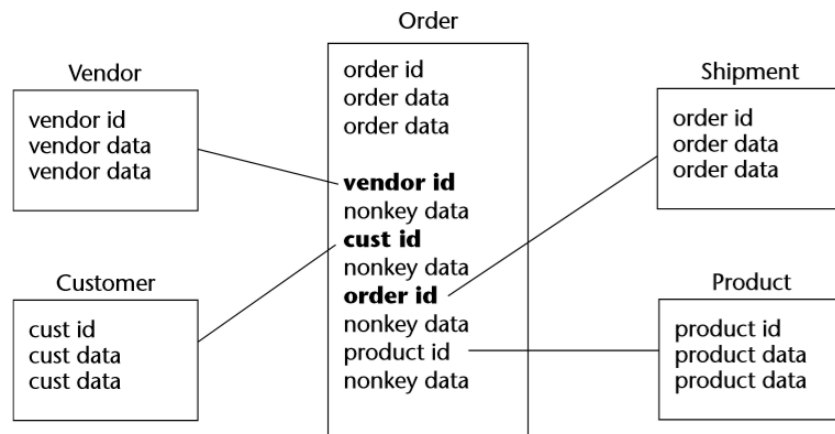
Terdapat tiga konsep *dimensional modeling* yang dikenal saat ini, yaitu:





1. *Star Schema*

*Star schema* merupakan sebuah struktur logis yang memiliki sebuah tabel fakta berisi data terbaru di tengah, yang dikelilingi tabel dimensi yang berisi data keterangan atau referensi (Connolly & Begg, 2015). Tabel fakta mengandung data kuantitatif, sedangkan tabel dimensi berisi data deskriptif mengenai subyek bisnis. Setiap tabel dimensi memiliki *primary key* yang berhubungan dengan tabel fakta sebagai *foreign key*. *Star schema* digunakan karena beberapa alasan, seperti mudah dimengerti, tidak kompleks, paling banyak digunakan untuk membangun sebuah *data warehouse* dan skema yang paling efektif untuk penanganan yang lebih sederhana (Taleb & Saedi, 2017). Ilustrasi mengenai *star schema* ditunjukkan pada Gambar 2.6.



Gambar 2.6 *Star Schema*

Sumber: Inmon (2005)

2. *Snowflake Schema*

*Snowflake schema* adalah variasi dari *star schema* dimana tabel dimensi dilakukan normalisasi (Ponniah, 2001).

3. *Fact Constellation Schema*

*Fact constellation schema* merupakan skema multi dimensional yang berisi lebih dari satu tabel fakta yang saling berbagi tabel dimensi. Jenis skema ini dapat dilihat sebagai gabungan dari berbagai *star schema* sehingga sering juga disebut sebagai skema galaksi (Han & Kamber, 2006).

Pada *dimensional modeling* terdapat lima komponen yang digunakan, yaitu:

1. *Fact* (fakta)

Fakta merupakan atribut numerik yang sangat spesifik yang biasanya digunakan sebagai sebuah ukuran dari performansi bisnis (Kimball & Ross, 2002).

2. *Fact table* (tabel fakta)

Tabel fakta merupakan tabel utama pada model dimensional. Tabel fakta berisi pengukuran dari proses bisnis serta berisi *foreign key* yang



berelasi dengan tabel dimensi (Taleb & Saedi, 2017). Tabel fakta merupakan struktur yang berisi data yang sering berubah. Di sekeliling tabel fakta terdapat tabel dimensi yang mendeskripsikan satu aspek penting pada tabel fakta (Inmon, 2005).

3. *Dimension* (dimensi)

Dimensi adalah sebuah entitas independen yang berfungsi sebagai titik masuk atau sebagai mekanisme untuk melakukan proses *slicing* dan *dicing* terhadap pengukuran tambahan pada tabel fakta pada model dimensional (Kimball & Ross, 2002). Setiap dimensi berisi informasi deskriptif dan memiliki *primary key* yang berfungsi sebagai dasar untuk berintegrasi dengan tabel fakta (Taleb & Saedi, 2017).

4. *Dimension table* (tabel dimensi)

Tabel dimensi adalah sebuah tabel dalam model dimensional yang memiliki *primary key* tunggal dan kolom dengan atribut deskriptif (Kimball & Ross, 2002). Sebagian besar tabel dimensi memiliki banyak atribut berisi data yang dikumpulkan dari berbagai sumber (Taleb & Saedi, 2017).

5. *Surrogate key*

Menurut Kimball dan Ross (2002) (Kimball & Ross, 2002), *surrogate key* merupakan kunci *integer* yang secara berurutan ditetapkan sesuai kebutuhan di *staging area* untuk mengisi tabel dimensi dan masuk ke dalam tabel fakta. Pada tabel dimensi, *surrogate key* merupakan *primary key*. Sedangkan pada tabel fakta, *surrogate key* adalah *foreign key* bagi dimensi tertentu dan dapat menjadi bagian dari *primary key* pada tabel fakta. *Surrogate key* dibutuhkan untuk menangani dimensi yang berubah secara perlahan serta menangani data yang hilang atau data yang tidak dapat digunakan.

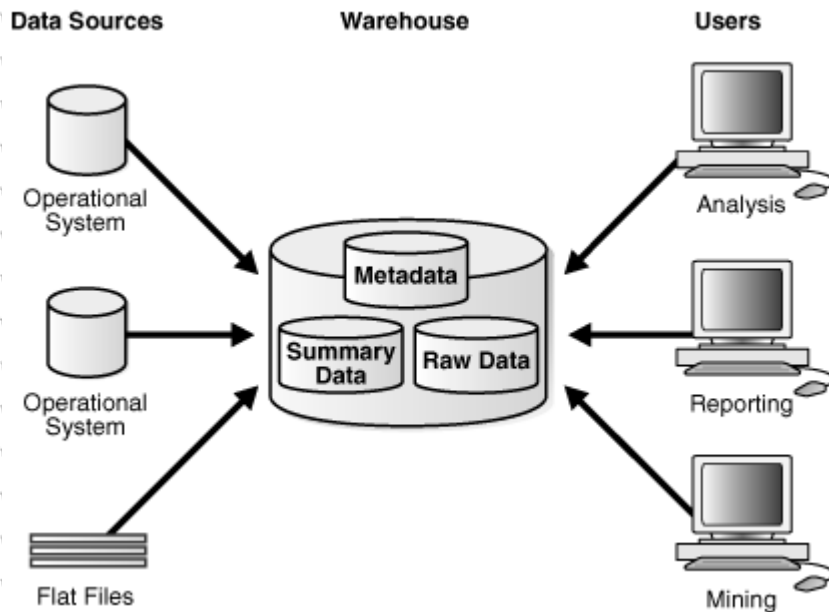
### 2.2.7.3 Arsitektur *Data Warehouse*

Arsitektur *data warehouse* yang digunakan pada suatu organisasi bergantung pada situasi tertentu suatu organisasi. Terdapat tiga arsitektur umum *data warehouse* yaitu arsitektur dasar, arsitektur *data warehouse* dengan *staging area*, serta arsitektur *data warehouse* dengan *staging area* dan *data mart* (Oracle, 2017). Pada arsitektur dasar *data warehouse*, pengguna akhir secara langsung mengakses data dari sumber eksternal melalui *data warehouse*. *Data warehouse* memuat *summary data*, *metadata* dan *raw data* yang dapat diakses oleh pengguna. Adapun sumber data yang digunakan dapat berupa sistem operasional maupun *flat file*. Dan pengguna dari *data warehouse* dapat melakukan proses analisis data, *data mining* dan juga *reporting* dari data yang tersimpan dalam *data warehouse*. Arsitektur jenis ini disebut juga sebagai arsitektur EDW (*Enterprise Data Warehouse*) karena *data warehouse* berisi data agregat yang menjangkau seluruh organisasi. Ilustrasi arsitektur dasar *data warehouse* dapat dilihat pada Gambar 2.7.

Jika pengguna ingin menyesuaikan arsitektur *data warehouse* untuk grup yang berbeda pada suatu organisasi, maka arsitektur *data warehouse* dengan *staging*

area dan *data mart* dapat digunakan. Arsitektur jenis ini dirancang untuk lini bisnis tertentu dari suatu organisasi.

Sedangkan pada arsitektur *data warehouse* dengan *staging area*, data dari sumber eksternal dibersihkan dan ditransformasi sebelum dimuat ke dalam *data warehouse*. Menurut Inmon (2005), *staging area* merupakan lokasi dimana data dari sumber data disimpan secara sementara sebelum diproses dan dimuat ke dalam *data warehouse*.



Gambar 2.7. Arsitektur dasar *data warehouse*

Sumber: Oracle (2017)

#### 2.2.7.4 Metodologi Perancangan *Data Warehouse*

Metode perancangan *data warehouse* yang diperkenalkan oleh Kimball meliputi sembilan tahap yang dikenal sebagai *Nine Step Design Methodology* (Kimball & Ross, 2002) yang prosesnya meliputi:

1. *Choosing the process* (pemilihan proses)

Pemilihan proses dari subyek yang dibutuhkan pada *data mart* yang akan dibangun harus dapat menjawab permasalahan bisnis yang penting. Proses ini dilakukan untuk memberi batasan mengenai *data warehouse* yang akan dibangun.

2. *Choosing the grain* (pemilihan grain)

Proses pemilihan *grain* diartikan sebagai proses untuk menentukan dengan tepat apa yang direpresentasikan oleh *record* pada tabel fakta. Tingkat detail (*granularity*) data akan ditentukan pada proses ini.

3. *Identifying and conforming the dimension* (identifikasi dan penyesuaian dimensi)



Pada proses ini dimensi dan *grain* yang telah ditentukan akan disesuaikan dan ditampilkan dalam bentuk matriks.

#### 4. *Choosing the Facts* (Pemilihan Fakta)

Pada proses ini *measurement* yang diperlukan akan diidentifikasi. Selain itu, fakta berupa angka dan harus memperkuat *grain* yang telah ditentukan. Pada model multidimensional, tabel fakta berisi *measurement* dari proses bisnis dan *foreign key* dari tabel dimensi.

#### 5. *Storing pre-calculation in the fact table* (penyimpanan pre-kalkulasi di tabel fakta)

Setelah tabel fakta dipilih maka tabel fakta diperiksa ulang untuk memeriksa apakah ada fakta yang dapat diterapkan pre-kalkulasi untuk kemudian disimpan pada tabel fakta.

#### 6. *Rounding out the dimension table* (melengkapi tabel dimensi)

Tabel dimensi dilengkapi dengan menambahkan deskripsi teks dan menentukan hierarki untuk mempermudah proses analisis. Deskripsi teks yang ditambahkan harus mudah digunakan dan dimengerti oleh pengguna.

#### 7. *Choosing the duration of the database* (pemilihan durasi database)

Pemilihan data historis dilakukan sesuai kebutuhan informasi. Semakin banyak data yang dimuat ke dalam *data warehouse* maka semakin lengkap informasi yang dapat dihasilkan selama data yang dimuat bukan data yang tidak bermanfaat. Oleh karena itu, isi dan format data yang akan dimuat harus diperhatikan sebelum memuat data ke dalam *data warehouse*.

#### 8. *Tracking slowly changing dimension* (pelacakan perubahan dimensi secara perlahan)

Dimensi dapat berubah secara perlahan seiring adanya kebutuhan dan waktu. Perubahan tersebut dapat berupa perubahan atribut dimensi yang di-*overwrite*, atau perubahan yang menyebabkan pembentukan *record* dimensi baru, serta perubahan yang menyebabkan pembentukan kolom baru.

#### 9. *Deciding the query priorities and the query modes* (penentuan prioritas *query* dan mode *query*)

Perancangan fisikal dipertimbangkan pada proses ini, seperti keberadaan *aggregate* dan *summary*. Selain itu, proses ETL (*Extract, Transform, Load*) serta masalah *backup* juga harus diperhatikan.

### 2.2.7.5 *Data Mart*

*Data mart* adalah struktur data per departemen yang berasal dari *data warehouse* dimana data dilakukan denormalisasi berdasarkan kebutuhan informasi setiap departemen (Inmon, 2005). Dalam bentuknya yang paling



sederhana, *data mart* mewakili data dari suatu proses bisnis (Kimball & Ross, 2002).

#### 2.2.7.6 Cube

*Cube* merupakan sebuah bentuk basis data dengan bentuk data berupa *cell* dimana posisi dari *cell* tersebut ditentukan oleh variabel yang disebut sebagai dimensi. Istilah *cube* digunakan untuk mempermudah visualisasi mengenai sifat multidimensional (Kusnawi, 2010).

#### 2.2.8 Extract, Transform, Load (ETL)

Proses ETL adalah sebuah set proses dimana sumber data operasional disiapkan untuk *data warehouse*. Proses ini terdiri atas penggalan data operasional dari sumber data, lalu mentransformasikannya, memuat dan mengindeksnya kemudian memastikan kualitasnya dan menerbitkannya (Kimball & Ross, 2002).

##### 2.2.8.1 Proses ETL

###### 1. Extraction

*Extraction* merupakan langkah pertama yang dilakukan dalam memuat data ke dalam lingkungan *data warehouse*. Mengekstraksi berarti membaca sumber data dan menyalin data yang diperlukan ke dalam *staging area* untuk proses manipulasi data selanjutnya. Setelah data diekstraksi ke *staging area*, terdapat banyak transformasi potensial, seperti proses *cleansing* data (memilah data ke dalam format standar, mengoreksi kesalahan ejaan, menyelesaikan konflik domain, atau mengurus elemen data yang hilang), menggabungkan data dari berbagai sumber, melakukan deduplikasi data dan menentukan *warehouse key*.

###### 2. Transformation

Pada proses transformasi, format data dari sumber data operasional diubah menjadi format data pada *data warehouse* yang lebih spesifik. Format data harus diubah sebelum dimuat ke dalam *data warehouse* untuk menangani masalah integritas data yang disebabkan oleh faktor berikut, antara lain:

- a. Ejaan kata yang berbeda untuk maksud yang sama, contohnya ejaan nama Jon dan John.
- b. Penggunaan nama yang berbeda, contohnya Cleveland dan Cleaveland.
- c. Kekosongan beberapa *file* data yang diperlukan.
- d. Penyebutan nama perusahaan dengan cara yang berbeda, seperti Google Inc dan Google.

###### 3. Loading

Memuat data ke dalam *data warehouse* merupakan tahap terakhir pada proses ETL. Tahap ini dapat dilakukan dengan dua cara, yaitu:

- a. *Refresh*



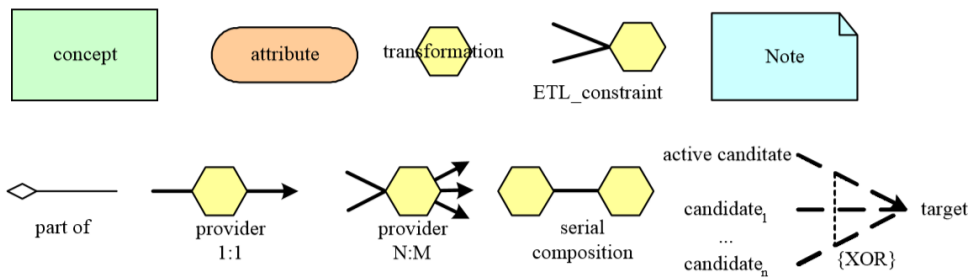
Data lama pada *data warehouse* akan digantikan dengan data baru. Proses *refresh* biasanya terjadi saat pertama kali melakukan *loading data*.

b. *Update*

*Update* berjalan tanpa mengubah atau menghapus data lama pada *data warehouse*. *Update* dilakukan secara reguler dimana jika terjadi perubahan pada data maka data baru tersebut akan digabungkan dengan data lama yang sudah ada pada *data warehouse*.

2.2.8.2 Konseptual ETL

Pemodelan konseptual ETL dilakukan untuk menspesifikasi entitas yang berorientasi pada pengguna yang digunakan untuk menangkap makna dari suatu proses ETL (Vassiliadis, Simitsis, & Skiadopoulos, 2002). Pada pemodelan konseptual ETL, terdapat beberapa notasi grafis yang digunakan yang ditunjukkan pada Gambar 2.8. Pada notasi UML (*Unified Modeling Language*) tersebut, atribut tidak ditanamkan pada entitas yang melingkupinya, seperti tabel relasional atau kelas UML.



Gambar 2.8 Notasi pada pemodelan konseptual ETL

Sumber: Vassiliadis, Simitsis, & Skiadopoulos (2002)

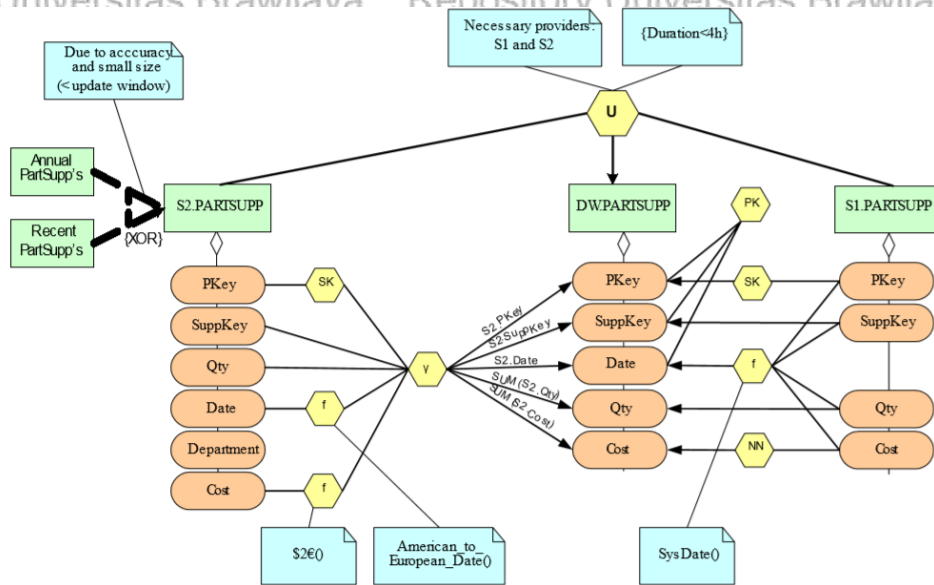
Pada notasi UML konseptual ETL tersebut, atribut berperan sama dengan model pada *entity relationship* dimana notasinya digambarkan dengan bentuk oval. Sedangkan *concept* mewakili suatu entitas dalam sumber basis data atau dalam *data warehouse*. Suatu *concept* memiliki atribut dan nama, sama seperti model *entity relationship*. Setiap entitas yang terdiri atas beberapa atribut merupakan *instance* dari kelas *concept*.

Transformasi adalah abstraksi yang menjalankan suatu fungsi atau tugas. Terdapat dua kategori pada transformasi, yaitu penyaringan dan pembersihan data, seperti pemeriksaan *foreign key*, serta operasi transformasi, seperti proses agregasi. Transformasi digambarkan dengan segi enam yang ditandai dengan simbol yang sesuai.

Pada contoh yang ditunjukkan pada Gambar 2.9, terdapat beberapa transformasi, salah satu contohnya adalah pemetaan S1.PARTSUPP ke DW.PARTSUPP. Pada transformasi tersebut terjadi penetapan *surrogate key*, perhitungan tanggal sistem (f) dan pemeriksaan *Not Null* pada atribut biaya.

ETL Constraint diterapkan sebagai batasan mulai dari atribut yang terlibat dan penargetan transformasi fasilitator. Sebagai contoh, batasan *primary key* dan nilai *null* ingin ditetapkan pada beberapa atribut. Sedangkan *note* digunakan sebagai *tag* informal untuk menyimpan komentar tambahan yang dibuat selama masa perancangan.

Notasi hubungan *part-of* digunakan untuk menekankan bahwa *concept* terdiri atas beberapa atribut. Sedangkan hubungan *provider* digunakan untuk memetakan atribut *input* menuju sekumpulan atribut *output* melalui proses transformasi yang relevan.



Gambar 2.9 Contoh diagram pemodelan konseptual

Sumber: Vassiliadis, Simitsis, & Skiadopoulos (2002)

## 2.2.9 Talend Open Studio for Data Integration

Talend Open Studio for Data Integration merupakan sebuah *tool* untuk mengintegrasikan data dari berbagai sumber data untuk kemudian dimuat atau dituliskan pada *data warehouse*, *data mart* maupun aplikasi OLAP untuk keperluan pelaporan, *dashboarding* dan keperluan analisis lainnya. *Tool* ini menyediakan komponen yang diperlukan untuk melakukan proses ETL dan menjadi penghubung antara basis data, *file*, *web service* dan layanan lainnya (Bowen, 2012).

### 2.2.10 Jedox

Jedox merupakan sebuah *tool* untuk keperluan analisis data sistematis. Jedox memiliki *server* untuk pemrosesan analitis multidimensional dan dirancang khusus untuk keperluan pelaporan, perencanaan dan konsolidasi data. Jedox menggunakan *spreadsheet* sebagai *user interface* sehingga pengguna dapat lebih mudah dalam mengoperasikan *tool* tersebut (Jedox, 2019).



### 2.2.11 Pengujian

Pengujian dilakukan untuk menguji *data warehouse* yang telah dibangun. Pengujian yang dilakukan, yaitu pengujian validasi dan pengujian performa proses ETL (*Extract, Transform, Load*).

#### 2.2.11.1 Pengujian Validasi

Untuk memastikan bahwa skema pada *data warehouse* telah sesuai dengan kebutuhan pengguna, maka dilakukan pengujian validasi. Pengujian ini dilakukan dengan melakukan pencocokan model dimensional atau skema dengan kebutuhan pengguna (Umam, Wicaksono, & Purnomo, 2019). Hal tersebut dilakukan dengan mengukur validitas data hasil proses ETL pada *data warehouse*.

#### 2.2.11.2 Pengujian Performa Proses ETL (*Extract, Transform, Load*)

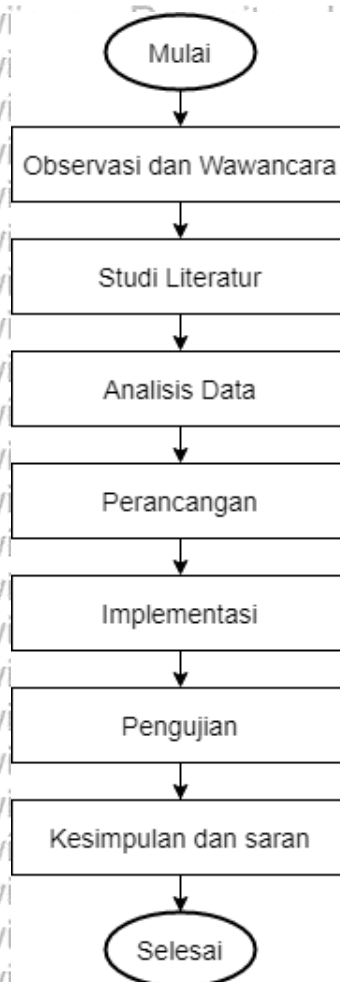
Pengujian performa dilakukan dengan tujuan untuk memverifikasi bahwa persyaratan teknis yang terkait dengan platform, kapasitas dan waktu terpenuhi. Faktor yang dapat mempengaruhi performa proses ETL, antara lain adalah kualitas data yang terdapat pada sumber data. Selain itu, hal yang perlu diperhatikan dalam proses ETL, yaitu waktu, jumlah data yang dimuat dan jumlah data yang terdapat dalam tabel target. Adapun performa proses ETL yang diuji pada penelitian ini dilihat dari segi lamanya waktu eksekusi dan baris data yang diproses dari sumber data (Firmansyah, 2017).





## BAB 3 METODOLOGI

Penelitian yang akan dilakukan bersifat implementatif dengan tujuan untuk menganalisis kebutuhan informasi serta merancang *data warehouse* untuk membantu proses rekapitulasi data prestasi dan akademik mahasiswa. Penelitian ini akan dilakukan di Fakultas Ilmu Komputer Universitas Brawijaya. Penelitian dilaksanakan mulai bulan September 2019 sampai dengan bulan Juli 2020 dengan pihak yang turut berpartisipasi adalah Bagian Akademik dan Bagian Kemahasiswaan dimana data akademik mahasiswa dan prestasi mahasiswa didapatkan. Pada bab ini akan dijelaskan mengenai metodologi yang akan ditempuh pada penelitian untuk mendukung tercapainya tujuan penelitian. Metodologi yang akan ditempuh, yaitu observasi, studi literatur, analisis data, perancangan, implementasi, pengujian dan pengambilan kesimpulan dan saran. Adapun tahapan metodologi penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Metodologi penelitian

### 3.1 Observasi dan Wawancara

Pada tahap observasi dan wawancara, penggalian masalah dilakukan dengan melakukan pengamatan di lapangan dan wawancara (*interview*) dengan Bagian



Kemahasiswaan Fakultas Ilmu Komputer. Dari observasi dan wawancara yang dilakukan nantinya masalah penelitian didapatkan untuk kemudian dijadikan sebagai latar belakang diadakannya penelitian. Adapun hasil wawancara dapat dilihat pada Lampiran A.

### 3.2 Studi Literatur

Pada studi literatur dibahas teori penelitian yang mendukung pengerjaan penelitian. Dasar teori yang digunakan didapatkan dari berbagai sumber, seperti jurnal, buku/ebook dan penelitian terdahulu. Teori yang dipelajari dalam penelitian ini, antara lain teori mengenai data, informasi, basis data, *Database Management System (DBMS)*, *data warehouse*, teknologi *OLAP* dan proses *ETL*.

Adapun penelitian sejenis yang telah dilakukan oleh peneliti lain yang digunakan sebagai referensi pendukung dalam penelitian ini, yaitu penelitian yang dilakukan oleh Ardista, Taufik dan Purbandini pada tahun 2016 dengan judul Rancang Bangun *Data Warehouse* untuk Pembuatan Laporan dan Analisis pada Data Kunjungan Pasien Rawat Jalan Rumah Sakit Universitas Airlangga Berbasis *Online Analytical Processing (OLAP)*. Penelitian selanjutnya adalah penelitian dengan judul Perancangan *Data Warehouse* untuk Mendukung Perencanaan Pemasaran Perguruan Tinggi oleh Prasetyo, Soedijono dan Amborowati pada tahun 2017. Penelitian ketiga adalah penelitian yang dilakukan oleh Oktavia pada tahun 2011 dengan judul Perancangan Model *Data Warehouse* dalam Mendukung Perusahaan Jasa Pengiriman. Penelitian keempat merupakan penelitian dengan judul Pengembangan Sistem Analisis Akademis Menggunakan *OLAP* dan *Data Clustering* Studi Kasus : Akademik Universitas Sebelas Maret Surakarta yang dilakukan oleh Bakhtiar, wijaya dan Cahyono pada 2015. Penelitian terakhir yang digunakan sebagai bahan pertimbangan dalam penelitian ini adalah penelitian yang dilakukan oleh Azizah, Hendrawan dan Vinarti pada tahun 2012 dengan judul Pembuatan *Data Warehouse* Penjualan Produk dan Penerapan Dalam Studi Kasus Divisi Greenscope Energy - PT. Tustika Nagata Surabaya.

### 3.3 Analisis Data

Data yang sebelumnya telah dikumpulkan kemudian dianalisis untuk menentukan kebutuhan informasi. Kebutuhan informasi yang didapatkan akan menjadi dasar pembuatan *information package* pada tahap perancangan *data warehouse*. *Information package* merupakan suatu konsep pengumpulan kebutuhan pada *data warehouse*.

### 3.4 Perancangan

#### 3.4.1 Arsitektur *Data Warehouse*

Sebelum tahap perancangan *data warehouse* dilakukan, arsitektur *data warehouse* yang digunakan dirancang terlebih dahulu. Pada pengembangan *data warehouse* prestasi mahasiswa, jenis arsitektur yang digunakan adalah arsitektur dasar *data warehouse*. Penggunaan arsitektur tersebut karena *data warehouse*



yang akan dikembangkan memuat data dari Bagian Kemahasiswaan dan Bagian Akademik sehingga data yang tersimpan di dalamnya saling terintegrasi. Pengguna dapat mengakses satu sumber data saja untuk memperoleh informasi mengenai prestasi dan akademik mahasiswa yang akan dilaporkan dalam bentuk *user report* yang sesuai dengan standar akreditasi.

### 3.4.2 Perancangan *Worksheet*

Perancangan *worksheet* dilakukan dengan tujuan untuk membuat standarisasi format data yang akan digunakan pada proses ETL (*Extract, Transform, Load*) nantinya. *Worksheet* yang dirancang nantinya akan digunakan sebagai format pengunggahan data prestasi dan data akademik mahasiswa.

### 3.4.3 Perancangan Data Penelitian

Tahap perancangan data dilakukan guna menghasilkan data berkualitas yang akan digunakan untuk proses analisis. Dalam tahap ini, data yang digunakan adalah data asli berupa *file* Excel dan beberapa data *dummy* yang dirancang dan dibentuk sesuai dengan kolom-kolom yang digunakan pada hasil rancangan *worksheet* yang telah dibuat sebelumnya dengan menggunakan *tool* Microsoft Excel.

### 3.4.4 Perancangan *Data Warehouse*

Tahap perancangan merupakan tahap awal sebelum membangun suatu *data warehouse*. Pada tahap ini, kebutuhan informasi digali dengan proses meliputi:

#### 1. *Choosing the process* (pemilihan proses)

Berdasarkan permasalahan yang telah didapatkan melalui hasil observasi dan wawancara, proses yang dipilih pada perancangan *data warehouse* prestasi mahasiswa, yaitu proses pengolahan data kemahasiswaan berupa data prestasi mahasiswa dan pengolahan data akademik mahasiswa Fakultas Ilmu Komputer.

#### 2. *Choosing the grain* (pemilihan grain)

Setelah proses bisnis ditentukan, tahap selanjutnya adalah menentukan *grain* dari *data warehouse* yang akan dibangun. Pemilihan *grain* dilakukan berdasarkan hasil wawancara yang telah dilakukan. Adapun *grain* yang ditentukan dalam perancangan *data warehouse* prestasi mahasiswa adalah prestasi, lulusan dan mahasiswa aktif.

#### 3. *Identifying and conforming the dimension* (identifikasi dan penyesuaian dimensi)

Pada tahap ini dimensi dan *grain* yang telah ditentukan akan disesuaikan dan ditampilkan dalam bentuk matriks *information package*. *Information package* dibuat sebagai acuan pada tahap perancangan *data warehouse* selanjutnya.

#### 4. *Choosing the Facts* (Pemilihan Fakta)



Pada tahap ini, dilakukan pemilihan fakta yang akan digunakan pada tabel fakta prestasi mahasiswa dan akademik mahasiswa. Pemilihan fakta dilakukan berdasarkan *grain* yang telah ditentukan pada tahap sebelumnya.

5. *Storing pre-calculation in the fact table* (penyimpanan pre-kalkulasi di tabel fakta)

Setelah tabel fakta dipilih maka tabel fakta diidentifikasi kembali untuk memastikan perlu tidaknya proses pre-kalkulasi atau agregasi dari suatu tabel fakta dilakukan.

6. *Rounding out the dimension table* (melengkapi tabel dimensi)

Pada tahap ini, gambaran teks terhadap dimensi ditambahkan. Hal ini bertujuan agar pengguna dapat lebih mudah dalam memahami tabel. Adapun hubungan antara tabel fakta dan dimensi digambarkan dengan menggunakan ERD (*Entity Relationship Diagram*).

7. *Choosing the duration of the database* (pemilihan durasi database)

Durasi data yang disimpan dalam *data warehouse* prestasi mahasiswa adalah data akademik dan prestasi mahasiswa selama empat tahun, yaitu mulai dari tahun 2016 sampai dengan 2019.

8. *Tracking slowly changing dimension* (pelacakan perubahan dimensi secara perlahan)

Perubahan dimensi pada *data warehouse* dapat terjadi dalam waktu yang cukup lama, sehingga pembaharuan perlu dilakukan untuk menjaga konsistensi data. Adapun perubahan dimensi yang dilakukan dikategorikan sebagai *Slowly Changing Dimension* tipe 1, dimana data lama yang mengalami perubahan akan ditimpa (*overwrite*) dengan data baru.

9. *Deciding the query priorities and the query modes* (penentuan prioritas *query* dan mode *query*)

Penentuan proses *backup* data dan proses ETL (*Extract, Transform, Load*) dilakukan. Pada penelitian ini, proses *backup* dilakukan setiap akhir periode akademik untuk fakta Mahasiswa Aktif, setiap akhir tahun untuk fakta Prestasi dan setiap bulan untuk fakta Lulusan. Adapun aktor yang melakukan proses *backup* ialah administrator basis data.

### 3.4.5 Perancangan Proses ETL (*Extract, Transform, Load*)

Pada tahap perancangan proses ETL, dilakukan perancangan mengenai pemetaan data mulai dari sumber data menuju *staging area* kemudian sampai data dimuat ke dalam *data warehouse*. Tahap ini dilakukan untuk merepresentasikan proses ETL.



### 3.5 Implementasi

#### 3.5.1 Implementasi *Data Warehouse*

Tahap implementasi dilakukan setelah tahap analisis dan perancangan dilakukan. Pada tahap implementasi *data warehouse*, rancangan *data warehouse* yang telah dibuat diimplementasikan dengan membuat *query Data Definition Language* (DDL) SQL untuk membuat semua obyek dalam basis data.

#### 3.5.2 Proses ETL (*Extract, Transform, Load*)

Setelah implementasi *data warehouse* dilakukan, tahap selanjutnya adalah melakukan proses ETL pada basis data yang sebelumnya telah dibuat pada tahap implementasi. Pada tahap ini, dilakukan pemetaan (*mapping*) data dari sumber data menuju basis data OLAP sehingga data yang akan dimuat dalam basis data tersebut sesuai dengan kebutuhan. Selain itu, penjadwalan (*scheduling*) eksekusi proses ETL juga diatur pada tahap ini.

#### 3.5.3 Implementasi OLAP (*Online Analytical Processing*)

Setelah proses ETL dijalankan, maka selanjutnya dilakukan implementasi OLAP yang berguna dalam menyajikan informasi hasil proses rekapitulasi data prestasi dan akademik mahasiswa dan informasi nilai akhir dari parameter prestasi mahasiswa dalam penilaian akreditasi. Tahap ini dilakukan dengan membuat dimensi dan *cube* pada *server* Jedox. Setelah *cube* dibuat, maka data pada tabel fakta ditulis ke dalam *cube* yang sesuai dengan fakta yang akan ditampilkan pada *user report*.

### 3.6 Pengujian

Tahap pengujian dilakukan setelah tahap implementasi. Pengujian pertama adalah pengujian validasi data akademik dan data prestasi mahasiswa Fakultas Ilmu Komputer dengan kebutuhan informasi. Pengujian tersebut dilakukan dengan mencocokkan skema *data warehouse* dengan kebutuhan pengguna. Pengujian kedua adalah pengujian performa proses ETL yang dilakukan dengan mengukur waktu yang dihabiskan untuk melakukan eksekusi proses ETL dan banyaknya baris data yang diproses.

#### 3.6.1 Pengujian Validasi

Pengujian validasi dilakukan untuk mengetahui apakah skema atau model multidimensional *data warehouse* yang telah dibuat telah sesuai dengan kebutuhan pengguna. Pada pengujian ini akan dilakukan pencocokan antara skema pada *data warehouse* prestasi mahasiswa dengan kebutuhan dari pengguna dengan mengukur validitas data hasil proses ETL.

#### 3.6.2 Pengujian Performa Proses ETL (*Extract, Transform, Load*)

Pengujian performa proses ETL pada *data warehouse* prestasi mahasiswa yang dibangun dilakukan dengan melihat waktu eksekusi yang dihabiskan dan jumlah



baris data yang diproses dari sumber data untuk kemudian dimuat ke dalam *data warehouse*.

### 3.7 Kesimpulan dan Saran

Kesimpulan dibuat setelah perancangan, implementasi dan pengujian terhadap *data warehouse* dilakukan sehingga dapat menjawab pertanyaan dari masalah yang telah ditetapkan. Sebelum melakukan perancangan *data warehouse*, tahap observasi dan wawancara serta analisis dilakukan terlebih dahulu untuk menghasilkan kebutuhan informasi yang nantinya akan menjadi acuan dalam merancang *data warehouse*. Penulisan saran berisi perbaikan dari kekurangan aplikasi *data warehouse*, sehingga dapat bermanfaat bagi penelitian selanjutnya.

## BAB 4 HASIL DAN PEMBAHASAN

### 4.1 Hasil Analisis Data

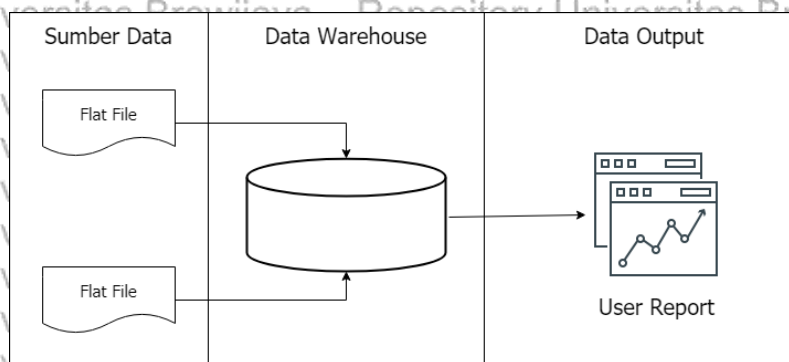
Data prestasi dan akademik mahasiswa yang telah dikumpulkan kemudian dianalisis untuk menentukan kebutuhan informasi sesuai dengan standar akreditasi. Dari hasil analisis data dan permasalahan yang dilakukan, didapatkan hasil berupa kebutuhan informasi yang diperlukan sebelum tahap perancangan *data warehouse* dilakukan. Adapun hasil analisis pada data dan permasalahan yang dilakukan, antara lain:

1. Laporan mengenai data prestasi mahasiswa yang meliputi jumlah prestasi setiap tahun, jumlah prestasi pada tingkat tertentu, jumlah prestasi berdasarkan program studi dan jumlah prestasi berdasarkan prestasi yang dicapai.
2. Laporan terhadap data lulusan yang meliputi jumlah lulusan setiap tahun akademik, program studi dan angkatan.
3. Laporan berisi data mahasiswa aktif yang meliputi jumlah mahasiswa aktif setiap program studi, tahun akademik dan angkatan.

### 4.2 Perancangan

#### 4.2.1 Arsitektur Data Warehouse

Dalam merancang *data warehouse* prestasi mahasiswa pada Fakultas Ilmu Komputer, arsitektur dasar *data warehouse* digunakan. Arsitektur jenis tersebut digunakan karena *data warehouse* memuat data dari dua sumber yang berbeda, yaitu Bagian Kemahasiswaan dan Bagian Akademik sehingga ketika informasi dari kedua sumber tersebut diperlukan, pengguna dapat mengakses satu sumber data saja. Pada arsitektur *data warehouse* yang digunakan terdapat beberapa komponen, antara lain sumber data, *data warehouse* dan penyampaian informasi kepada pengguna. Arsitektur *data warehouse* prestasi mahasiswa dapat dilihat pada Gambar 4.1.



Gambar 4.1 Arsitektur *data warehouse*

Sumber data yang digunakan dalam pembangunan *data warehouse* pada Fakultas Ilmu Komputer adalah *file excel* prestasi mahasiswa dan akademik





PRODI	ANGKATAN	NIM	TAHUN_AJARAN	IPK_LULUS

Gambar 4.3 Worksheet Akademik Mahasiswa (Lulusan dan Mahasiswa Aktif)

### 4.2.3 Perancangan Data Penelitian

Data yang digunakan pada penelitian ini adalah data akademik dan prestasi mahasiswa Fakultas Ilmu Komputer. Tahap ini dilakukan dengan merancang dan membentuk data agar sesuai dengan kebutuhan penelitian dan mengaitkannya dengan hasil rancangan *worksheet* yang telah dilakukan pada tahap sebelumnya dengan menggunakan Microsoft Excel.

### 4.2.4 Perancangan Data Warehouse

Tahap perancangan *data warehouse* prestasi mahasiswa Fakultas Ilmu Komputer dilakukan sesuai dengan sembilan tahapan metodologi.

#### 4.2.4.1 Pemilihan Proses

Berdasarkan hasil observasi yang dilakukan, proses yang dipilih pada perancangan *data warehouse* prestasi mahasiswa, yaitu proses pengolahan data kemahasiswaan dan pengolahan data akademik. Pada proses pengolahan data prestasi mahasiswa Fakultas Ilmu Komputer, terdapat beberapa data yang digunakan. Beberapa data tersebut, antara lain nama mahasiswa, program studi, prestasi, nama kegiatan, tingkat perlombaan dan tahun pelaksanaan. Sedangkan data yang digunakan pada proses pengolahan data akademik mahasiswa Fakultas Ilmu Komputer, antara lain NIM mahasiswa yang disamarkan, angkatan mahasiswa, program studi, IPK Lulus mahasiswa dan tahun akademik.

#### 4.2.4.2 Pemilihan Grain

Pada tahap pemilihan *grain*, apa yang direpresentasikan oleh *record* dalam tabel fakta ditentukan. Adapun *grain* yang ditentukan dalam perancangan *data warehouse* prestasi mahasiswa Fakultas Ilmu Komputer adalah prestasi, lulusan dan mahasiswa. Dari *grain* prestasi dapat dilakukan analisis jumlah prestasi berdasarkan program studi, jumlah prestasi berdasarkan tingkat perlombaan, jumlah prestasi berdasarkan juara dan jumlah prestasi berdasarkan tahun diperolehnya prestasi. Dari *grain* lulusan dapat dilakukan analisis jumlah lulusan berdasarkan tahun akademik, program studi dan angkatan. Sedangkan dari *grain*



mahasiswa dapat dilakukan analisis jumlah mahasiswa sesuai dengan program studi, tahun akademik dan angkatan.

#### 4.2.4.3 Identifikasi dan Penyesuaian Dimensi

Pada tahap ini, dimensi yang terkait dengan *grain* yang telah ditentukan sebelumnya diidentifikasi dan disesuaikan dalam bentuk matriks hubungan dimensi dengan *grain*. *Grain* dan dimensi yang dipilih untuk setiap tabel fakta dapat dilihat pada Tabel 4.1, Tabel 4.2 dan Tabel 4.3.

**Tabel 4.1 Diagram *information package* Prestasi Mahasiswa**

Hierarki	Dimensi			
	Tingkat	Juara	Tahun	Prodi
	Tingkat	Posisi	Tahun	Prodi
Fakta: jumlah prestasi.				

**Tabel 4.2 Diagram *information package* Mahasiswa Lulus**

Hierarki	Dimensi		
	Tahun Akademik	Angkatan	Prodi
	Tahun Semester	Angkatan	Prodi
Fakta: jumlah lulusan.			

**Tabel 4.3 Diagram *information package* Mahasiswa Aktif**

Hierarki	Dimensi		
	Tahun Akademik	Angkatan	Prodi
	Tahun Semester	Angkatan	Prodi
Fakta: jumlah mahasiswa.			

#### 4.2.4.4 Pemilihan Fakta

Pada tahap pemilihan fakta, dilakukan pemilihan fakta yang akan digunakan pada tabel fakta prestasi mahasiswa, lulusan dan mahasiswa aktif. Tahap ini dilakukan dengan melakukan representasi kebutuhan analisis sesuai dengan *grain* yang telah ditentukan. Adapun fakta yang dipilih ditunjukkan pada Tabel 4.4.



**Tabel 4.4 Fakta yang dipilih**

Fakta	Deskripsi
Prestasi Mahasiswa	Fakta yang menyimpan data prestasi mahasiswa, jumlah prestasi yang diperoleh setiap program studi, jumlah prestasi sesuai dengan juara yang diperoleh, jumlah prestasi pada setiap tingkat perlombaan dan jumlah prestasi yang diperoleh setiap tahun.
Lulusan	Fakta berisi data akademik lulusan, jumlah lulusan berdasarkan tahun akademik, program studi dan angkatan.
Mahasiswa Aktif	Fakta yang menyimpan data akademik mahasiswa aktif, jumlah mahasiswa sesuai dengan tahun akademik, program studi dan angkatan.

**4.2.4.5 Penyimpanan Pre-kalkulasi di Tabel Fakta**

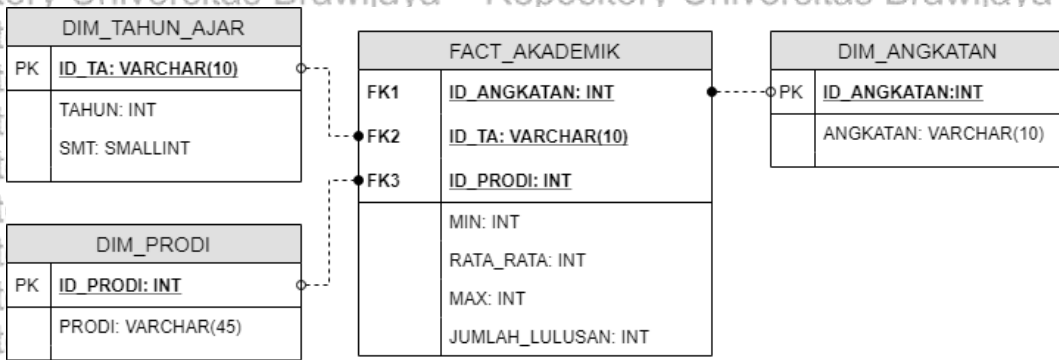
Pada tahap ini, tabel fakta yang masih memerlukan proses agregasi atau pre-kalkulasi dilakukan identifikasi. Nilai dari hasil kalkulasi tersebut nantinya akan disimpan dalam tabel fakta. Adapun pre-kalkulasi yang akan disimpan pada tabel fakta, yaitu jumlah prestasi, jumlah lulusan dan jumlah mahasiswa aktif.

**4.2.4.6 Melengkapi Tabel Dimensi**

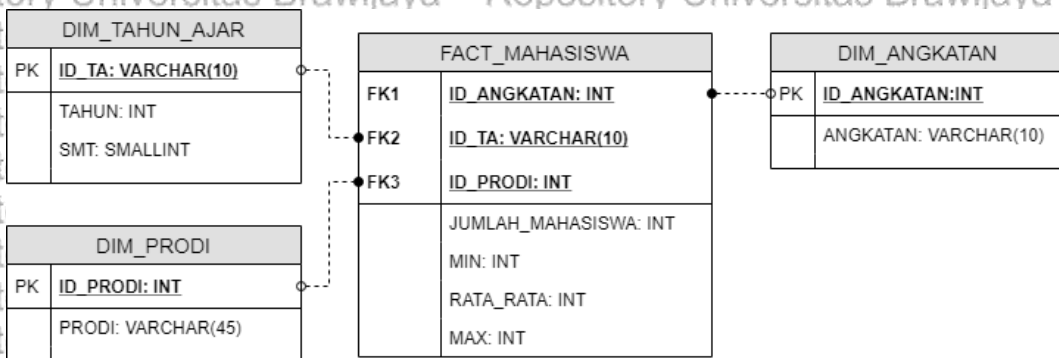
Pada tahap ini, gambaran teks terhadap dimensi ditambahkan untuk dapat digunakan pengguna untuk memahami tabel. Hubungan antara tabel fakta dan tabel dimensi yang didefinisikan dengan *Entity Relationship Diagram* dapat dilihat pada *Star Schema* pada Gambar 4.4, Gambar 4.5 dan Gambar 4.6.



**Gambar 4.4 Star Schema Prestasi Mahasiswa**



Gambar 4.5 Star Schema Lulusan



Gambar 4.6 Star Schema Mahasiswa Aktif

Adapun penjelasan dari setiap tabel dimensi yang digunakan untuk menunjang tabel fakta diuraikan sebagai berikut.

1. Tabel Dimensi Program Studi (DIM\_PRODI)

Struktur tabel dimensi Program Studi dapat dilihat pada Tabel 4.5. Tabel dimensi tersebut digunakan untuk menyimpan informasi mengenai program studi yang terdapat pada Fakultas Ilmu Komputer. *Primary key* yang digunakan pada tabel dimensi ini adalah ID\_PRODI. Adapun kolom PRODI berguna untuk menyimpan informasi mengenai program studi yang terdapat pada Fakultas Ilmu Komputer.

Tabel 4.5 Dimensi Program Studi

Kolom	Tipe Data	Panjang
ID_PRODI	INT	-
PRODI	VARCHAR	45

2. Tabel Dimensi Angkatan (DIM\_ANGKATAN)

Struktur tabel dimensi Angkatan ditunjukkan pada Tabel 4.6. Tabel dimensi Angkatan berisi informasi mengenai angkatan mahasiswa. *Primary key* yang



digunakan pada tabel dimensi ini adalah ID\_ANGKATAN. Adapun kolom ANGKATAN berisi informasi mengenai angkatan dari mahasiswa.

**Tabel 4.6 Dimensi Angkatan**

Kolom	Tipe Data	Panjang
ID_ANGKATAN	INT	5
ANGKATAN	VARCHAR	10

### 3. Tabel Dimensi Tahun (DIM\_TAHUN)

Tabel 4.7 menunjukkan struktur tabel dimensi Tahun. Tabel dimensi tersebut digunakan untuk menyimpan informasi mengenai tahun diperolehnya prestasi. *Primary key* yang digunakan pada tabel dimensi ini adalah ID\_TAHUN. Sedangkan kolom TAHUN digunakan untuk menyimpan informasi mengenai tahun.

**Tabel 4.7 Dimensi Tahun**

Kolom	Tipe Data	Panjang
ID_TAHUN	INT	5
TAHUN	INT	5

### 4. Tabel Dimensi Juara (DIM\_JUARA)

Struktur tabel dimensi Juara dapat dilihat pada Tabel 4.8. Tabel tersebut menyimpan informasi mengenai prestasi yang diperoleh oleh mahasiswa. Kolom ID\_POSISI digunakan sebagai *primary key* dari tabel dimensi tersebut. Sedangkan kolom JUARA digunakan untuk menyimpan informasi mengenai posisi juara atau prestasi yang didapatkan.

**Tabel 4.8 Dimensi Juara**

Kolom	Tipe Data	Panjang
ID_POSISI	INT	5
JUARA	VARCHAR	45

### 5. Tabel Dimensi Tingkat Lomba (DIM\_TINGKAT)

Struktur tabel dimensi Tingkat Lomba ditunjukkan pada Tabel 4.9. Tabel ini digunakan untuk menyimpan informasi mengenai tingkat perlombaan yang diikuti mahasiswa. Kolom ID\_TINGKAT digunakan sebagai *primary key* pada tabel ini, sedangkan kolom TINGKAT digunakan untuk menyimpan informasi mengenai deskripsi tingkat perlombaan.

Tabel 4.9 Dimensi Tingkat Lomba

Kolom	Tipe Data	Panjang
ID_TINGKAT	INT	-
TINGKAT	VARCHAR	45

6. Tabel Dimensi Tahun Akademik (DIM\_TAHUN\_AJAR)

Struktur tabel Tahun Akademik dapat dilihat pada Tabel 4.10. Tabel dimensi Tahun Akademik menyimpan informasi mengenai tahun akademik/ajaran. Pada tabel ini, *primary key* yang digunakan adalah ID\_TA, sedangkan kolom yang menyimpan informasi mengenai tahun akademik adalah kolom TAHUN dan kolom yang menyimpan informasi mengenai semester dalam tahun akademik adalah kolom SMT.

Tabel 4.10 Dimensi Tahun Akademik

Kolom	Tipe Data	Panjang
ID_TA	VARCHAR	10
TAHUN	INT	-
SMT	SMALLINT	-

4.2.4.7 Pemilihan Durasi Basis Data

Durasi data yang disimpan dalam *data warehouse* adalah data prestasi dan akademik mahasiswa selama empat tahun, yaitu mulai dari tahun 2016 sampai dengan 2019. Data yang digunakan untuk proses pengujian pada penelitian ini adalah data akademik (lulusan dan mahasiswa aktif) dan data prestasi mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya.

4.2.4.8 Pelacakan Perubahan Dimensi secara Perlahan

Atribut pada *data warehouse* dapat mengalami perubahan dalam waktu yang cukup lama. Oleh sebab itu, untuk menjaga konsistensi data maka perlu dilakukan pembaharuan. Pada tahap ini, perubahan dimensi dikategorikan sebagai *Slowly Changing Dimension* tipe 1, dimana data lama yang mengalami perubahan pada *data warehouse* akan ditimpa dengan data baru.

4.2.4.9 Penentuan Prioritas Query dan Mode Query

Pada tahap ini dilakukan penentuan proses *backup* data dan proses ETL (*Extract, Transform, Load*). Proses ETL dilakukan untuk memuat data dari sumber eksternal ke dalam *data warehouse*. Dalam proses ETL, data yang akan dimuat ke dalam *data warehouse* akan dikumpulkan dan ditransformasi terlebih dahulu sehingga data yang akan dimuat dapat memenuhi kriteria kebutuhan *data*

*warehouse*. Adapun kebutuhan mengenai proses *backup* data dapat dilihat pada Tabel 4.11.

**Tabel 4.11** Tabel kebutuhan *backup*

Pelaku <i>Backup</i>	Waktu	Keterangan
Administrator Basis Data	Akhir tahun	Proses <i>backup</i> dilakukan untuk mengatasi gagalnya proses ETL fakta Prestasi.
Administrator Basis Data	Akhir periode akademik	Proses <i>backup</i> dilakukan untuk mengatasi gagalnya proses ETL fakta Mahasiswa Aktif.
Administrator Basis Data	Setiap bulan	Proses <i>backup</i> dilakukan untuk mengatasi gagalnya proses ETL fakta Lulusan.

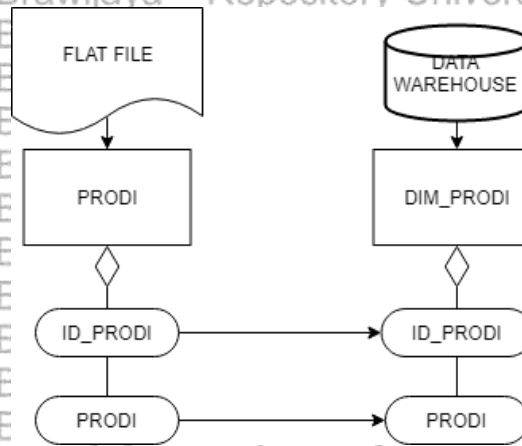
#### 4.2.5 Perancangan Proses ETL

Setelah *data warehouse* dirancang pada tahap sebelumnya, tahap berikutnya adalah melakukan perancangan proses ETL dari sumber data eksternal menuju ke *data warehouse*. Proses ETL terdiri dari tiga tahap, yaitu tahap ekstraksi data dari sumber data, kemudian tahap transformasi data dimana data yang telah diekstraksi ditampung sementara pada *staging area* untuk kemudian ditransformasi ke dalam bentuk data yang dibutuhkan di *data warehouse*, lalu tahap yang terakhir adalah proses memuat data ke dalam *data warehouse*.

Tahap perancangan proses ETL dilakukan untuk merepresentasikan proses ETL. Pada tahap ini, dilakukan pemetaan data dari sumber data menuju *staging area* dan kemudian *data warehouse*.

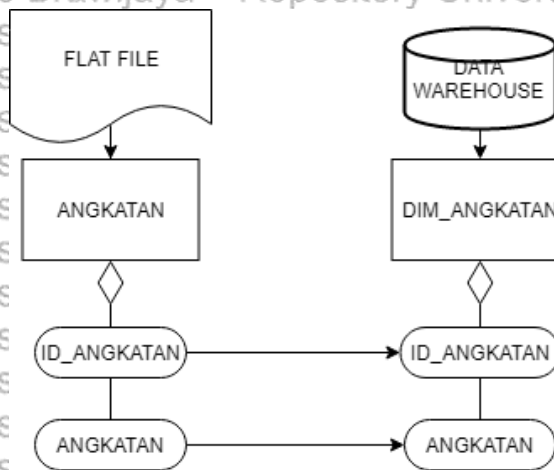
##### 4.2.5.1 Dimensi Program Studi

Gambar 4.7 merepresentasikan proses ETL dari *flat file* PRODI menuju tabel DIM\_PRODI. Pada gambar tersebut, data pada DIM\_PRODI berasal dari *flat file* PRODI. Pada pemetaan tersebut, tidak terdapat perbedaan kolom yang dibutuhkan antara *flat file* PRODI dengan DIM\_PRODI, sehingga proses transformasi tidak perlu dilakukan.



Gambar 4.7 Pemetaan data dimensi Program Studi

#### 4.2.5.2 Dimensi Angkatan



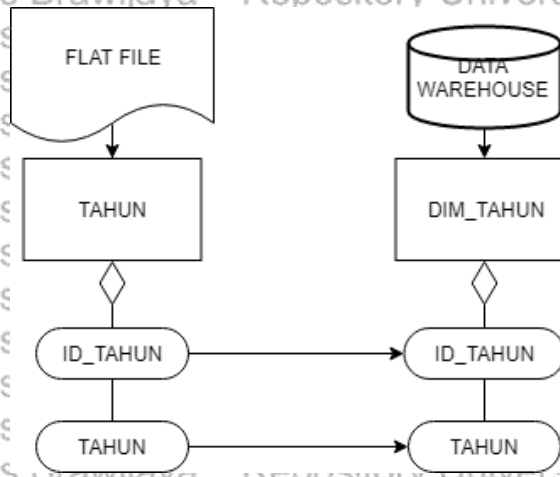
Gambar 4.8 Pemetaan data dimensi Angkatan

Representasi proses ETL dari *flat file* ANGGKATAN menuju tabel DIM\_ANGKATAN ditunjukkan pada Gambar 4.8. Pada gambar tersebut, data pada DIM\_ANGKATAN berasal dari *flat file* ANGGKATAN. Pada pemetaan tersebut, tidak terdapat perbedaan kolom yang dibutuhkan antara *flat file* ANGGKATAN dengan DIM\_ANGKATAN sehingga proses transformasi tidak perlu dilakukan.

#### 4.2.5.3 Dimensi Tahun

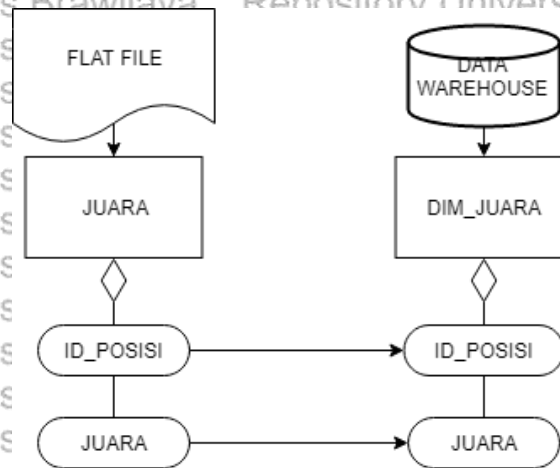
Gambar 4.9 merepresentasikan proses ETL dari *flat file* TAHUN menuju tabel DIM\_TAHUN. Pada gambar tersebut, data pada DIM\_TAHUN berasal dari *flat file* TAHUN. Pada pemetaan tersebut, tidak terdapat perbedaan kolom yang dibutuhkan antara *flat file* TAHUN dengan DIM\_TAHUN, sehingga proses transformasi tidak perlu dilakukan.





Gambar 4.9 Pemetaan data dimensi Tahun

#### 4.2.5.4 Dimensi Juara

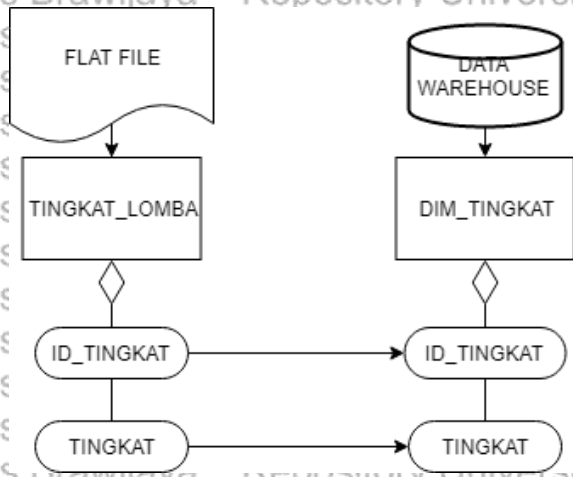


Gambar 4.10 Pemetaan data dimensi Juara

Representasi proses ETL dari *flat file* JUARA menuju tabel DIM\_JUARA dapat dilihat pada Gambar 4.10. Pada gambar tersebut, data pada DIM\_JUARA berasal dari *flat file* JUARA. Pada pemetaan tersebut, tidak terdapat perbedaan kolom yang dibutuhkan antara *flat file* JUARA dengan DIM\_JUARA, sehingga proses transformasi tidak perlu dilakukan.

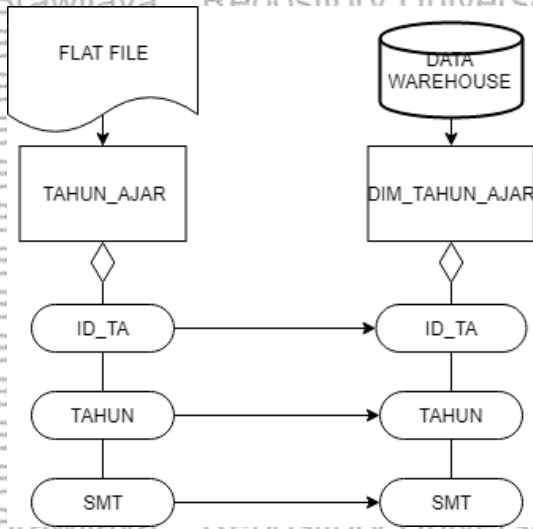
#### 4.2.5.5 Dimensi Tingkat Lomba

Representasi proses ETL dari *flat file* TINGKAT LOMBA menuju tabel DIM\_TINGKAT ditunjukkan pada Gambar 4.11. Pada gambar tersebut, data pada DIM\_TINGKAT berasal dari *flat file* TINGKAT\_LOMBA. Pada pemetaan tersebut, tidak terdapat perbedaan kolom yang dibutuhkan antara *flat file* TINGKAT\_LOMBA dengan DIM\_TINGKAT, sehingga proses transformasi tidak perlu dilakukan.



Gambar 4.11 Pemetaan data dimensi Tingkat Lomba

4.2.5.6 Dimensi Tahun Akademik



Gambar 4.12 Pemetaan data dimensi Tahun Akademik

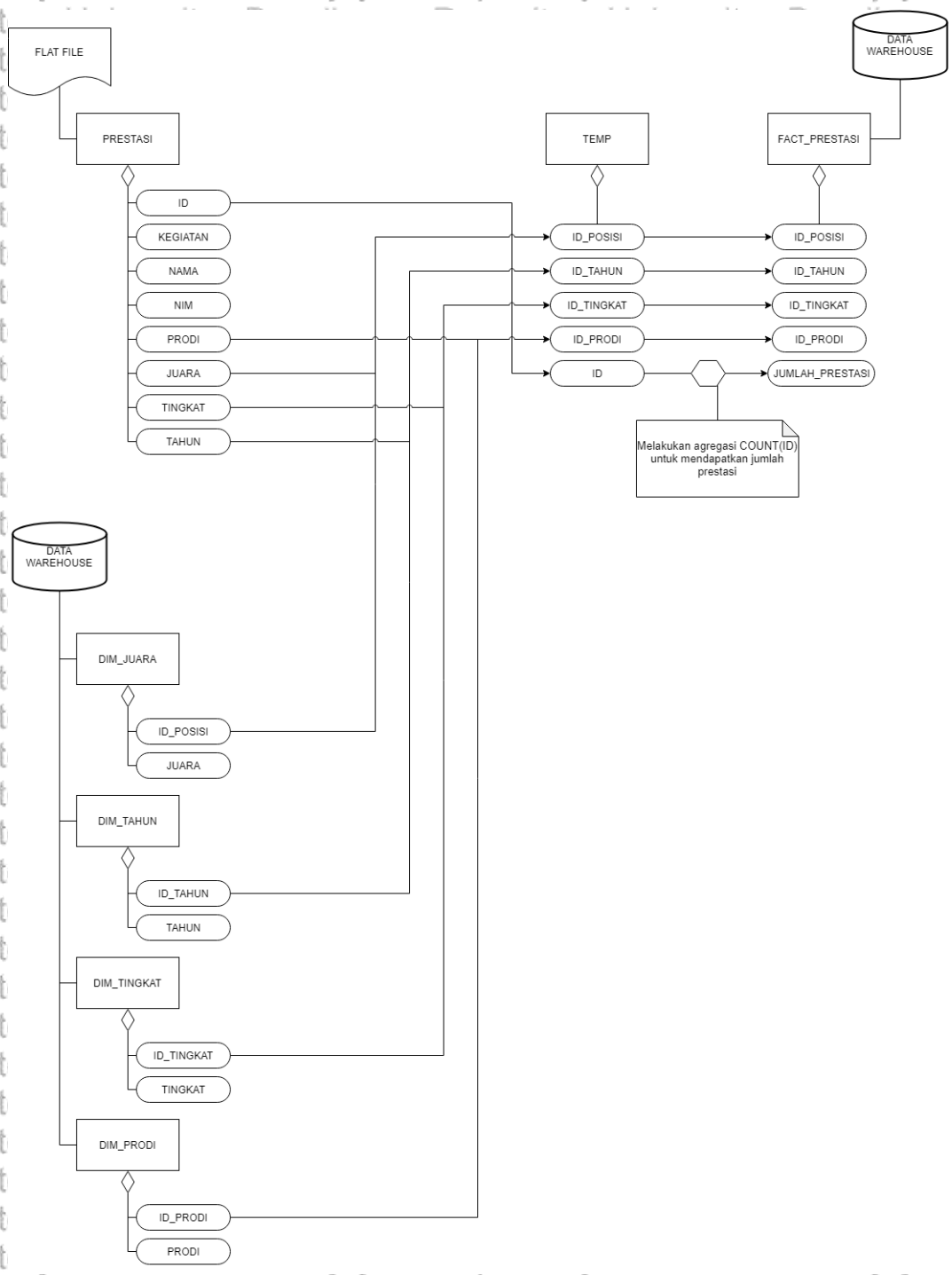
Gambar 4.12 merepresentasikan proses ETL dari *flat file* TAHUN\_AJAR menuju tabel DIM\_TAHUN\_AJAR. Pada gambar tersebut, data pada DIM\_TAHUN\_AJAR berasal dari *flat file* TAHUN\_AJAR. Pada pemetaan tersebut, tidak terdapat perbedaan kolom yang dibutuhkan antara *flat file* TAHUN\_AJAR dengan DIM\_TAHUN\_AJAR, sehingga proses transformasi tidak perlu dilakukan.

4.2.5.7 Fakta Prestasi Mahasiswa

Gambar 4.13 menunjukkan bahwa data yang terdapat pada tabel FACT\_PRESTASI berasal dari *flat file* PRESTASI. Proses transformasi dilakukan karena pemetaan tabel belum sesuai dengan kebutuhan *data warehouse*. Adapun proses transformasi yang dilakukan, adalah:

1. Dilakukan *join* antara *flat file* PRESTASI dengan tabel Dimensi Juara, Dimensi Tahun, Dimensi Tingkat dan Dimensi Program Studi. Setelah itu, kolom-kolom pada *flat file* diseleksi sesuai dengan kolom yang dibutuhkan pada tabel fakta.

2. Proses transformasi dilakukan untuk mendapatkan data agregasi yang dibutuhkan. Adapun kolom yang mengalami transformasi tersebut adalah kolom ID.
3. Data dimuat dan disimpan pada tabel fakta.



Gambar 4.13 Pemetaan data fakta Prestasi Mahasiswa



#### 4.2.5.8 Fakta Lulusan

Gambar 4.14 menjelaskan bahwa data yang terdapat pada tabel FACT\_AKADEMIK berasal dari *flat file* LULUSAN. Proses transformasi dilakukan karena pemetaan tabel belum sesuai dengan kebutuhan *data warehouse*. Adapun proses transformasi yang dilakukan, adalah:

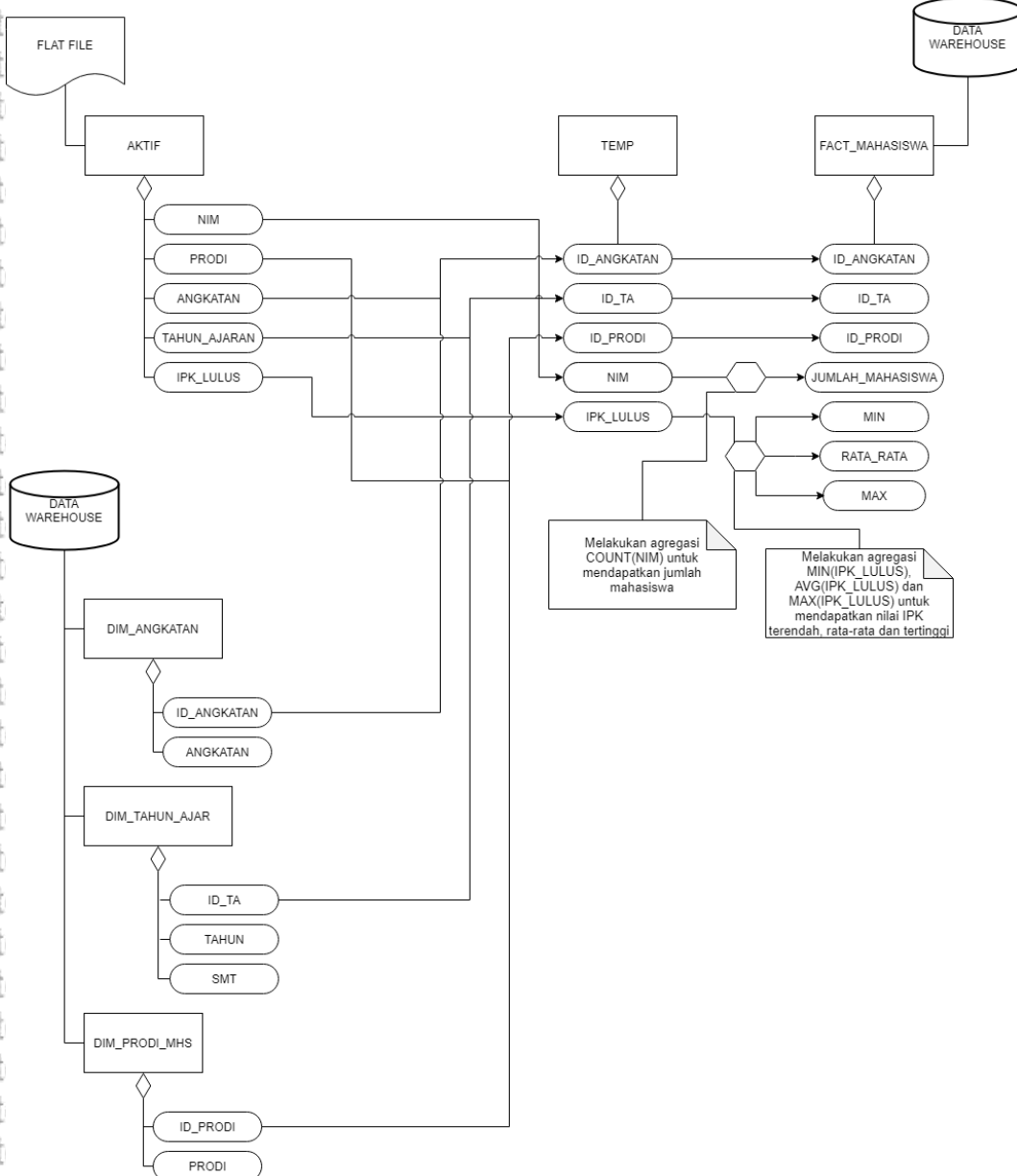
1. Dilakukan *join* antara *flat file* LULUSAN dengan tabel Dimensi Angkatan, Dimensi Tahun Akademik dan Dimensi Program Studi. Kemudian dilakukan seleksi pada kolom-kolom dari *flat file* yang disesuaikan dengan kebutuhan pada tabel fakta
2. Proses transformasi dilakukan untuk mendapatkan data agregasi yang dibutuhkan. Adapun kolom yang mengalami transformasi tersebut adalah kolom NIM.
3. Data dimuat dan disimpan pada tabel fakta.

#### 4.2.5.9 Fakta Mahasiswa Aktif

Gambar 4.15 menjelaskan bahwa data yang terdapat pada tabel FACT\_MAHASISWA berasal dari *flat file* AKTIF. Proses transformasi dilakukan karena pemetaan tabel belum sesuai dengan kebutuhan *data warehouse*. Adapun proses transformasi yang dilakukan, adalah:

1. Dilakukan *join* antara *flat file* AKTIF dengan tabel Dimensi Angkatan, Dimensi Tahun Akademik dan Dimensi Program Studi. Lalu kolom-kolom pada *flat file* dilakukan seleksi untuk menyesuaikan dengan kebutuhan tabel fakta pada *data warehouse*.
2. Proses transformasi dilakukan untuk mendapatkan data agregasi yang dibutuhkan. Adapun kolom yang mengalami transformasi tersebut adalah kolom NIM.
3. Data dimuat dan disimpan pada tabel fakta.





Gambar 4.15 Pemetaan data fakta Mahasiswa Aktif

### 4.3 Implementasi

#### 4.3.1 Implementasi Data Warehouse

Implementasi data warehouse dilakukan dengan mengimplementasi rancangan skema data warehouse ke dalam bentuk Data Definition Language (DDL). Implementasi dilakukan dengan melakukan deployment menggunakan tools IBM Data Studio. Pada implementasi DDL, buffer pool dan table space dibuat pertama kali. Buffer pool berfungsi untuk menyimpan data dan data index. Sedangkan table space berfungsi untuk menyimpan objek/data yang terdapat dalam basis data, baik data reguler, data index maupun data berukuran besar.



Adapun DDL yang dibuat untuk membuat *buffer pool* dan *table space* ditunjukkan pada Tabel 4.12.

**Tabel 4.12 Implementasi DDL untuk *buffer pool* dan *table space***

```

1. CREATE BUFFERPOOL BPDW IMMEDIATE ALL DBPARTITONNUMS SIZE
   1000 AUTOMATIC PAGESIZE 16K;
2. CREATE REGULAR TABLESPACE TSOLAP IN DATABASE PARTITION
   GROUP IBMDEFAULTGROUP PAGESIZE 16K MANAGED BY AUTOMATIC
   STORAGE USING STOGROUP IBMSTOGROUP AUTORESIZE YES
3. BUFFERPOOL BPDW OVERHEAD INHERIT TRANSFERRATE INHERIT
   DROPPED TABLE RECOVERY ON DATA TAG NONE;
   CREATE REGULAR TABLESPACE TSINDEX IN DATABASE PARTITION
   GROUP IBMDEFAULTGROUP PAGESIZE 16K MANAGED BY AUTOMATIC
4. STORAGE USING STOGROUP IBMSTOGROUP AUTORESIZE YES
   BUFFERPOOL BPDW OVERHEAD INHERIT TRANSFERRATE INHERIT
   DROPPED TABLE RECOVERY ON DATA TAG NONE;
   CREATE LARGE TABLESPACE TSLARGE IN DATABASE PARTITION
   GROUP IBMDEFAULTGROUP PAGESIZE 16K MANAGED BY AUTOMATIC
   STORAGE USING STOGROUP IBMSTOGROUP AUTORESIZE YES
   BUFFERPOOL BPDW OVERHEAD INHERIT TRANSFERRATE INHERIT
   DROPPED TABLE RECOVERY ON DATA TAG NONE;

```

Penjelasan dari kode DDL di atas adalah:

Baris 1: Merupakan perintah untuk membuat sebuah *buffer pool* bernama BPDW dengan page size 16K.

Baris 2: Merupakan perintah untuk membuat *table space* reguler bernama TSOLAP dengan menggunakan *storage group* otomatis. *Table space* ini berfungsi sebagai tempat penyimpanan data reguler.

Baris 3: Merupakan perintah untuk membuat *table space* reguler bernama TSINDEX dengan menggunakan *storage group* otomatis. *Table space* ini berfungsi sebagai tempat penyimpanan data index.

Baris 4: Merupakan perintah untuk membuat long *table space* bernama TSLARGE dengan menggunakan *storage group* otomatis. *Table space* ini berfungsi sebagai tempat penyimpanan data dengan sifat LOB (*Large Object*).

#### 4.3.1.1 Tabel Dimensi Juara (DIM\_JUARA)

Dimensi Juara merupakan dimensi yang berfungsi untuk menyimpan informasi mengenai posisi atau prestasi yang diperoleh. Adapun DDL yang dibuat untuk membuat tabel dimensi Juara dapat dilihat pada Tabel 4.13.

**Tabel 4.13 Implementasi DDL untuk dimensi Juara**

```

1. CREATE TABLE DB2.DIM_JUARA (
2. ID_POSISI INTEGER NOT NULL,
3. JUARA VARCHAR (45),
4. PRIMARY KEY (ID_POSISI)
5. ) IN TSOLAP INDEX IN TSINDEX LONG IN TSLARGE;

```

Penjelasan dari kode DDL di atas adalah:

Baris 1: Merupakan perintah untuk membuat tabel bernama DIM\_JUARA.



Baris 2: Merupakan deklarasi atribut bernama ID\_POSISI dengan tipe data INTEGER dan bersifat NOT NULL.

Baris 3: Merupakan deklarasi atribut bernama JUARA bertipe data VARCHAR dengan panjang 45 karakter.

Baris 4: Merupakan deklarasi untuk menjadikan atribut bernama ID\_POSISI sebagai *primary key* dari tabel bernama DIM\_JUARA.

Baris 5: Merupakan deklarasi untuk menggunakan *table space* bernama TSOLAP sebagai tempat penyimpanan data reguler, TSINDEX sebagai tempat penyimpanan data index dan TSLARGE sebagai tempat penyimpanan data berjenis LOB pada tabel DIM\_JUARA.

#### 4.3.1.2 Tabel Dimensi Tingkat Lomba (DIM\_TINGKAT)

Dimensi Tingkat Lomba merupakan dimensi yang berfungsi untuk menyimpan informasi mengenai tingkat perlombaan yang diikuti. Adapun DDL yang dibuat untuk membuat tabel dimensi Tingkat Lomba dapat dilihat pada Tabel 4.14.

**Tabel 4.14 Implementasi DDL untuk dimensi Tingkat Lomba**

```
1. CREATE TABLE DB2.DIM_TINGKAT (
2.     ID_TINGKAT INTEGER NOT NULL,
3.     TINGKAT VARCHAR (45),
4.     PRIMARY KEY (ID_TINGKAT)
5. ) IN TSOLAP INDEX IN TSINDEX LONG IN TSLARGE;
```

Penjelasan dari kode DDL di atas adalah:

Baris 1: Merupakan perintah untuk membuat tabel bernama DIM\_TINGKAT.

Baris 2: Merupakan deklarasi atribut bernama ID\_TINGKAT dengan tipe data INTEGER dan bersifat NOT NULL.

Baris 3: Merupakan deklarasi atribut bernama TINGKAT bertipe data VARCHAR dengan panjang 45 karakter.

Baris 4: Merupakan deklarasi untuk menjadikan atribut bernama ID\_TINGKAT sebagai *primary key* dari tabel bernama DIM\_TINGKAT.

Baris 5: Merupakan deklarasi untuk menggunakan *table space* bernama TSOLAP sebagai tempat penyimpanan data reguler, TSINDEX sebagai tempat penyimpanan data index dan TSLARGE sebagai tempat penyimpanan data berjenis LOB pada tabel DIM\_TINGKAT.

#### 4.3.1.3 Tabel Dimensi Tahun (DIM\_TAHUN)

Dimensi Tahun adalah dimensi yang berfungsi untuk menyimpan informasi mengenai tahun diperolehnya prestasi. Adapun DDL yang dibuat untuk membuat tabel dimensi Tahun dapat dilihat pada Tabel 4.15.

**Tabel 4.15 Implementasi DDL untuk dimensi Tahun**

```
1. CREATE TABLE DB2.DIM_TAHUN (
2.     ID_TAHUN INTEGER NOT NULL,
3.     TAHUN INTEGER,
```





```
4. PRIMARY KEY (ID_TAHUN)
5. ) IN TSOLAP INDEX IN TSINDEX LONG IN TSLARGE;
```

Penjelasan dari kode DDL di atas adalah:

- Baris 1: Merupakan perintah untuk membuat tabel bernama DIM\_TAHUN.
- Baris 2: Merupakan deklarasi atribut bernama ID\_TAHUN bertipe data INTEGER dan bersifat NOT NULL.
- Baris 3: Merupakan deklarasi atribut bernama TAHUN bertipe data INTEGER.
- Baris 4: Merupakan deklarasi untuk menjadikan atribut bernama ID\_TAHUN menjadi *primary key* dari tabel bernama DIM\_TAHUN.
- Baris 5: Merupakan deklarasi untuk menggunakan *table space* bernama TSOLAP sebagai tempat penyimpanan data reguler, TSINDEX sebagai tempat penyimpanan data index dan TSLARGE sebagai tempat penyimpanan data berjenis LOB pada tabel DIM\_TAHUN.

#### 4.3.1.4 Tabel Dimensi Program Studi (DIM\_PRODI)

Dimensi Program Studi adalah dimensi yang berfungsi untuk menyimpan informasi mengenai program studi pada Fakultas Ilmu Komputer Universitas Brawijaya. Adapun DDL yang dibuat untuk membuat tabel dimensi Program Studi dapat dilihat pada Tabel 4.16.

**Tabel 4.16 Implementasi DDL untuk dimensi Program Studi**

```
1. CREATE TABLE DB2.DIM_PRODI (
2. ID_PRODI INTEGER NOT NULL,
3. PRODI VARCHAR (45),
4. PRIMARY KEY (ID_PRODI)
5. ) IN TSOLAP INDEX IN TSINDEX LONG IN TSLARGE;
```

Penjelasan dari kode DDL di atas adalah:

- Baris 1: Merupakan perintah untuk membuat tabel bernama DIM\_PRODI.
- Baris 2: Merupakan deklarasi atribut bernama ID\_PRODI bertipe data INTEGER dan bersifat NOT NULL.
- Baris 3: Merupakan deklarasi atribut bernama PRODI bertipe data VARCHAR dengan panjang 45 karakter.
- Baris 4: Merupakan deklarasi untuk menjadikan atribut bernama ID\_PRODI sebagai *primary key* dari tabel bernama DIM\_PRODI.
- Baris 5: Merupakan deklarasi untuk menggunakan *table space* bernama TSOLAP sebagai tempat penyimpanan data reguler, TSINDEX sebagai tempat penyimpanan data index dan TSLARGE sebagai tempat penyimpanan data berjenis LOB pada tabel DIM\_PRODI.

#### 4.3.1.5 Tabel Dimensi Angkatan (DIM\_ANGKATAN)

Dimensi Angkatan adalah dimensi yang berfungsi untuk menyimpan informasi mengenai angkatan mahasiswa. Adapun DDL yang dibuat untuk membuat tabel dimensi Angkatan dapat dilihat pada Tabel 4.17.



**Tabel 4.17 Implementasi DDL untuk dimensi Angkatan**

```

1. CREATE TABLE DB2.DIM_ANGKATAN (
2.     ID_ANGKATAN INTEGER NOT NULL,
3.     ANGGKATAN VARCHAR (10),
4.     PRIMARY KEY (ID_ANGKATAN)
5. ) IN TSOLAP INDEX IN TSINDEX LONG IN TSLARGE;

```

Penjelasan dari kode DDL di atas adalah:

Baris 1: Merupakan perintah untuk membuat tabel bernama DIM\_ANGKATAN.

Baris 2: Merupakan deklarasi atribut bernama ID\_ANGKATAN bertipe data INTEGER dan bersifat NOT NULL.

Baris 3: Merupakan deklarasi atribut bernama ANGGKATAN bertipe data VARCHAR dengan panjang 10 karakter.

Baris 4: Merupakan deklarasi untuk menjadikan atribut bernama ID\_ANGKATAN sebagai *primary key* dari tabel bernama DIM\_ANGKATAN.

Baris 5: Merupakan deklarasi untuk menggunakan *table space* bernama TSOLAP sebagai tempat penyimpanan data reguler, TSINDEX sebagai tempat penyimpanan data index dan TSLARGE sebagai tempat penyimpanan data berjenis LOB pada tabel DIM\_ANGKATAN.

#### 4.3.1.6 Tabel Dimensi Tahun Akademik (DIM\_TAHUN\_AJAR)

Dimensi Tahun Akademik adalah dimensi yang berfungsi untuk menyimpan informasi mengenai tahun akademik/ajaran. Adapun DDL yang dibuat untuk membuat tabel dimensi Tahun Akademik dapat dilihat pada Tabel 4.18.

**Tabel 4.18 Implementasi DDL untuk dimensi Tahun Akademik**

```

1. CREATE TABLE DB2.DIM_TAHUN_AJAR (
2.     ID_TA VARCHAR (10) NOT NULL,
3.     TAHUN INTEGER,
4.     SMT SMALLINT,
5.     PRIMARY KEY (ID_TA)
6. ) IN TSOLAP INDEX IN TSINDEX LONG IN TSLARGE;

```

Penjelasan dari kode DDL di atas adalah:

Baris 1: Merupakan perintah untuk membuat tabel bernama DIM\_TAHUN\_AJAR.

Baris 2: Merupakan deklarasi atribut bernama ID\_TA bertipe data VARCHAR dengan Panjang 10 karakter dan bersifat NOT NULL.

Baris 3: Merupakan deklarasi atribut bernama TAHUN bertipe data INTEGER.

Baris 4: Merupakan deklarasi atribut bernama SMT bertipe data SMALLINT.

Baris 5: Merupakan deklarasi untuk menjadikan atribut bernama ID\_TA menjadi *primary key* dari tabel bernama DIM\_TAHUN\_AJAR.

Baris 6: Merupakan deklarasi untuk menggunakan *table space* bernama TSOLAP sebagai tempat penyimpanan data reguler, TSINDEX sebagai tempat

penyimpanan data index dan TSLARGE sebagai tempat penyimpanan data berjenis LOB pada tabel DIM\_TAHUN\_AJAR.

#### 4.3.1.7 Tabel Fakta Prestasi Mahasiswa (FACT\_PRESTASI)

Tabel fakta prestasi mahasiswa adalah tabel yang berfungsi untuk menyimpan informasi mengenai jumlah prestasi mahasiswa. Tabel tersebut menggunakan empat dimensi sebagai referensi untuk mendukung jalannya fungsi pada tabel tersebut, yaitu dimensi program studi, dimensi tahun, dimensi tingkat dan dimensi juara. Adapun DDL yang dibuat untuk membuat tabel fakta prestasi mahasiswa ditunjukkan pada Tabel 4.19.

**Tabel 4.19 Implementasi DDL untuk fakta Prestasi Mahasiswa**

```
1. CREATE TABLE DB2.FACT_PRESTASI (  
2. ID_POSISI INTEGER NOT NULL,  
3. ID_TAHUN INTEGER NOT NULL,  
4. ID_TINGKAT INTEGER NOT NULL,  
5. ID_PRODI INTEGER NOT NULL,  
6. JUMLAH_PRESTASI INTEGER,  
7. PRIMARY KEY (ID_POSISI, ID_TAHUN, ID_TINGKAT,  
8. ID_PRODI),  
9. FOREIGN KEY (ID_POSISI) REFERENCES DB2.DIM_JUARA  
10. (ID_POSISI),  
11. FOREIGN KEY (ID_TAHUN) REFERENCES DB2.DIM_TAHUN  
12. (ID_TAHUN),  
13. FOREIGN KEY (ID_TINGKAT) REFERENCES DB2.DIM_TINGKAT  
14. (ID_TINGKAT),  
15. FOREIGN KEY (ID_PRODI) REFERENCES DB2.DIM_PRODI  
16. (ID_PRODI)  
17. ) IN TSOLAP INDEX IN TSINDEX LONG IN TSLARGE;
```

Penjelasan dari kode DDL di atas adalah:

Baris 1: Merupakan perintah untuk membuat tabel bernama FACT\_PRESTASI.

Baris 2: Merupakan deklarasi atribut bernama ID\_POSISI dengan tipe data INTEGER dan bersifat NOT NULL.

Baris 3: Merupakan deklarasi atribut bernama ID\_TAHUN dengan tipe data INTEGER dan bersifat NOT NULL.

Baris 4: Merupakan deklarasi atribut bernama ID\_TINGKAT dengan tipe data INTEGER dan bersifat NOT NULL.

Baris 5: Merupakan deklarasi atribut bernama ID\_PRODI dengan tipe data INTEGER dan bersifat NOT NULL.

Baris 6: Merupakan deklarasi atribut bernama JUMLAH\_PRESTASI dengan tipe data INTEGER yang berguna untuk menyimpan data fakta.

Baris 7: Merupakan deklarasi atribut bernama ID\_POSISI, ID\_TAHUN, ID\_TINGKAT dan ID\_PRODI sebagai *primary key*.

Baris 8: Merupakan deklarasi *foreign key* bernama ID\_POSISI yang digunakan untuk menghubungkan tabel fakta prestasi mahasiswa dengan dimensi juara.



Baris 9: Merupakan deklarasi *foreign key* bernama ID\_TAHUN yang digunakan untuk menghubungkan tabel fakta prestasi mahasiswa dengan dimensi tahun.

Baris 10: Merupakan deklarasi *foreign key* bernama ID\_TINGKAT yang digunakan untuk menghubungkan tabel fakta prestasi mahasiswa dengan dimensi tingkat.

Baris 11: Merupakan deklarasi *foreign key* bernama ID\_PRODI yang digunakan untuk menghubungkan tabel fakta prestasi mahasiswa dengan dimensi program studi.

Baris 12: Merupakan deklarasi untuk menggunakan *table space* bernama TSOLAP sebagai tempat penyimpanan data reguler, TSINDEX sebagai tempat penyimpanan data index dan TSLARGE sebagai tempat penyimpanan data berjenis LOB pada tabel FACT\_PRESTASI.

#### 4.3.1.8 Tabel Fakta Lulusan (FACT\_AKADEMIK)

Tabel fakta lulusan merupakan tabel yang berfungsi untuk menyimpan informasi mengenai jumlah lulusan. Tabel tersebut menggunakan tiga dimensi sebagai referensi untuk mendukung jalannya fungsi pada tabel tersebut, yaitu dimensi program studi, dimensi tahun akademik dan dimensi angkatan. Adapun DDL yang dibuat untuk membuat tabel fakta lulusan ditunjukkan pada Tabel 4.20.

**Tabel 4.20 Implementasi DDL untuk fakta Lulusan**

```

1. CREATE TABLE DB2.FACT_AKADEMIK (
2. ID_ANGKATAN INTEGER NOT NULL,
3. ID_TA VARCHAR (10) NOT NULL,
4. ID_PRODI INTEGER NOT NULL,
5. MIN DOUBLE,
6. RATA_RATA DOUBLE,
7. MAX DOUBLE,
8. JUMLAH_LULUSAN INTEGER,
9. PRIMARY KEY (ID_ANGKATAN, ID_TA, ID_PRODI),
10. FOREIGN KEY (ID_ANGKATAN) REFERENCES
    DB2.DIM_ANGKATAN (ID_ANGKATAN),
11. FOREIGN KEY (ID_TA) REFERENCES DB2.DIM_TAHUN_AJAR
    (ID_TA),
12. FOREIGN KEY (ID_PRODI) REFERENCES DB2.DIM_PRODI
    (ID_PRODI)
13. ) IN TSOLAP INDEX IN TSINDEX LONG IN TSLARGE;
```

Penjelasan dari kode DDL di atas adalah:

Baris 1: Merupakan perintah untuk membuat tabel bernama FACT\_AKADEMIK.

Baris 2: Merupakan deklarasi atribut bernama ID\_ANGKATAN dengan tipe data INTEGER dan bersifat NOT NULL.

Baris 3: Merupakan deklarasi atribut bernama ID\_TA dengan tipe data VARCHAR dengan panjang karakter 10 dan bersifat NOT NULL.

Baris 4: Merupakan deklarasi atribut bernama ID\_PRODI dengan tipe data INTEGER dan bersifat NOT NULL.



Baris 5: Merupakan deklarasi atribut bernama MIN dengan tipe data DOUBLE yang berfungsi untuk menyimpan data nilai IPK terendah.

Baris 6: Merupakan deklarasi atribut bernama RATA\_RATA dengan tipe data DOUBLE yang berfungsi untuk menyimpan data nilai IPK rata-rata.

Baris 7: Merupakan deklarasi atribut bernama MAX dengan tipe data DOUBLE yang berfungsi untuk menyimpan data nilai IPK tertinggi.

Baris 8: Merupakan deklarasi atribut bernama JUMLAH\_LULUSAN dengan tipe data INTEGER yang berfungsi untuk menyimpan data fakta.

Baris 9: Merupakan deklarasi atribut bernama ID\_ANGKATAN, ID\_TA dan ID\_PRODI sebagai *primary key*.

Baris 10: Merupakan deklarasi *foreign key* bernama ID\_ANGKATAN yang digunakan untuk menghubungkan tabel fakta lulusan dengan dimensi angkatan.

Baris 11: Merupakan deklarasi *foreign key* bernama ID\_TA yang digunakan untuk menghubungkan tabel fakta lulusan dengan dimensi tahun akademik.

Baris 12: Merupakan deklarasi *foreign key* bernama ID\_PRODI yang digunakan untuk menghubungkan tabel fakta lulusan dengan dimensi program studi.

Baris 13: Merupakan deklarasi untuk menggunakan *table space* bernama TSOLAP sebagai tempat penyimpanan data reguler, TSINDEX sebagai tempat penyimpanan data index dan TSLARGE sebagai tempat penyimpanan data berjenis LOB pada tabel FACT\_AKADEMIK.

#### 4.3.1.9 Tabel Fakta Mahasiswa Aktif (FACT\_MAHASISWA)

Tabel fakta mahasiswa aktif adalah tabel yang berfungsi untuk menyimpan informasi mengenai jumlah mahasiswa aktif. Tabel tersebut menggunakan tiga dimensi sebagai referensi untuk mendukung jalannya fungsi pada tabel tersebut, yaitu dimensi program studi, dimensi tahun akademik dan dimensi angkatan.

Adapun DDL yang dibuat untuk membuat tabel fakta mahasiswa aktif ditunjukkan pada Tabel 4.21:

**Tabel 4.21 Implementasi DDL untuk fakta Mahasiswa Aktif**

```

1. CREATE TABLE DB2.FACT_MAHASISWA (
2. ID_ANGKATAN INTEGER NOT NULL,
3. ID_TA VARCHAR (10) NOT NULL,
4. ID_PRODI INTEGER NOT NULL,
5. JUMLAH_MAHASISWA INTEGER,
6. MIN DOUBLE,
7. RATA_RATA DOUBLE,
8. MAX DOUBLE,
9. PRIMARY KEY (ID_ANGKATAN, ID_TA, ID_PRODI),
10. FOREIGN KEY (ID_ANGKATAN) REFERENCES
    DB2.DIM_ANGKATAN (ID_ANGKATAN),
11. FOREIGN KEY (ID_TA) REFERENCES DB2.DIM_TAHUN_AJAR
    (ID_TA),

```



```
12. FOREIGN KEY (ID_PRODI) REFERENCES DB2.DIM.PRODI
    (ID_PRODI)
13. ) IN TSOLAP INDEX IN TSINDEX LONG IN TSLARGE;
```

Penjelasan dari kode DDL di atas adalah:

Baris 1: Merupakan perintah untuk membuat tabel bernama `FACT_MAHASISWA`.

Baris 2: Merupakan deklarasi atribut bernama `ID_ANGKATAN` bertipe data `INTEGER` dan bersifat `NOT NULL`.

Baris 3: Merupakan deklarasi atribut bernama `ID_TA` bertipe data `VARCHAR` dengan panjang 10 karakter dan bersifat `NOT NULL`.

Baris 4: Merupakan deklarasi atribut bernama `ID_PRODI` bertipe data `INTEGER` dan bersifat `NOT NULL`.

Baris 5: Merupakan deklarasi atribut bernama `JUMLAH_MAHASISWA` bertipe data `INTEGER` yang digunakan untuk menyimpan data fakta.

Baris 6: Merupakan deklarasi atribut bernama `MIN` bertipe data `DOUBLE` yang digunakan untuk menyimpan data nilai IPK terendah.

Baris 7: Merupakan deklarasi atribut bernama `RATA_RATA` bertipe data `DOUBLE` yang digunakan untuk menyimpan data nilai IPK rata-rata.

Baris 8: Merupakan deklarasi atribut bernama `MAX` bertipe data `DOUBLE` yang digunakan untuk menyimpan data nilai IPK tertinggi.

Baris 9: Merupakan deklarasi atribut bernama `ID_ANGKATAN`, `ID_TA` dan `ID_PRODI` sebagai *primary key*.

Baris 10: Merupakan deklarasi *foreign key* bernama `ID_ANGKATAN` yang digunakan untuk menghubungkan tabel fakta mahasiswa aktif dengan dimensi angkatan.

Baris 11: Merupakan deklarasi *foreign key* bernama `ID_TA` yang digunakan untuk menghubungkan tabel fakta mahasiswa aktif dengan dimensi tahun akademik.

Baris 12: Merupakan deklarasi *foreign key* bernama `ID_PRODI` yang digunakan untuk menghubungkan tabel fakta mahasiswa aktif dengan dimensi program studi.

Baris 13: Merupakan deklarasi untuk menggunakan *table space* bernama `TSOLAP` sebagai tempat penyimpanan data reguler, `TSINDEX` sebagai tempat penyimpanan data index dan `TSLARGE` sebagai tempat penyimpanan data berjenis LOB pada tabel `FACT_MAHASISWA`.

#### 4.3.2 Proses ETL (*Extract, Transform, Load*)

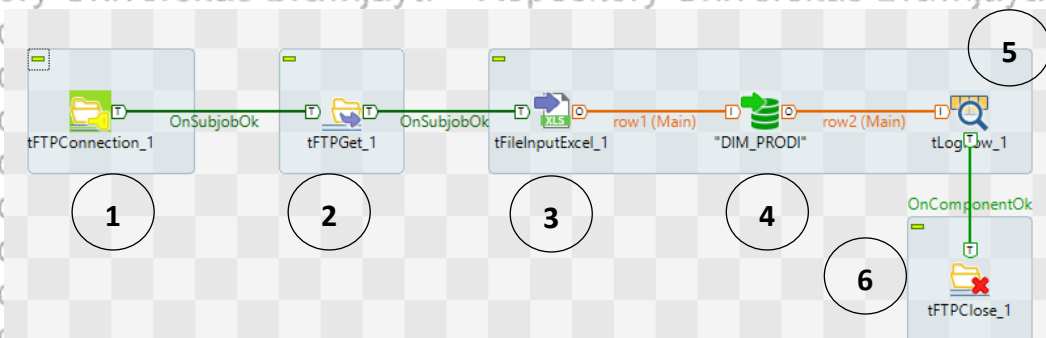
Proses ETL dilakukan berdasarkan perancangan proses ETL yang sebelumnya telah dilakukan. Pada tahap implementasi proses ETL dilakukan pemetaan data

pada setiap dimensi dan fakta sesuai dengan perancangan pada tahap sebelumnya.

#### 4.3.2.1 Implementasi Proses ETL Dimensi Program Studi

Implementasi proses ETL dimensi Program Studi ditunjukkan pada Gambar 4.16. Pada gambar tersebut, sumber data berasal dari *flat file* yang diunggah pada server FTP untuk kemudian dilakukan *input* data menuju tabel target, yaitu tabel DIM\_PRODI. Berikut merupakan komponen yang digunakan pada tahap implementasi proses ETL dimensi Program Studi:

1. tFTPConnection merupakan komponen yang digunakan untuk membuka koneksi menuju server FTP dimana sumber data berada.
2. tFTPGet merupakan komponen yang digunakan untuk memperoleh *flat file* yang akan digunakan sebagai sumber data, yaitu *file* bernama Dimensi.xlsx.
3. tFileInputExcel merupakan komponen yang digunakan untuk memperoleh informasi pada *file* excel yang menjadi sumber data. Pada komponen ini, *sheet* pada *file* excel dimana informasi akan diperoleh adalah *sheet* bernama DIM\_PRODI.
4. tDB2Output merupakan komponen berupa tabel pada basis data dimana data dari *flat file* yang telah diperoleh melalui komponen sebelumnya akan dimuat. Tabel yang digunakan adalah tabel dimensi bernama DIM\_PRODI.
5. tLogRow merupakan komponen yang digunakan untuk menampilkan hasil proses ETL berupa data yang telah berhasil dimuat dalam tabel dimensi.
6. tFTPClose merupakan komponen yang digunakan untuk menutup koneksi dari server FTP.



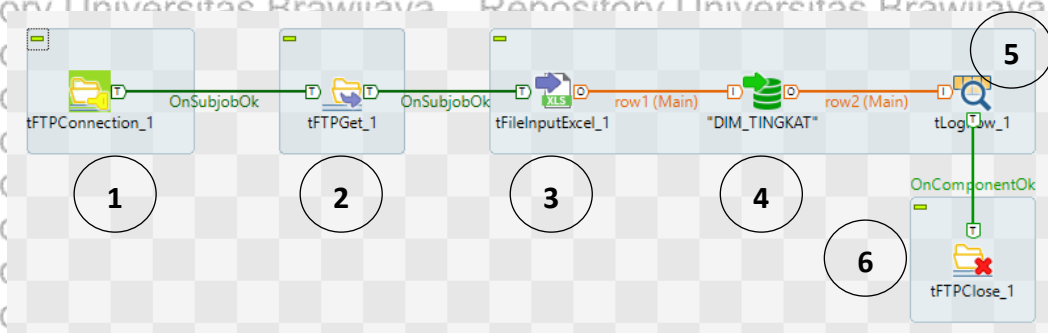
Gambar 4.16 Implementasi proses ETL tabel DIM\_PRODI

#### 4.3.2.2 Implementasi Proses ETL Dimensi Tingkat

Implementasi proses ETL dimensi Tingkat ditunjukkan pada Gambar 4.17. Pada gambar tersebut, sumber data berasal dari *flat file* yang diunggah pada server FTP untuk kemudian dilakukan *input* data menuju tabel target, yaitu tabel DIM\_TINGKAT. Berikut merupakan komponen yang digunakan pada tahap implementasi proses ETL dimensi Tingkat:



1. tFTPConnection merupakan komponen yang digunakan untuk membuka koneksi menuju server FTP dimana sumber data berada.
2. tFTPGet merupakan komponen yang digunakan untuk memperoleh *flat file* yang akan digunakan sebagai sumber data, yaitu *file* bernama Dimensi.xlsx.
3. tFileInputExcel merupakan komponen yang digunakan untuk memperoleh informasi pada *file* excel yang menjadi sumber data. Pada komponen ini, *sheet* pada *file* excel dimana informasi akan diperoleh adalah *sheet* bernama DIM\_TINGKAT.
4. tDB2Output merupakan komponen berupa tabel pada basis data dimana data dari *flat file* yang telah diperoleh melalui komponen sebelumnya akan dimuat. Tabel yang digunakan adalah tabel dimensi bernama DIM\_TINGKAT.
5. tLogRow merupakan komponen yang digunakan untuk menampilkan hasil proses ETL berupa data yang telah berhasil dimuat dalam tabel dimensi.
6. tFTPClose merupakan komponen yang digunakan untuk menutup koneksi dari server FTP.

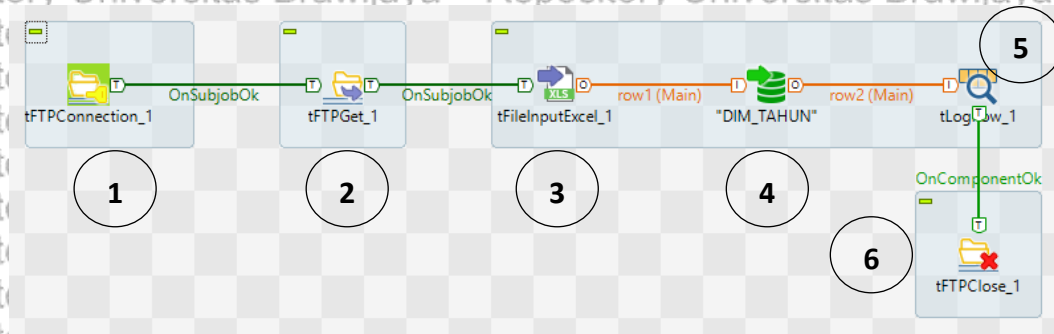


Gambar 4.17 Implementasi proses ETL tabel DIM\_TINGKAT

#### 4.3.2.3 Implementasi Proses ETL Dimensi Tahun

Implementasi proses ETL dimensi Tahun ditunjukkan pada Gambar 4.18. Pada gambar tersebut, sumber data berasal dari *flat file* yang diunggah pada server FTP untuk kemudian dilakukan *input* data menuju tabel target, yaitu tabel DIM\_TAHUN.





**Gambar 4.18 Implementasi proses ETL tabel DIM\_TAHUN**

Adapun komponen yang digunakan pada tahap implementasi proses ETL dimensi Tahun, antara lain:

1. tFTPConnection merupakan komponen yang digunakan untuk membuka koneksi menuju server FTP dimana sumber data berada.
2. tFTPGet merupakan komponen yang digunakan untuk memperoleh *flat file* yang akan digunakan sebagai sumber data, yaitu *file* bernama Dimensi.xls.
3. tFileInputExcel merupakan komponen yang digunakan untuk memperoleh informasi pada *file* excel yang menjadi sumber data. Pada komponen ini, *sheet* pada *file* excel dimana informasi akan diperoleh adalah *sheet* bernama DIM\_TAHUN.
4. tDB2Output merupakan komponen berupa tabel pada basis data dimana data dari *flat file* yang telah diperoleh melalui komponen sebelumnya akan dimuat. Tabel yang digunakan adalah tabel dimensi bernama DIM\_TAHUN.
5. tLogRow merupakan komponen yang digunakan untuk menampilkan hasil proses ETL berupa data yang telah berhasil dimuat dalam tabel dimensi.
6. tFTPClose merupakan komponen yang digunakan untuk menutup koneksi dari server FTP.

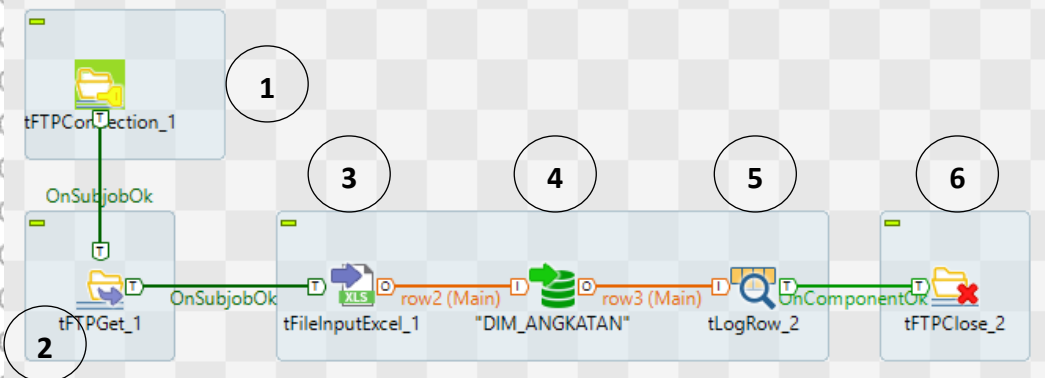
#### 4.3.2.4 Implementasi Proses ETL Dimensi Angkatan

Implementasi proses ETL dimensi Angkatan ditunjukkan pada Gambar 4.19. Pada gambar tersebut, sumber data berasal dari *flat file* yang diunggah pada server FTP untuk kemudian dilakukan *input* data menuju tabel target, yaitu tabel DIM\_ANGKATAN. Berikut merupakan komponen yang digunakan pada tahap implementasi proses ETL dimensi Angkatan:

1. tFTPConnection merupakan komponen yang digunakan untuk membuka koneksi menuju server FTP dimana sumber data berada.
2. tFTPGet merupakan komponen yang digunakan untuk memperoleh *flat file* yang akan digunakan sebagai sumber data, yaitu *file* bernama Dimensi.xls.
3. tFileInputExcel merupakan komponen yang digunakan untuk memperoleh informasi pada *file* excel yang menjadi sumber data. Pada komponen ini, *sheet*

pada *file excel* dimana informasi akan diperoleh adalah *sheet* bernama DIM\_ANGKATAN.

4. tDB2Output merupakan komponen berupa tabel pada basis data dimana data dari *flat file* yang telah diperoleh melalui komponen sebelumnya akan dimuat. Tabel yang digunakan adalah tabel dimensi bernama DIM\_ANGKATAN;
5. tLogRow merupakan komponen yang digunakan untuk menampilkan hasil proses ETL berupa data yang telah berhasil dimuat dalam tabel dimensi
6. tFTPClose merupakan komponen yang digunakan untuk menutup koneksi dari server FTP.



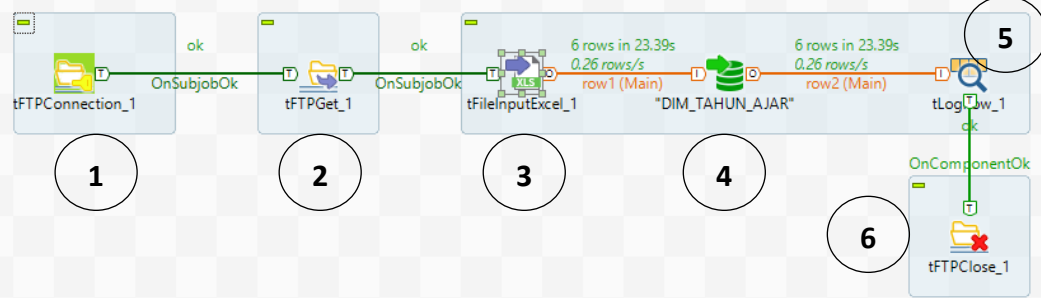
Gambar 4.19 Implementasi proses ETL tabel DIM\_ANGKATAN

#### 4.3.2.5 Implementasi Proses ETL Dimensi Tahun Akademik

Implementasi proses ETL dimensi Tahun Akademik ditunjukkan pada Gambar 4.20. Pada gambar tersebut, sumber data berasal dari *flat file* yang diunggah pada server FTP untuk kemudian dilakukan *input* data menuju tabel target, yaitu tabel DIM\_TAHUN\_AJAR. Adapun komponen yang digunakan pada tahap implementasi proses ETL dimensi Tahun Akademik, antara lain:

1. tFTPConnection merupakan komponen yang digunakan untuk membuka koneksi menuju server FTP dimana sumber data berada.
2. tFTPGet merupakan komponen yang digunakan untuk memperoleh *flat file* yang akan digunakan sebagai sumber data, yaitu *file* bernama Dimensi.xlsx.
3. tFileInputExcel merupakan komponen yang digunakan untuk memperoleh informasi pada *file excel* yang menjadi sumber data. Pada komponen ini, *sheet* pada *file excel* dimana informasi akan diperoleh adalah *sheet* bernama DIM\_TAHUN\_AJAR.
4. tDB2Output merupakan komponen berupa tabel pada basis data dimana data dari *flat file* yang telah diperoleh melalui komponen sebelumnya akan dimuat. Tabel yang digunakan adalah tabel dimensi bernama DIM\_TAHUN\_AJAR.
5. tLogRow merupakan komponen yang digunakan untuk menampilkan hasil proses ETL berupa data yang telah berhasil dimuat dalam tabel dimensi.

6. tFTPClose merupakan komponen yang digunakan untuk menutup koneksi dari server FTP.

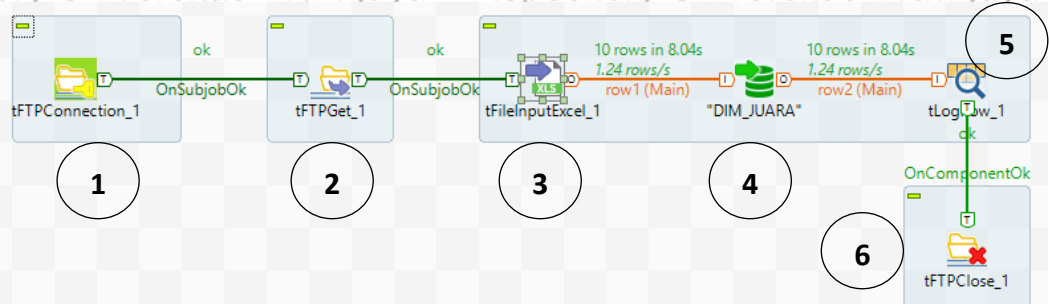


Gambar 4.20 Implementasi proses ETL tabel DIM\_TAHUN\_AJAR

#### 4.3.2.6 Implementasi Proses ETL Dimensi Juara

Implementasi proses ETL dimensi Juara ditunjukkan pada Gambar 4.21. Pada gambar tersebut, sumber data berasal dari *flat file* yang diunggah pada server FTP untuk kemudian dilakukan *input* data menuju tabel target, yaitu tabel DIM\_JUARA. Berikut merupakan komponen yang digunakan pada tahap implementasi proses ETL dimensi Juara:

1. tFTPConnection merupakan komponen yang digunakan untuk membuka koneksi menuju server FTP dimana sumber data berada.
2. tFTPGet merupakan komponen yang digunakan untuk memperoleh *flat file* yang akan digunakan sebagai sumber data, yaitu *file* bernama Dimensi.xlsx.
3. tFileInputExcel merupakan komponen yang digunakan untuk memperoleh informasi pada *file* excel yang menjadi sumber data. Pada komponen ini, *sheet* pada *file* excel dimana informasi akan diperoleh adalah *sheet* bernama DIM\_JUARA.
4. tDB2Output merupakan komponen berupa tabel pada basis data dimana data dari *flat file* yang telah diperoleh melalui komponen sebelumnya akan dimuat. Tabel yang digunakan adalah tabel dimensi bernama DIM\_JUARA.
5. tLogRow merupakan komponen yang digunakan untuk menampilkan hasil proses ETL berupa data yang telah berhasil dimuat dalam tabel dimensi.
6. tFTPClose merupakan komponen yang digunakan untuk menutup koneksi dari server FTP.



Gambar 4.21 Implementasi proses ETL tabel DIM\_JUARA

4.3.2.7 Implementasi Proses ETL Fakta Prestasi

Implementasi proses ETL fakta Prestasi ditunjukkan pada Gambar 4.22. Pada gambar tersebut, sumber data berasal dari flat file yang diunggah pada server FTP untuk kemudian dilakukan input data menuju tabel target, yaitu tabel FACT\_PRESTASI. Sebelum data dimuat ke dalam tabel target, data terlebih dahulu dilakukan proses agregasi untuk kemudian dijadikan fakta pada tabel FACT\_PRESTASI. Adapun komponen yang digunakan untuk melakukan proses ETL fakta Prestasi ditunjukkan pada Tabel 4.22.

Tabel 4.22 Komponen pada implementasi proses ETL fakta Prestasi

Kode Komponen	Nama Komponen	Deskripsi
1	tFTPConnection	Komponen yang digunakan untuk membuka koneksi menuju server FTP dimana sumber data berada.
2	tFTPGet	Komponen yang digunakan untuk memperoleh flat file yang akan digunakan sebagai sumber data, yaitu file bernama Kemahasiswaan.xlsx.
3	tFileInputExcel	Komponen yang digunakan untuk memperoleh informasi pada file excel yang menjadi sumber data. Pada komponen ini, sheet pada file excel dimana informasi akan diperoleh adalah sheet bernama Prestasi.
4	tMap	Komponen yang berfungsi untuk melakukan pemetaan data dari kolom-kolom pada tabel input menuju kolom-kolom pada tabel output. Pada pemetaan data pertama yang ditunjukkan pada Gambar 4.23, kolom-kolom pada tabel input yang berasal dari flat file dipetakan menuju kolom-

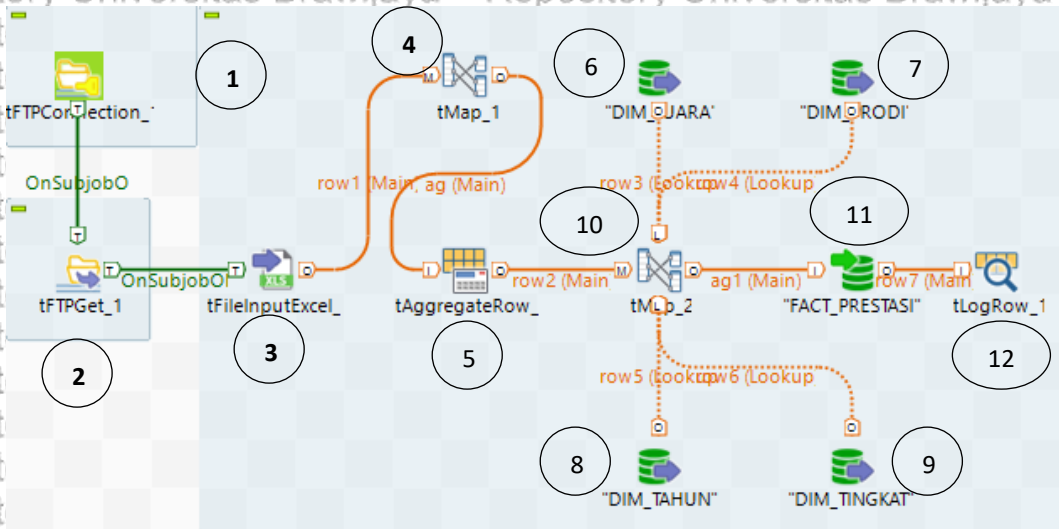


Tabel 4.22 Komponen pada implementasi proses ETL fakta Prestasi

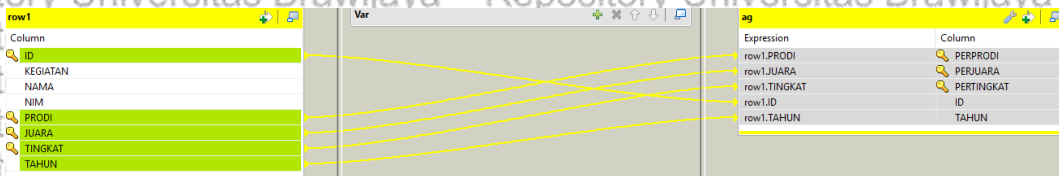
Kode Komponen	Nama Komponen	Deskripsi
5	tAggregateRow	kolom pada tabel output dimana proses agregasi data akan dilakukan. Komponen yang digunakan untuk melakukan operasi agregasi data dari tabel input yang berasal dari tabel output pada komponen tMap. Pada komponen ini, dilakukan <i>grouping</i> berdasarkan dimensi yang digunakan pada tabel fakta Prestasi kemudian dilakukan operasi agregasi dengan menghitung jumlah prestasi seperti yang ditunjukkan pada Gambar 4.24.
6, 7, 8, 9	tDB2Input	Komponen tabel pada basis data dimana data pada tabel-tabel tersebut akan dibaca sebagai <i>input</i> sebelum dipetakan menuju tabel <i>output</i> pada komponen tMap. Tabel-tabel yang digunakan adalah tabel dimensi bernama DIM_JUARA, DIM_PRODI, DIM_TAHUN DAN DIM_TINGKAT.
10	tMap	Komponen yang berfungsi untuk melakukan pemetaan data dari kolom-kolom pada tabel <i>input</i> yang merupakan tabel <i>output</i> pada proses agregasi dan tabel dimensi yang dilakukan <i>lookup</i> menuju kolom-kolom pada tabel <i>output</i> , yaitu tabel FACT_PRESTASI. Pada pemetaan kedua yang ditunjukkan pada Gambar 4.25, dilakukan <i>join</i> pada konten dari tabel hasil agregasi dengan konten dari setiap tabel dimensi yang digunakan sebelum dipetakan menuju tabel FACT_PRESTASI.
11	tDB2Output	Komponen berupa tabel pada basis data dimana data hasil agregasi dan <i>foreign key</i> dari tabel-tabel dimensi yang telah dipetakan pada komponen sebelumnya akan dimuat. Tabel yang

Tabel 4.22 Komponen pada implementasi proses ETL fakta Prestasi

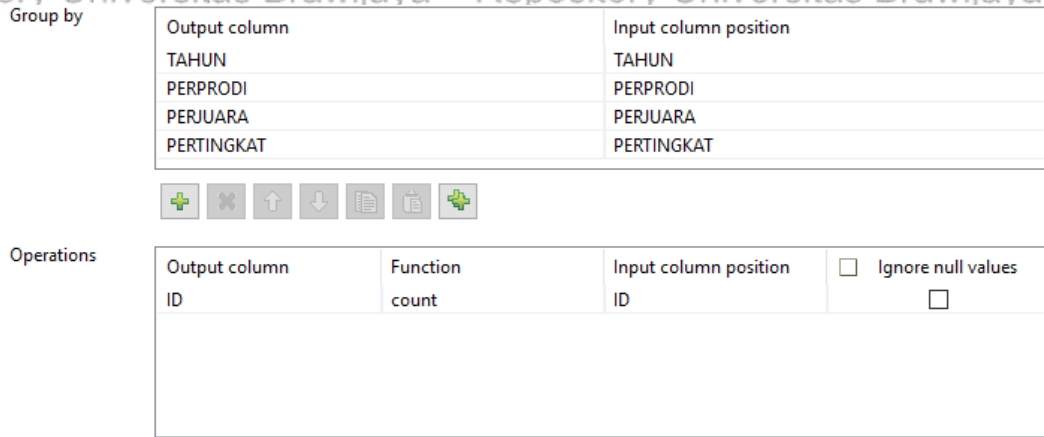
Kode Komponen	Nama Komponen	Deskripsi
12	tLogRow	digunakan adalah tabel fakta FACT_PRESTASI. Komponen yang digunakan untuk menampilkan hasil proses ETL berupa data yang telah berhasil dimuat dalam tabel fakta FACT_PRESTASI.



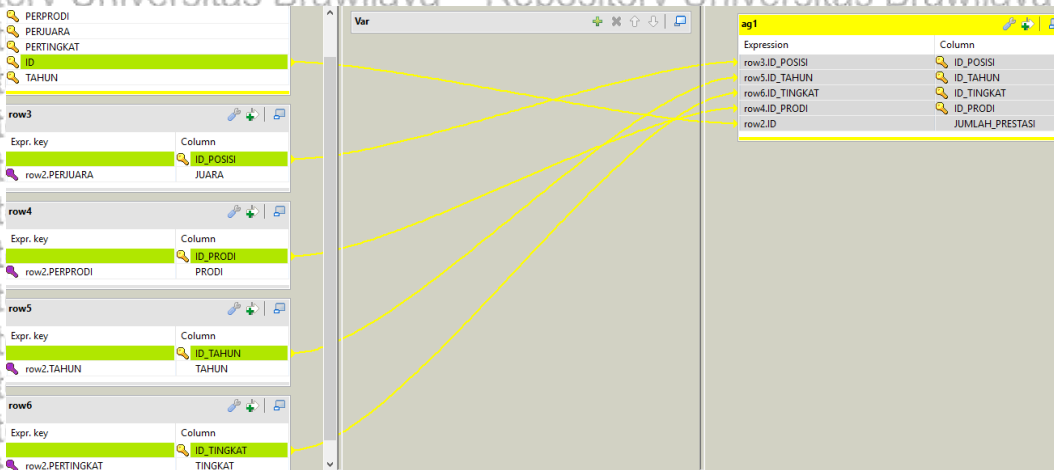
Gambar 4.22 Implementasi proses ETL fakta Prestasi



Gambar 4.23 Pemetaan data pertama pada fakta Prestasi



Gambar 4.24 Agregasi pada fakta Prestasi



Gambar 4.25 Pemetaan data kedua pada fakta Prestasi

#### 4.3.2.8 Implementasi Proses ETL Fakta Lulusan

Implementasi proses ETL fakta Lulusan ditunjukkan pada Gambar 4.26. Pada gambar tersebut, sumber data berasal dari *flat file* yang diunggah pada server FTP untuk kemudian dilakukan *input* data menuju tabel target, yaitu tabel FACT\_AKADEMIK. Sebelum data dimuat ke dalam tabel target, data terlebih dahulu dilakukan proses agregasi untuk kemudian dijadikan fakta pada tabel FACT\_AKADEMIK. Adapun komponen yang digunakan untuk melakukan proses ETL fakta Lulusan dapat dilihat pada Tabel 4.23.

Tabel 4.23 Komponen pada implementasi proses ETL fakta Lulusan

Kode Komponen	Nama Komponen	Deskripsi
1	tFTPConnection	Komponen yang digunakan untuk membuka koneksi menuju server FTP dimana sumber data berada.



Tabel 4.23. Komponen pada implementasi proses ETL fakta Lulusan

Kode Komponen	Nama Komponen	Deskripsi
2	tFTPGet	Komponen yang digunakan untuk memperoleh <i>flat file</i> yang akan digunakan sebagai sumber data, yaitu <i>file</i> bernama Akademik.xlsx.
3	tFileInputExcel	Komponen yang digunakan untuk memperoleh informasi pada <i>file excel</i> yang menjadi sumber data. Pada komponen ini, <i>sheet</i> pada <i>file excel</i> dimana informasi akan diperoleh adalah <i>sheet</i> bernama Lulusan.
4	tMap	Komponen yang berfungsi untuk melakukan pemetaan data dari kolom-kolom pada tabel <i>input</i> menuju kolom-kolom pada tabel <i>output</i> . Pada pemetaan data pertama yang ditunjukkan pada Gambar 4.27, kolom-kolom pada tabel input yang berasal dari <i>flat file</i> dipetakan menuju kolom-kolom pada tabel output dimana proses agregasi data akan dilakukan.
5	tAggregateRow	Komponen yang digunakan untuk melakukan operasi agregasi data dari tabel input yang berasal dari tabel output pada komponen tMap. Pada komponen ini, dilakukan <i>grouping</i> berdasarkan dimensi yang digunakan pada tabel fakta Akademik kemudian dilakukan operasi agregasi dengan menghitung jumlah lulusan, serta operasi untuk mendapatkan nilai IPK terendah, rata-rata dan tertinggi seperti yang ditunjukkan pada Gambar 4.28.
6, 7, 8	tDB2Input	Komponen tabel pada basis data dimana data pada tabel-tabel tersebut akan dibaca sebagai <i>input</i> sebelum dipetakan menuju tabel <i>output</i> pada komponen tMap. Tabel-tabel yang digunakan adalah tabel dimensi





Tabel 4.23. Komponen pada implementasi proses ETL fakta Lulusan

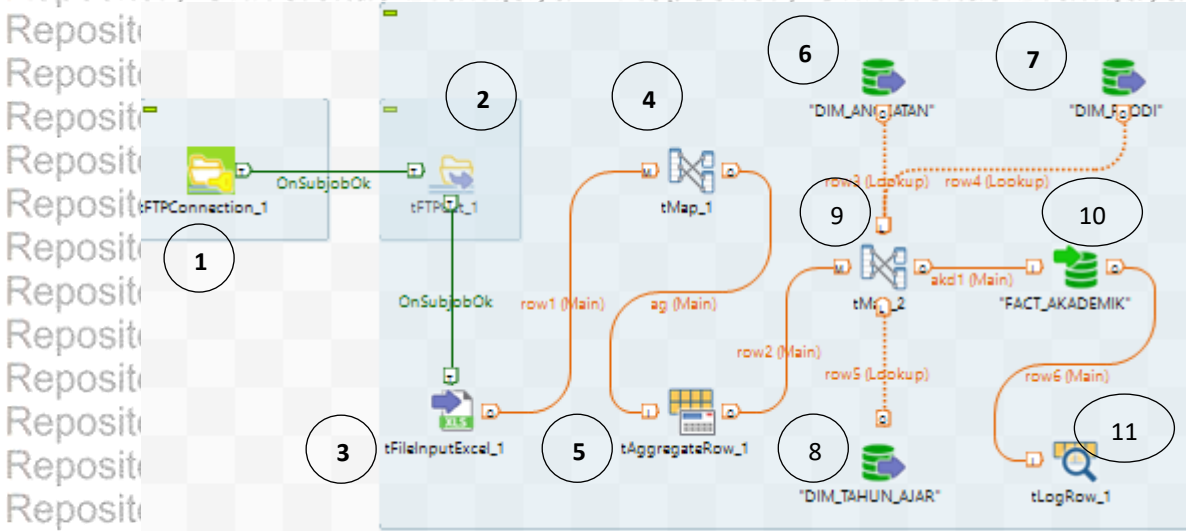
Kode Komponen	Nama Komponen	Deskripsi
9	tMap	bernama DIM_ANGKATAN, DIM_PRODI dan DIM_TAHUN_AJAR. Komponen yang berfungsi untuk melakukan pemetaan data dari kolom-kolom pada tabel <i>input</i> yang merupakan tabel <i>output</i> pada proses agregasi dan tabel dimensi yang dilakukan <i>lookup</i> menuju kolom-kolom pada tabel <i>output</i> , yaitu tabel FACT_AKADEMIK. Pada pemetaan kedua yang ditunjukkan pada Gambar 4.29, dilakukan <i>join</i> pada konten dari tabel hasil agregasi dengan konten dari setiap tabel dimensi yang digunakan sebelum dipetakan menuju tabel FACT_AKADEMIK.
10	tDB2Output	Komponen berupa tabel pada basis data dimana data hasil agregasi dan <i>foreign key</i> dari tabel-tabel dimensi yang telah dipetakan pada komponen sebelumnya akan dimuat. Tabel yang digunakan adalah tabel fakta FACT_AKADEMIK.
11	tLogRow	Komponen yang digunakan untuk menampilkan hasil proses ETL berupa data yang telah berhasil dimuat dalam tabel fakta FACT_AKADEMIK.



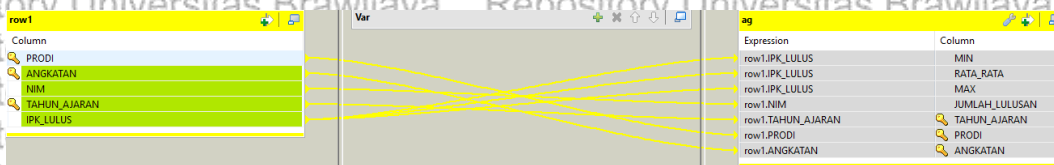
Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository  
 Repository  
 Repository  
 Repository  
 Repository  
 Repository



Gambar 4.26 Implementasi proses ETL fakta Lulusan



Gambar 4.27 Pemetaan data pertama pada fakta Lulusan

Group by		Output column	Input column position
		TAHUN_AJARAN	TAHUN_AJARAN
		PRODI	PRODI
		ANGKATAN	ANGKATAN

Operations			
Output column	Function	Input column position	<input type="checkbox"/> Ignore null values
MIN	min	MIN	<input type="checkbox"/>
RATA_RATA	avg	RATA_RATA	<input type="checkbox"/>
MAX	max	MAX	<input type="checkbox"/>
JUMLAH_LULUSAN	count	JUMLAH_LULUSAN	<input type="checkbox"/>

Gambar 4.28 Agregasi pada fakta Lulusan

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository  
 Repository  
 Repository  
 Repository  
 Repository  
 Repository

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

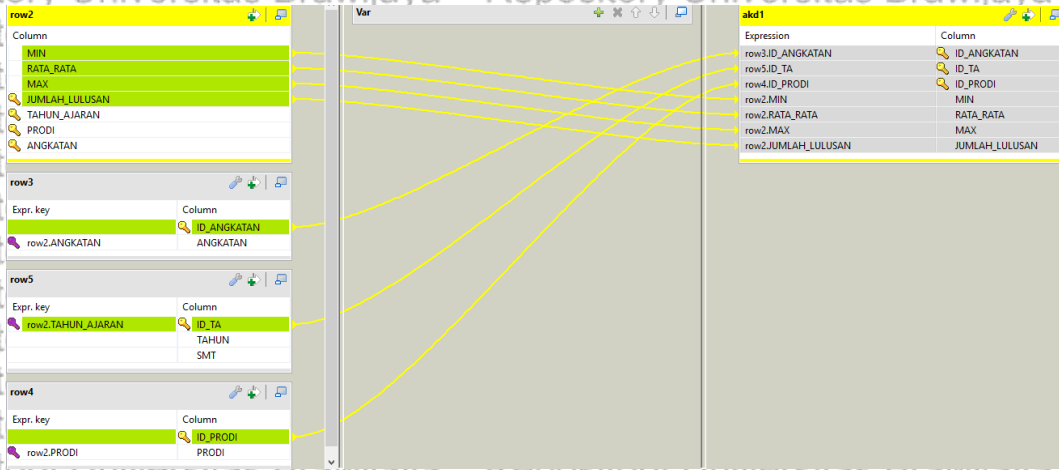
Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository  
 Repository  
 Repository  
 Repository  
 Repository  
 Repository

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository  
 Repository  
 Repository  
 Repository  
 Repository  
 Repository



Gambar 4.29 Pemetaan data kedua pada fakta Lulusan

#### 4.3.2.9 Implementasi Proses ETL Fakta Mahasiswa Aktif

Implementasi proses ETL fakta Mahasiswa ditunjukkan pada Gambar 4.30. Pada gambar tersebut, sumber data berasal dari *flat file* yang diunggah pada server FTP untuk kemudian dilakukan *input* data menuju tabel target, yaitu tabel FACT\_MAHASISWA. Sebelum data dimuat ke dalam tabel target, data terlebih dahulu dilakukan proses agregasi untuk kemudian dijadikan fakta pada tabel FACT\_MAHASISWA. Tabel 4.24 menunjukkan komponen yang digunakan untuk melakukan proses ETL fakta Mahasiswa Aktif.

Tabel 4.24 Komponen pada implementasi proses ETL fakta Mahasiswa Aktif

Kode Komponen	Nama Komponen	Deskripsi
1	tFTPConnection	Komponen yang digunakan untuk membuka koneksi menuju server FTP dimana sumber data berada.
2	tFTPGet	Komponen yang digunakan untuk memperoleh <i>flat file</i> yang akan digunakan sebagai sumber data, yaitu <i>file</i> bernama Akademik.xlsx.
3	tFileInputExcel	Komponen yang digunakan untuk memperoleh informasi pada <i>file</i> excel yang menjadi sumber data. Pada komponen ini, <i>sheet</i> pada <i>file</i> excel dimana informasi akan diperoleh adalah <i>sheet</i> bernama Aktif.
4	tMap	Komponen yang berfungsi untuk melakukan pemetaan data dari kolom-kolom pada tabel <i>input</i> menuju kolom-kolom pada tabel <i>output</i> . Pada pemetaan data pertama yang



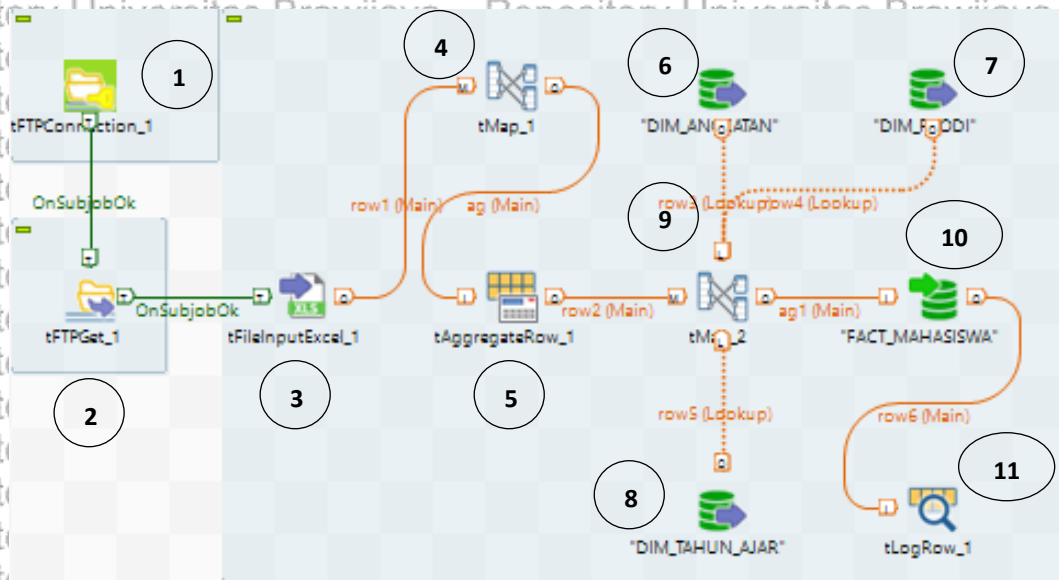
Tabel 4.24 Komponen pada implementasi proses ETL fakta Mahasiswa Aktif

Kode Komponen	Nama Komponen	Deskripsi
		ditunjukkan pada Gambar 4.31, kolom-kolom pada tabel input yang berasal dari <i>flat file</i> dipetakan menuju kolom-kolom pada tabel output dimana proses agregasi data akan dilakukan.
5	tAggregateRow	Komponen yang digunakan untuk melakukan operasi agregasi data dari tabel input yang berasal dari tabel output pada komponen tMap. Pada komponen ini, dilakukan <i>grouping</i> berdasarkan dimensi yang digunakan pada tabel fakta Mahasiswa Aktif kemudian dilakukan operasi agregasi dengan menghitung jumlah mahasiswa aktif, serta operasi untuk mendapatkan nilai IPK terendah, rata-rata dan tertinggi seperti yang ditunjukkan pada Gambar 4.32.
6, 7, 8	tDB2Input	Komponen tabel pada basis data dimana data pada tabel-tabel tersebut akan dibaca sebagai <i>input</i> sebelum dipetakan menuju tabel <i>output</i> pada komponen tMap. Tabel-tabel yang digunakan adalah tabel dimensi bernama DIM_ANGKATAN, DIM_TAHUN_AJAR dan DIM_PRODI.
9	tMap	Komponen yang berfungsi untuk melakukan pemetaan data dari kolom-kolom pada tabel <i>input</i> yang merupakan tabel <i>output</i> pada proses agregasi dan tabel dimensi yang dilakukan <i>lookup</i> menuju kolom-kolom pada tabel <i>output</i> , yaitu tabel FACT_MAHASISWA. Pada pemetaan kedua yang ditunjukkan pada Gambar 4.33, dilakukan <i>join</i> pada konten dari tabel hasil agregasi dengan konten dari setiap tabel dimensi yang digunakan sebelum dipetakan menuju tabel FACT_MAHASISWA.

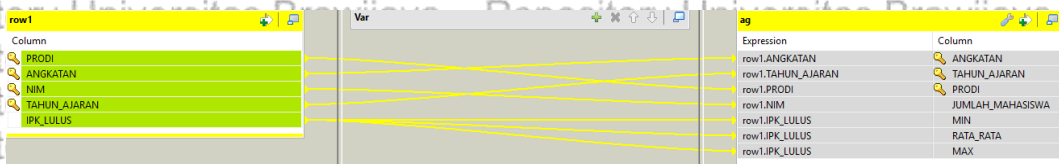


Tabel 4.24 Komponen pada implementasi proses ETL fakta Mahasiswa Aktif

Kode Komponen	Nama Komponen	Deskripsi
10	tDB2Output	Komponen berupa tabel pada basis data dimana data hasil agregasi dan <i>foreign key</i> dari tabel-tabel dimensi yang telah dipetakan pada komponen sebelumnya akan dimuat. Tabel yang digunakan adalah tabel fakta FACT_MAHASISWA.
11	tLogRow	Komponen yang digunakan untuk menampilkan hasil proses ETL berupa data yang telah berhasil dimuat dalam tabel fakta FACT_MAHASISWA.



Gambar 4.30 Implementasi proses ETL fakta Mahasiswa Aktif



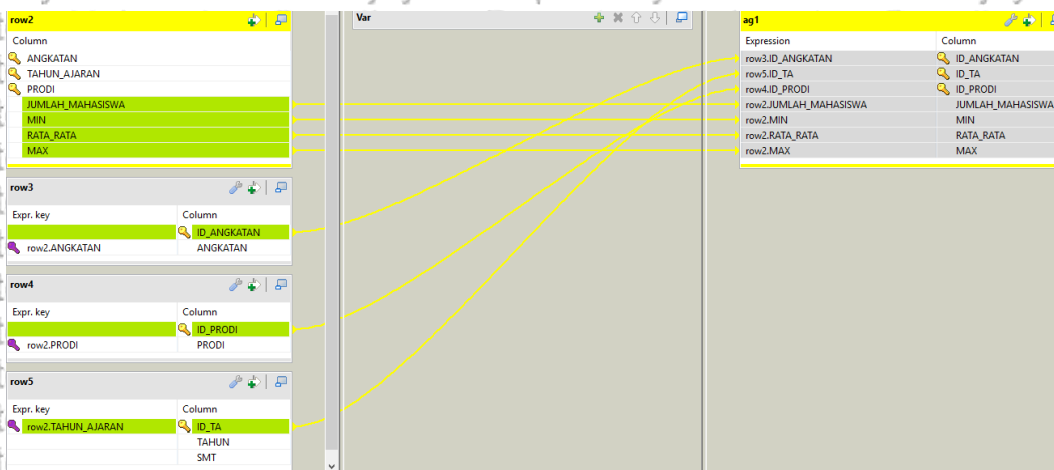
Gambar 4.31 Pemetaan data pertama pada fakta Mahasiswa Aktif

Group by	Output column	Input column position
	ANGKATAN	ANGKATAN
	TAHUN_AJARAN	TAHUN_AJARAN
	PRODI	PRODI

Operations	Output column	Function	Input column position	<input type="checkbox"/> Ignore null values
	JUMLAH_MAHASISWA	count	JUMLAH_MAHASISWA	<input checked="" type="checkbox"/>
	MIN	min	MIN	<input checked="" type="checkbox"/>
	RATA_RATA	avg	RATA_RATA	<input checked="" type="checkbox"/>
	MAX	max	MAX	<input checked="" type="checkbox"/>

Gambar 4.32 Agregasi pada fakta Mahasiswa Aktif



Gambar 4.33 Pemetaan data kedua pada fakta Mahasiswa Aktif

#### 4.3.2.10 Penjadwalan

Penjadwalan (*scheduling*) pada pengembangan *data warehouse* dilakukan untuk melakukan eksekusi terhadap alur proses ETL yang telah dibangun. Pada tahap ini, waktu eksekusi untuk alur proses ETL diatur. Adapun waktu eksekusi yang dipilih untuk penjadwalan proses ETL pada fakta Prestasi ditunjukkan pada Gambar 4.34 dengan pengaturan, yaitu:

1. Interval dijalankannya proses ETL dilakukan secara bulanan.
2. Eksekusi proses ETL dilakukan setiap hari Jumat minggu pertama dan terakhir bulan Januari dan Desember.

Pada fakta Mahasiswa Aktif, waktu eksekusi yang dipilih dapat dilihat pada Gambar 4.35. Adapun pengaturan yang dilakukan untuk melakukan eksekusi proses ETL pada fakta tersebut, antara lain:

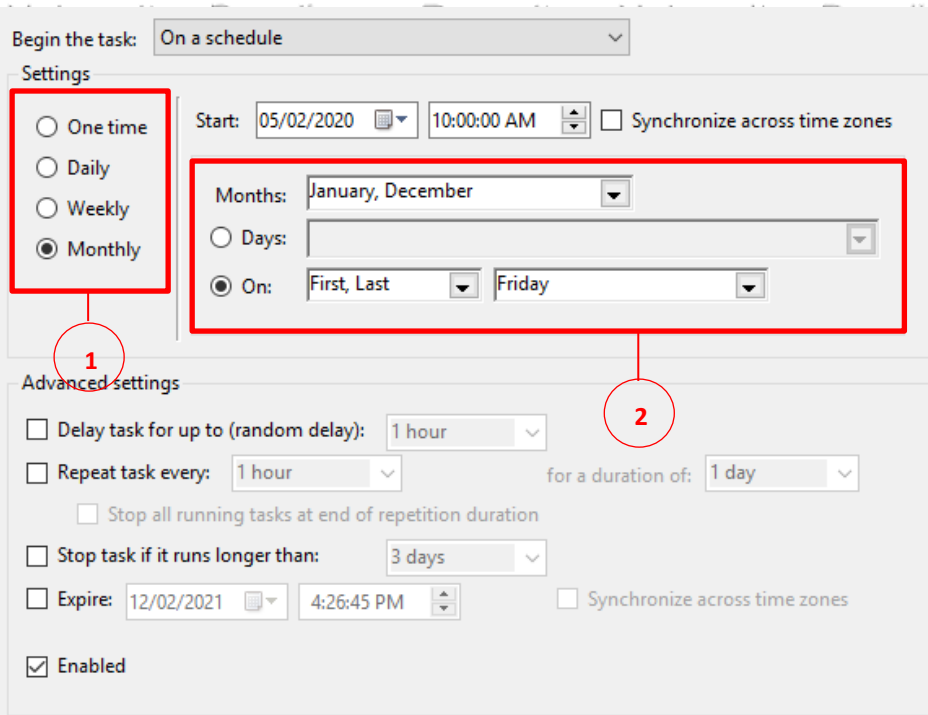
1. Interval dijalankannya proses ETL dilakukan secara bulanan.



2. Eksekusi proses ETL dilakukan pada akhir periode akademik, yaitu setiap hari Jumat minggu kedua dan terakhir bulan Januari, Agustus, September dan Desember.

Pada fakta Lulusan, waktu eksekusi yang dipilih untuk penjadwalan proses ETL ditunjukkan pada Gambar 4.36. Berikut pengaturan yang dilakukan untuk melakukan eksekusi proses ETL pada fakta tersebut:

1. Interval dijalankannya proses ETL dilakukan secara bulanan.
2. Eksekusi proses ETL dilakukan setiap bulan, yaitu setiap hari Senin minggu ketiga dan terakhir bulan Januari, Februari, Maret, April, Mei, Juni, Juli, Agustus, September, Oktober, November dan Desember. Dipilihnya waktu eksekusi tersebut menyesuaikan dengan kalender wisuda Universitas Brawijaya.



**Gambar 4.34 Waktu eksekusi yang dipilih untuk penjadwalan proses ETL fakta Prestasi**



Begin the task: On a schedule

Settings

One time  
 Daily  
 Weekly  
 Monthly

Start: 12/02/2020 10:00:00 PM  Synchronize across time zones

Months: January, August, Septemb...

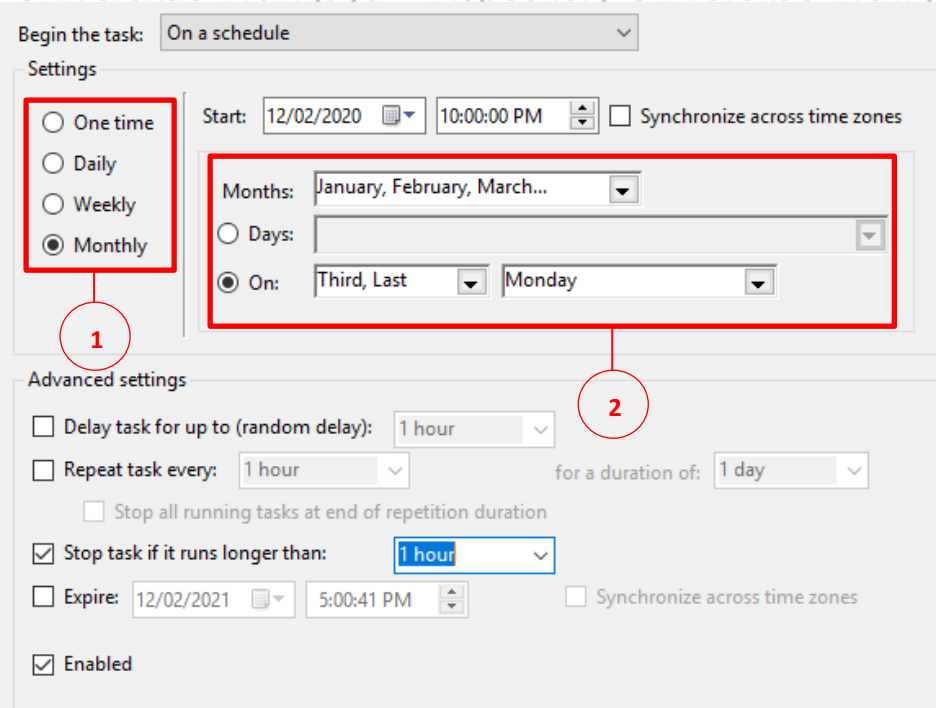
Days:  
 On: Second, Last Friday

Advanced settings

Delay task for up to (random delay): 1 hour  
 Repeat task every: 1 hour for a duration of: 1 day  
 Stop all running tasks at end of repetition duration  
 Stop task if it runs longer than: 1 hour  
 Expire: 12/02/2021 4:34:54 PM  Synchronize across time zones  
 Enabled

Gambar 4.35 Waktu eksekusi yang dipilih untuk penjadwalan proses ETL fakta Mahasiswa Aktif





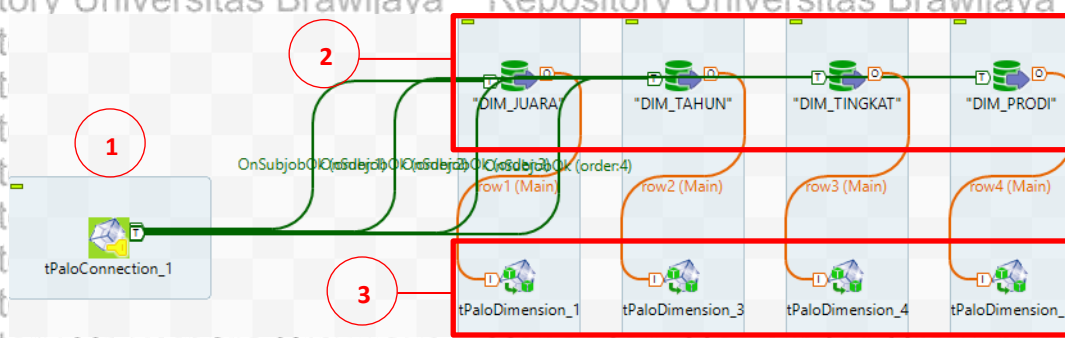
Gambar 4.36 Waktu eksekusi yang dipilih untuk penjadwalan proses ETL fakta Lulusan

### 4.3.3 Implementasi OLAP

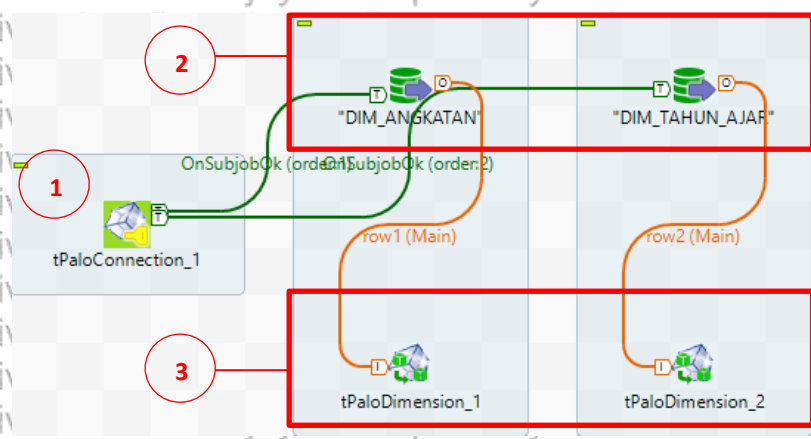
#### 4.3.3.1 Pembuatan Dimensi

Tahap pertama dalam mengimplementasikan OLAP (*Online Analytical Processing*) adalah dengan membuat dimensi yang akan digunakan untuk masing-masing fakta. Gambar 4.37 menunjukkan alur pembuatan dimensi pada OLAP untuk fakta Prestasi. Terdapat tiga komponen yang digunakan, yaitu *tPaloConnection*, *tDB2Input* dan *tPaloDimension*. Komponen *tPaloConnection* berfungsi untuk membuat koneksi dengan server Jedox dimana dimensi dan *cube* akan disimpan. Komponen *tDB2Input* digunakan untuk mengirim skema tabel dimensi pada basis data yang telah dibuat sebelumnya sebagai acuan dalam pembuatan dimensi pada server Jedox. Pada komponen *tDB2Input* digunakan empat tabel, yaitu tabel DIM\_JUARA, DIM\_PRODI, DIM\_TAHUN dan DIM\_TINGKAT. Komponen *tPaloDimension* digunakan untuk membuat dimensi pada server Jedox sesuai dengan skema yang didapatkan dari komponen *tDB2Input*.

Alur pembuatan dimensi pada OLAP untuk fakta Lulusan dan Mahasiswa Aktif ditunjukkan pada Gambar 4.38. Adapun komponen yang digunakan sama dengan komponen pada Gambar 4.37 namun dengan tabel dimensi yang berbeda pada komponen *tDB2Input*. Tabel dimensi yang digunakan, yaitu tabel DIM\_ANGKATAN dan DIM\_TAHUN\_AJAR.



**Gambar 4.37 Pembuatan dimensi pada OLAP untuk fakta Prestasi**



**Gambar 4.38 Pembuatan dimensi pada OLAP untuk fakta Lulusan dan Mahasiswa Aktif**

**4.3.3.2 Pembuatan Cube**

Setelah dimensi yang dibutuhkan berhasil dibuat, maka tahap selanjutnya adalah membuat *cube*. Terdapat dua komponen yang digunakan untuk membuat *cube*, yaitu komponen *tPaloConnection* dan *tPaloCube* seperti yang ditunjukkan pada Gambar 4.39. Komponen *tPaloConnection* digunakan untuk membuat koneksi dengan *server* Jedox, sedangkan komponen *tPaloCube* digunakan untuk menjalankan operasi pada *cube*, baik membuat *cube* maupun menghapus *cube*.

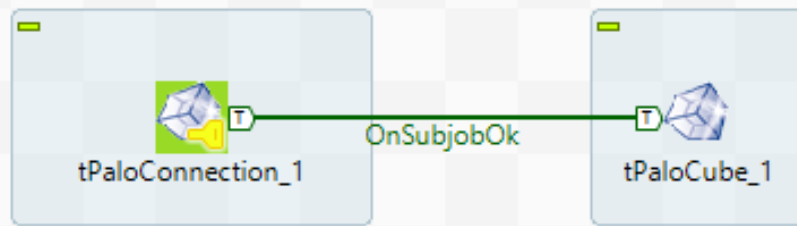
Dalam proses pembuatan *cube*, dilakukan pengaturan pada komponen *tPaloCube* seperti yang dapat dilihat pada Gambar 4.40, Gambar 4.41 dan Gambar 4.42. Adapun pengaturan yang dilakukan ditunjukkan pada Tabel 4.25.

**Tabel 4.25 Pengaturan pada komponen tPaloCube**

Kode Komponen	Nama Komponen	Deskripsi
1	Database	Berisi informasi mengenai basis data pada server Jedox dimana <i>cube</i> akan dibuat. Basis data yang digunakan untuk membuat <i>cube</i> pada penelitian ini bernama OLAP.

Tabel 4.25 Pengaturan pada komponen tPaloCube

Kode Komponen	Nama Komponen	Deskripsi
2	Cube	Berisi informasi mengenai nama <i>cube</i> yang akan dibuat.
3	Cube type	Tipe <i>cube</i> yang akan dibuat.
4	Action on cube	Berisi pilihan operasi yang akan dijalankan pada <i>cube</i> , yaitu operasi membuat atau menghapus <i>cube</i> . Operasi yang dijalankan pada penelitian ini adalah membuat <i>cube</i> .
5	Dimension list	Berisi informasi mengenai dimensi-dimensi yang akan digunakan pada sebuah <i>cube</i> .



Gambar 4.39 Komponen untuk membuat *cube*

Database 1 "OLAP"

Cube 2 "Prestasi16-19"

Cube type 3 Normal Action on cube Create cube if does not exist 4

Dimension list 5

Dimension	"DIM_JUARA"
	"DIM_TAHUN"
	"DIM_TINGKAT"
	"DIM_PRODI"

Gambar 4.40 Pengaturan untuk membuat *cube* Prestasi



The screenshot shows the configuration for a cube named "Lulusan". The database is set to "OLAP". The cube type is "Normal" and the action on cube is "Create cube". The dimension list includes "DIM\_ANGKATAN", "DIM\_TAHUN\_AJAR", and "DIM\_PRODI". A red box highlights the dimension list, and a red line labeled "5" points to it.

Gambar 4.41 Pengaturan untuk membuat *cube* Lulusan

The screenshot shows the configuration for a cube named "Mahasiswa". The database is set to "OLAP". The cube type is "Normal" and the action on cube is "Create cube". The dimension list includes "DIM\_ANGKATAN", "DIM\_TAHUN\_AJAR", and "DIM\_PRODI". A red box highlights the dimension list, and a red line labeled "5" points to it.

Gambar 4.42 Pengaturan untuk membuat *cube* Mahasiswa

### 4.3.3.3 Penulisan Data pada *Cube*

*Cube* yang telah berhasil dibuat akan diisi dengan data yang telah tersimpan pada tabel fakta terkait pada basis data. Terdapat enam komponen yang digunakan untuk menulis data dari tabel fakta menuju *cube* seperti yang ditunjukkan pada Gambar 4.43, Gambar 4.47 dan Gambar 4.51. Tabel 4.26 menunjukkan deskripsi setiap komponen yang digunakan pada proses penulisan data pada *cube*.

Tabel 4.26 Komponen untuk proses penulisan data pada *cube*

Kode Komponen	Nama Komponen	Deskripsi
1, 2, 3, 4, 5a	tDB2Input	Komponen tabel pada basis data dimana data pada tabel tersebut merupakan data yang akan ditulis pada <i>cube</i> .
5	tMap	Komponen yang berfungsi untuk melakukan pemetaan data dari kolom-kolom pada tabel <i>input</i> menuju kolom-kolom pada tabel <i>output</i> . Pada pemetaan data yang ditunjukkan pada Gambar 4.44, Gambar 4.48 dan Gambar 4.52, kolom-kolom pada tabel <i>input</i> yang berasal dari tabel fakta dan dimensi

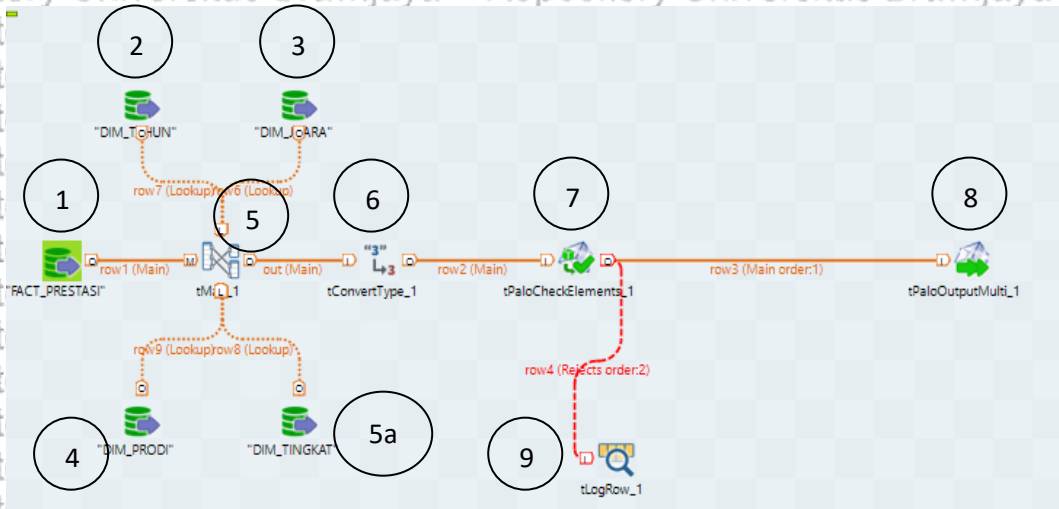


Tabel 4.26 Komponen untuk proses penulisan data pada cube

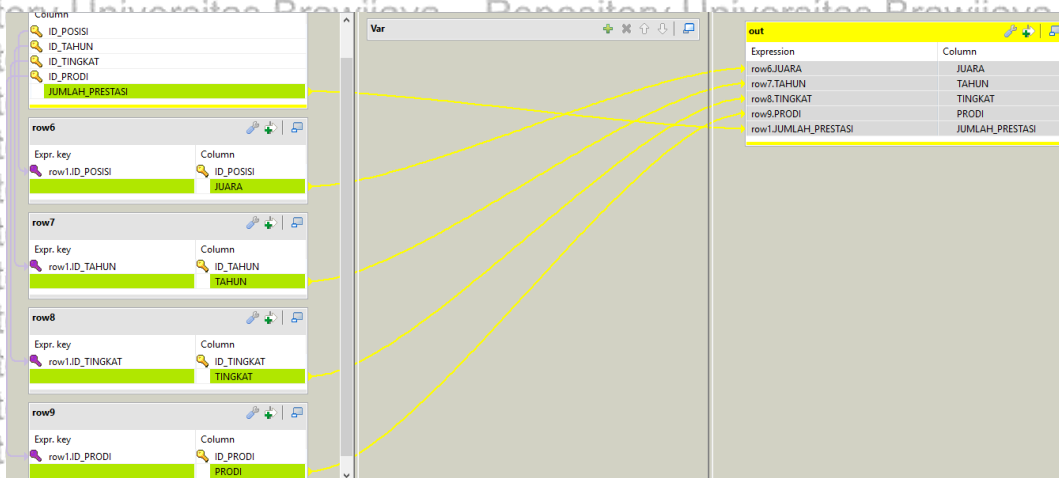
Kode Komponen	Nama Komponen	Deskripsi
6	tConvertType	Komponen yang digunakan untuk melakukan konversi tipe data pada tabel <i>input</i> untuk menyesuaikan tipe data pada <i>cube</i> dimana data akan ditulis. Tipe data yang digunakan untuk <i>primary key</i> setiap dimensi adalah String, sedangkan tipe data yang digunakan untuk fakta adalah Double seperti yang ditunjukkan pada Gambar 4.45, Gambar 4.49 dan Gambar 4.53.
7	tPaloCheckElements	Komponen yang digunakan untuk memeriksa elemen pada tabel fakta sebelum dimuat ke dalam <i>cube</i> . Pemeriksaan ini dilakukan dengan memeriksa setiap elemen yang terdapat pada masing-masing dimensi dengan elemen yang didefinisikan pada komponen ini. Jika elemen tersebut sesuai dengan elemen yang didefinisikan, maka data akan diteruskan untuk ditulis pada <i>cube</i> oleh komponen nomor 8. Namun jika elemen tidak sesuai, maka data akan ditolak menuju komponen nomor 9. Pengaturan pada komponen ini ditunjukkan pada Gambar 4.46, Gambar 4.50 dan Gambar 4.54.
8	tPaloOutputMulti	Komponen yang digunakan untuk menuliskan data dari tabel <i>input</i> menuju <i>cube</i> yang dituju.
9	tLogRow	Komponen yang digunakan untuk menampilkan <i>error</i> akibat ketidaksesuaian antara elemen pada

Tabel 4.26 Komponen untuk proses penulisan data pada cube

Kode Komponen	Nama Komponen	Deskripsi
		tabel fakta dengan elemen pada cube.



Gambar 4.43 Penulisan data pada cube Prestasi



Gambar 4.44 Pemetaan data dari data warehouse menuju OLAP



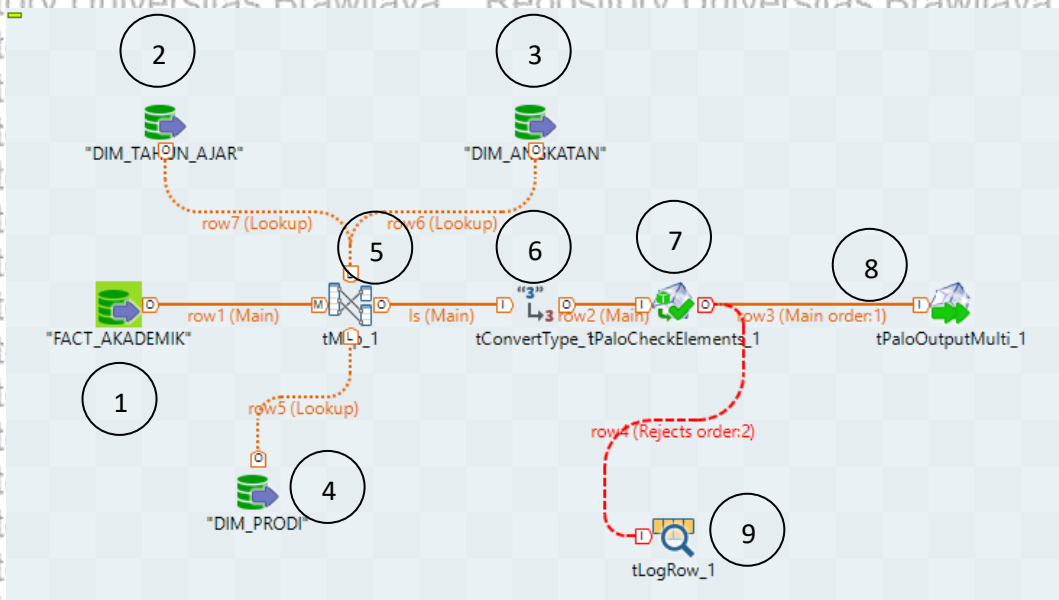
out (Input)										tConvertType_1 (Output)									
Column	Key	Type	<input checked="" type="checkbox"/>	N...	Date Pa...	Len...	Pre...	D...	Co...	Column	Key	Type	<input checked="" type="checkbox"/>	N...	Date Pa...	Len...	Pre...	D...	Co...
JUARA	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			45	0			JUARA	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			45	0		
TAHUN	<input type="checkbox"/>	Int...	<input checked="" type="checkbox"/>			45	0			TAHUN	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			45	0		
TINGKAT	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			45	0			TINGKAT	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			45	0		
PRODI	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			45	0			PRODI	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			45	0		
JUMLAH_P...	<input type="checkbox"/>	Int...	<input checked="" type="checkbox"/>			10	0			JUMLAH_P...	<input type="checkbox"/>	Do...	<input checked="" type="checkbox"/>			10	0		

Gambar 4.45 Konversi tipe data sebelum data ditulis pada *cube* Prestasi

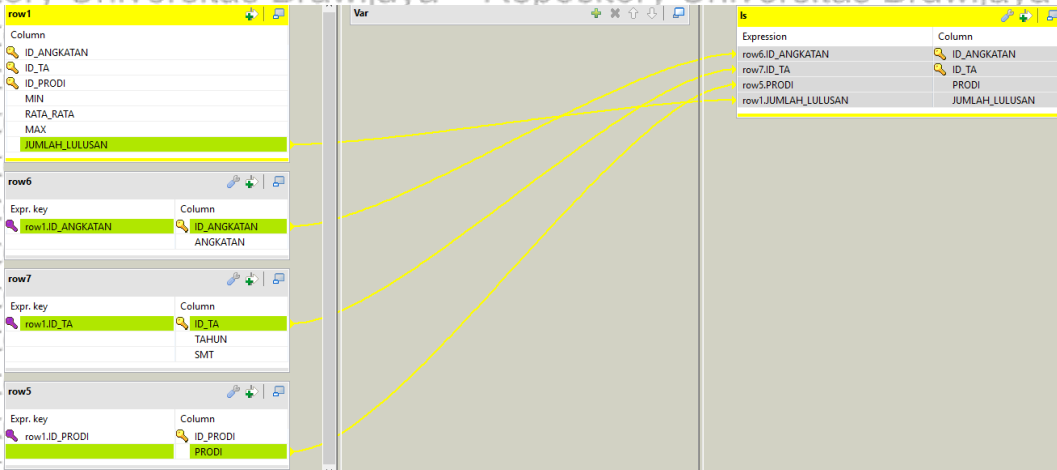
Database: "OLAP"  
 Cube: "Prestasi16-19"  
 On Element error: Reject row  
 Schema: Built-In Edit schema Sync columns

Column	Element type	Default value
JUARA	Element	"Juara 1", "Gold", "Emas", "Juara 2"...
TAHUN	Element	"2016", "2017", "2018", "2019"
TINGKAT	Element	"Internasional", "Nasional", "Regio..."
PRODI	Element	"Ilmu Komputer", "Teknik Inform..."
JUMLAH_PRESTASI	Measure	""

Gambar 4.46 Pemeriksaan elemen yang terdapat pada tabel fakta Prestasi



Gambar 4.47 Penulisan data pada *cube* Lulusan



Gambar 4.48 Pemetaan data dari data warehouse menuju OLAP

ls (Input)										tConvertType_1 (Output)									
Column	Key	Type	<input checked="" type="checkbox"/>	N.	Date Pa...	Len...	Pre...	D...	Co...	Column	Key	Type	<input checked="" type="checkbox"/>	N.	Date Pa...	Len...	Pre...	D...	Co...
ID_ANGKA...	<input checked="" type="checkbox"/>	int	<input checked="" type="checkbox"/>			10	0			ID_ANGKA...	<input checked="" type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			10	0		
ID_TA	<input checked="" type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			10	0			ID_TA	<input checked="" type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			10	0		
PRODI	<input type="checkbox"/>	Stri...	<input type="checkbox"/>			45	0			PRODI	<input type="checkbox"/>	Stri...	<input type="checkbox"/>			45	0		
JUMLAH_L...	<input type="checkbox"/>	Int...	<input type="checkbox"/>	<input checked="" type="checkbox"/>		10	0			JUMLAH_L...	<input type="checkbox"/>	Do...	<input checked="" type="checkbox"/>			10	0		

Gambar 4.49 Konversi tipe data sebelum data ditulis pada cube Lulusan

Database: "OLAP"

Cube: "Lulusan"

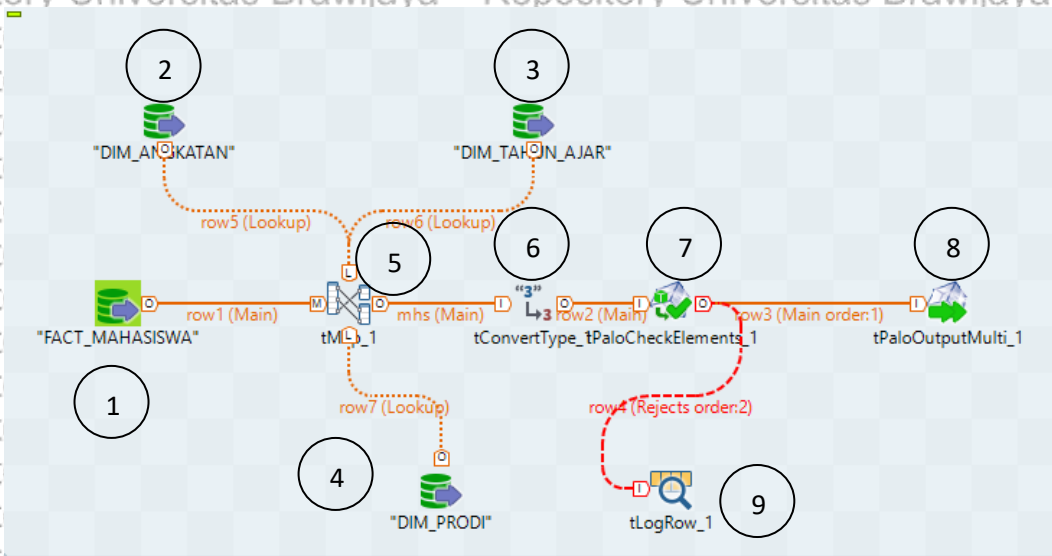
On Element error: Reject row

Schema: Built-In Edit schema Sync columns

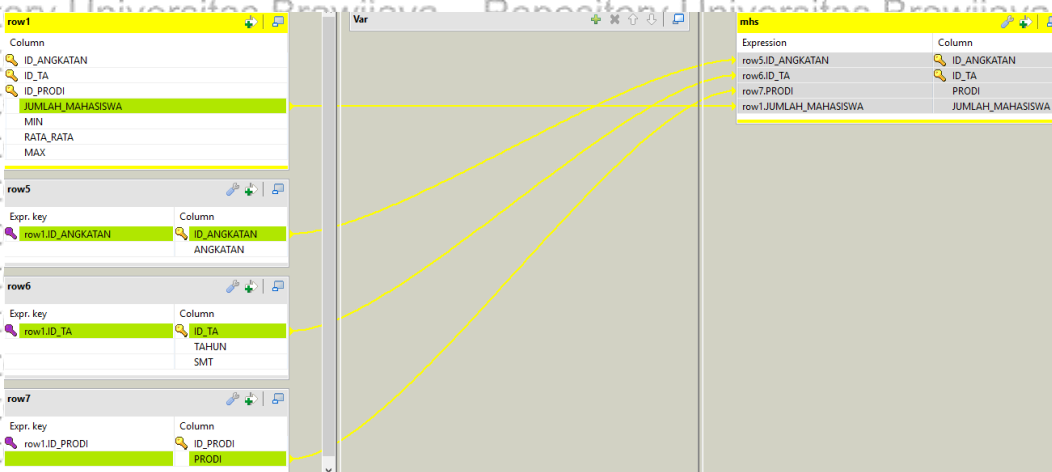
Column	Element type	Default value
ID_ANGKATAN	Element	"2009", "2010", "2011", "2012", "201...
ID_TA	Element	"2016-1", "2016-2", "2017-1", "2017...
PRODI	Element	"Ilmu Komputer", "Teknik Inform...
JUMLAH_LULUSAN	Measure	""

Gambar 4.50 Pemeriksaan elemen yang terdapat pada tabel fakta Lulusan





Gambar 4.51 Penulisan data pada *cube* Mahasiswa



Gambar 4.52 Pemetaan data dari *data warehouse* menuju OLAP

mhs (Input)										tConvertType_1 (Output)									
Column	Key	Type	☑	N..	Date Pa...	Len...	Pre...	D...	Co...	Column	Key	Type	☑	N..	Date Pa...	Len...	Pre...	D...	Co...
ID_ANGKA...	☑	int	☑			10	0			ID_ANGKA...	☑	Stri...	☑			10	0		
ID_TA	☑	Stri...	☑			10	0			ID_TA	☑	Stri...	☑			10	0		
PRODI	☐	Stri...	☑			10	0			PRODI	☑	Stri...	☑			10	0		
JUMLAH_...	☐	Int...	☑			10	0			JUMLAH_...	☐	Do...	☑			10	0		

Gambar 4.53 Konversi tipe data sebelum data ditulis pada *cube* Mahasiswa

Database "OLAP"

Cube "Mahasiswa"

On Element error Reject row

Schema Built-In Edit schema Sync columns

Column	Element type	Default value
ID_ANGKATAN	Element	"2009", "2010", "2011", "2012", "201...
ID_TA	Element	"2016-1", "2016-2", "2017-1", "2017...
PRODI	Element	"Ilmu Komputer", "Teknik Inform...
JUMLAH_MAHASISWA	Measure	""

**Gambar 4.54** Pemeriksaan elemen yang terdapat pada tabel fakta Mahasiswa Aktif

#### 4.3.3.4 Penyajian Data

Data yang telah ditulis ke dalam setiap *cube* kemudian ditampilkan pada sebuah *user report*. Pada *user report* yang ditampilkan, terdapat grafik yang berisi informasi mengenai jumlah prestasi ataupun lulusan dan mahasiswa aktif. Selain grafik, terdapat pula deskripsi singkat mengenai grafik yang ditampilkan untuk mempermudah pengguna dalam memahami isi grafik.

Pada Gambar 4.55 ditampilkan *user report* berisi grafik jumlah lulusan pada program studi Sistem Informasi berdasarkan angkatan yang lulus pada setiap tahun akademik. Lalu pada Gambar 4.56 ditampilkan *user report* berisi grafik mengenai jumlah prestasi yang diperoleh program studi Sistem Informasi sesuai dengan prestasi yang diperoleh pada setiap tingkat perlombaan beserta skor pada bidang prestasi mahasiswa yang didapat sesuai dengan rumus yang telah ditetapkan pada standar akreditasi BAN-PT. Sedangkan Gambar 4.57 menampilkan *user report* berisi grafik jumlah mahasiswa aktif pada program studi Sistem Informasi berdasarkan angkatan pada setiap tahun akademik.



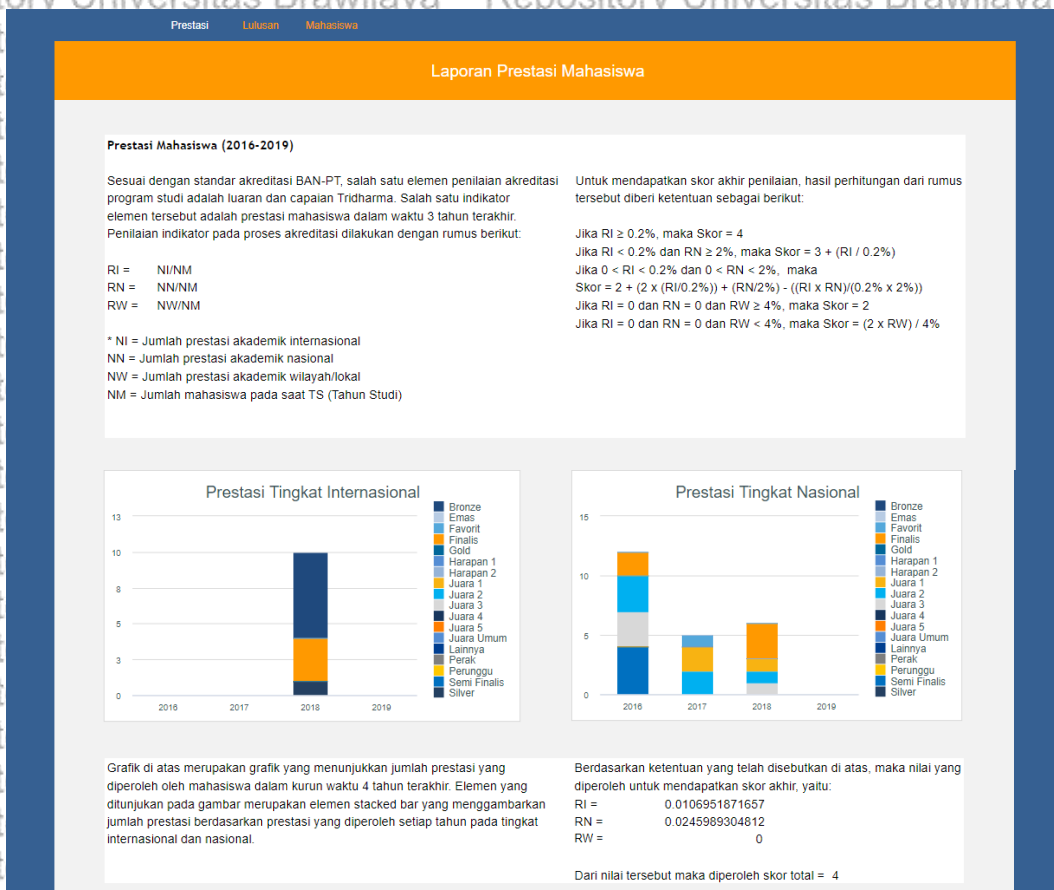
Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya



**Gambar 4.55 User report Lulusan**

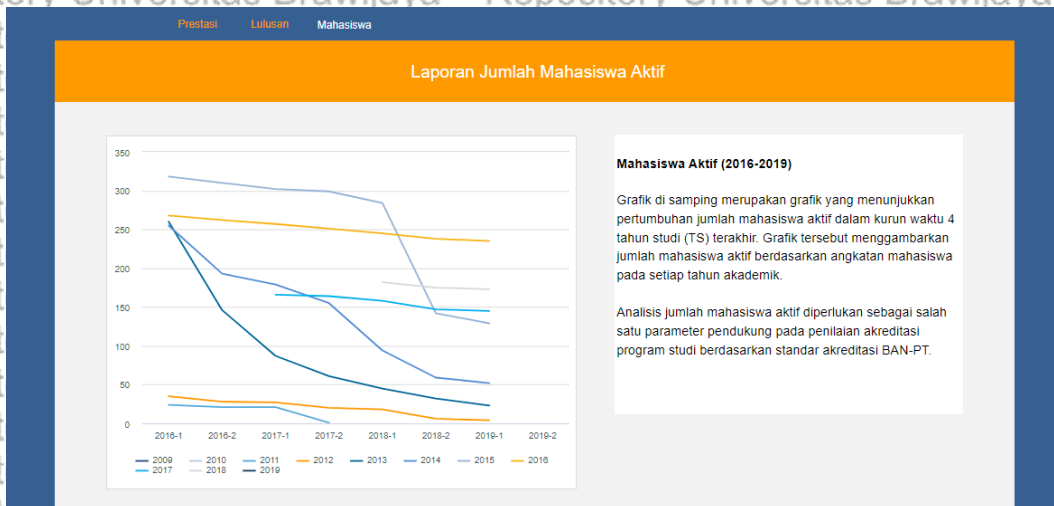


**Gambar 4.56 User report Prestasi Mahasiswa**

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya

Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya  
 Repository Universitas Brawijaya



Gambar 4.57 User report Mahasiswa Aktif

#### 4.4 Pengujian

Tahap yang dilakukan setelah tahap implementasi adalah tahap pengujian. Berdasarkan metodologi penelitian yang telah dibahas pada bab sebelumnya, jenis pengujian yang akan dilakukan adalah pengujian validasi dan pengujian performa proses ETL (*Extract, Transform, Load*).

##### 4.4.1 Pengujian Validasi

Pencocokan model multidimensional atau skema dengan kebutuhan pengguna dilakukan pada pengujian validasi. Dengan melakukan pengujian ini, nantinya akan diketahui apakah model multidimensional yang dibuat telah sesuai dengan kebutuhan pengguna dengan melakukan pengukuran terhadap validitas data yang ditampilkan dari hasil proses ETL (*Extract, Transform, Load*). Adapun hasil pengujian validasi *data warehouse* dapat dilihat pada Tabel 4.27. Untuk data yang ditampilkan berdasarkan *query* yang dijalankan dapat dilihat pada Lampiran B.

Tabel 4.27 Hasil pengujian validasi

Kasus Uji	Hasil yang Diharapkan	Query	Status
Fakta Prestasi	Data warehouse mampu menampilkan jumlah prestasi berdasarkan tahun, program studi, tingkat perlombaan dan prestasi yang diperoleh	<pre>select d.prodi, h.tahun,k.tingkat, j.juara, p.jumlah_prestasi from db2.dim_prodi d, db2.dim_tahun h, db2.dim_tingkat k, db2.dim_juara j,db2.fact_prestasi p where d.id_prodi = p.id_prodi and h.idtahun = p.id_tahun and k.id_tingkat = p.id_tingkat and</pre>	Valid

Tabel 4.27 Hasil pengujian validasi

Kasus Uji	Hasil yang Diharapkan	Query	Status
		j.id_posisi = p.id_posisi order by d.prodi, h.tahun	
Fakta Lulusan	Data warehouse mampu menampilkan jumlah lulusan setiap program studi, tahun akademik dan angkatan	select d.prodi, g.angkatan, t.tahun, t.smt, k.jumlah_lulusan from db2.dim_prodi d, db2.dim_angkatan g, db2.dim_tahun_ajar t, db2.fact_akademik k where d.id_prodi = k.id_prodi and g.id_angkatan = k.id_angkatan and t.id_ta = k.id_ta order by d.prodi, g.angkatan, t.tahun	Valid
Fakta Mahasiswa	Data warehouse mampu menampilkan jumlah mahasiswa aktif setiap program studi, tahun akademik dan angkatan	select d.prodi, g.angkatan, t.tahun, t.smt, k.jumlah_mahasiswa from db2.dim_prodi d, db2.dim_angkatan g, db2.dim_tahun_ajar t, db2.fact_mahasiswa k where d.id_prodi = k.id_prodi and g.id_angkatan = k.id_angkatan and t.id_ta = k.id_ta order by d.prodi, t.tahun, t.smt, g.angkatan	Valid

#### 4.4.2 Pengujian Performa Proses ETL (Extract, Transform, Load)

Pengujian performa proses ETL dilakukan untuk mengetahui performa proses ETL dan juga untuk memverifikasi apakah seluruh baris data pada sumber data telah dimuat secara keseluruhan ke dalam *data warehouse*. Pengujian performa proses ETL dilakukan dengan melakukan pengukuran terhadap jumlah baris data yang diproses dan kecepatan eksekusi proses tersebut. Adapun hasil pengujian performa ETL ditunjukkan pada Tabel 4.28.

Tabel 4.28 Hasil pengujian performa ETL

Proses ETL	Jumlah Baris Data	Waktu (detik)
Jumlah Prestasi Mahasiswa	257	00:00:03
Jumlah Lulusan	2.624	00:00:14
Jumlah Mahasiswa Aktif	29.865	00:00:18



Pada proses ETL Jumlah Prestasi Mahasiswa, terdapat 257 baris data yang diproses. Proses ETL ini menghasilkan jumlah prestasi yang diperoleh pada setiap tahun, program studi, tingkat perlombaan dan prestasi yang diraih.

Pada proses ETL Jumlah Lulusan, dihasilkan jumlah lulusan pada setiap tahun akademik, program studi dan angkatan dari 2.624 baris data yang diproses. Sedangkan proses ETL Jumlah Mahasiswa Aktif menghasilkan jumlah mahasiswa aktif setiap program studi, tahun akademik dan angkatan dari 29.865 baris data.

**Tabel 4.29 Hasil pengujian performa proses ETL secara keseluruhan**

Proses ETL	Waktu (detik)
Jumlah Prestasi Mahasiswa	00:00:03
Jumlah Lulusan	00:00:14
Jumlah Mahasiswa Aktif	00:00:18
<b>Rata-Rata</b>	<b>00:00:11</b>

Dari pengujian seluruh proses tersebut, estimasi waktu yang dibutuhkan saat melakukan eksekusi terhadap ketiga proses ETL di atas adalah  $\pm 11$  detik seperti yang ditunjukkan pada Tabel 4.29. Dengan hasil tersebut, dapat dikatakan bahwa proses ETL memerlukan waktu yang tidak begitu lama. Selain itu, data yang dihasilkan bersifat akurat karena baris data yang terdapat dalam sumber data telah dimuat secara keseluruhan ke dalam *data warehouse*.



## BAB 5 PENUTUP

### 5.1 Kesimpulan

Berdasarkan perancangan, implementasi dan pengujian yang telah dilakukan pada penelitian yang bertujuan untuk dapat memudahkan proses rekapitulasi data prestasi mahasiswa dan akademik mahasiswa sesuai dengan standar akreditasi BAN-PT ini, maka kesimpulan yang dapat ditarik, antara lain:

1. Perancangan *data warehouse* prestasi mahasiswa yang dilakukan berdasarkan *Nine Step Design Methodology* berhasil dilakukan. Hal tersebut mengacu pada hasil pengujian validasi yang membuktikan bahwa pengembangan *data warehouse* telah sesuai dengan kebutuhan dari sisi pengguna. Perancangan *data warehouse* prestasi mahasiswa menghasilkan 3 fakta, yaitu fakta Prestasi, Lulusan dan Mahasiswa Aktif dengan 6 dimensi yang digunakan untuk dapat menghasilkan data yang diinginkan dimana masing-masing tabel fakta tersebut membentuk 3 *star schema*.
2. Pengujian validasi pada tiga fakta yang dilakukan menghasilkan status yang valid yang berarti data yang ditampilkan pada pengujian ini telah sesuai dengan kebutuhan pengguna.
3. Pengujian performa proses ETL yang dilakukan menghasilkan informasi mengenai waktu yang diperlukan untuk melakukan proses ETL pada setiap tabel fakta, yaitu < 20 detik dengan rata-rata waktu eksekusi untuk semua tabel fakta membutuhkan waktu  $\pm 11$  detik. Sehingga dapat dikatakan bahwa eksekusi proses ETL pada *data warehouse* yang dikembangkan memerlukan waktu yang tidak begitu lama dan menghasilkan data yang akurat karena baris data yang terdapat dalam sumber data telah dimuat secara keseluruhan ke dalam *data warehouse*.
4. Penelitian ini menghasilkan *user report* yang dapat digunakan pihak manajemen untuk menganalisis data hasil eksekusi proses ETL dengan tampilan grafik sehingga mempermudah dalam mendukung keputusan pada sisi manajerial.

### 5.2 Saran

Pada penelitian dengan judul "Pengembangan *Data Warehouse* Prestasi Mahasiswa (Studi Pada: Fakultas Ilmu Komputer Universitas Brawijaya)" ini masih terdapat banyak kekurangan yang dapat dilakukan perbaikan selanjutnya. Adapun saran yang dapat dilakukan pada pengembangan penelitian selanjutnya, yaitu:

1. Dalam pengembangan selanjutnya, mungkin dapat menggunakan aplikasi OLAP lainnya yang dapat diakses pada jaringan luas sehingga tidak hanya dapat diakses pada jaringan lokal saja.



## LAMPIRANA HASIL WAWANCARA

Tanggal : 14 November 2019

Narasumber : Hermawan Dwi Putra, S.H., M.H.

Jabatan : Staf Kemahasiswaan

1. Dalam penilaian akreditasi program studi, hal apa saja yang menjadi parameter penilaiannya?

Jawaban: Untuk parameter penilaiannya sendiri ada banyak, mulai dari dosen program studinya, kemudian kualitas lulusannya, pencapaian mahasiswanya, dan yang lainnya.

2. Apakah prestasi mahasiswa dapat dijadikan sebagai salah satu parameter penilaian akreditasi program studi?

Jawaban: Oh jelas, prestasi mahasiswa kan termasuk dalam pencapaian mahasiswa. Dan untuk prestasi mahasiswa ini tidak hanya digunakan untuk kepentingan penilaian akreditasi saja, banyak juga digunakan untuk kepentingan lainnya.

3. Analisis seperti apa yang dibutuhkan dari prestasi mahasiswa untuk kebutuhan akreditasi?

Jawaban: Contohnya seperti analisis prestasi mahasiswa dilihat dari tingkat perlombaan, misalkan perlombaan internasional, nasional atau wilayah. Kemudian dilihat dari tahun diperolehnya prestasi juga bisa.

4. Apakah ada hal lainnya yang diperlukan untuk proses analisis prestasi mahasiswa tersebut?

Jawaban: Untuk penilaian akreditasi pada parameter prestasi mahasiswa dibutuhkan juga data mahasiswa dari program studi yang ingin dinilai akreditasinya.

5. Hal yang dibutuhkan untuk analisis mengenai prestasi mahasiswa dilihat dari segi apa saja ya?

Jawaban: Pertama mungkin dilihat dari tahun, kemudian tingkat perlombaan seperti yang tadi sudah saya jelaskan, dan yang terakhir bisa dilihat dari prestasi yang diperoleh, misalkan juara 1, juara 2 dan seterusnya.

6. Data prestasi mahasiswa yang diperlukan oleh program studi bersumber dari mana ya mas?





Jawaban: Kalau data prestasi mahasiswa didapatkan dari Bagian Kemahasiswaan jadi setiap kali program studi butuh data prestasi mahasiswa, ya harus minta ke bagian tersebut.

7. Jadi belum ada integrasi antara data prestasi dengan data mahasiswa ya mas?

Jawaban: Iya belum ada. Karena program studi harus mengambil dua data yang berbeda itu di dua sumber yang berbeda.

8. Untuk proses penilaian dari sisi prestasinya itu bagaimana ya mas?

Jawaban: Kalau untuk detil penilaiannya bisa dilihat di situs BAN-PT, di sana lengkap tertulis.

Peneliti,

Sausan Zahrah

Narasumber,

Hermawan Dwi Putra, S.H., M.H.

#### Lampiran 1 Naskah wawancara



LAMPIRAN B HASIL PENGUJIAN

B.1 Pengujian Validasi

PRODI	TAHUN	TINGKAT	JUARA	JUMLAH PRESTASI
Pendidikan Teknologi Informasi	2017	Nasional	Juana 1	1
Pendidikan Teknologi Informasi	2017	Nasional	Juana 2	1
Pendidikan Teknologi Informasi	2018	Nasional	Juana 1	1
Pendidikan Teknologi Informasi	2018	Nasional	Finalis	5
Pendidikan Teknologi Informasi	2018	Internasional	Finalis	2
Pendidikan Teknologi Informasi	2018	Internasional	Silver	1
Pendidikan Teknologi Informasi	2018	Internasional	Bronze	1
Pendidikan Teknologi Informasi	2019	Internasional	Juana 1	1
Sistem Informasi	2016	Nasional	Semi Finalis	4
Sistem Informasi	2016	UB	Juana 2	2
Sistem Informasi	2016	Nasional	Juana 2	3
Sistem Informasi	2016	Nasional	Juana 3	3
Sistem Informasi	2016	Nasional	Finalis	2
Sistem Informasi	2017	Nasional	Juana 2	2
Sistem Informasi	2017	Nasional	Favorit	1
Sistem Informasi	2017	Nasional	Juana 1	2
Sistem Informasi	2018	Internasional	Finalis	3
Sistem Informasi	2018	Nasional	Juana 2	1
Sistem Informasi	2018	Nasional	Juana 3	1
Sistem Informasi	2018	Nasional	Juana 1	1
Sistem Informasi	2018	Nasional	Finalis	3
Sistem Informasi	2018	Internasional	Bronze	6
Sistem Informasi	2018	Internasional	Silver	1
Teknik Informatika	2016	Nasional	Juana 3	9
Teknik Informatika	2016	Nasional	Juana 1	5
Teknik Informatika	2016	Internasional	Gold	7
Teknik Informatika	2016	Internasional	Juana 3	1
Teknik Informatika	2016	UB	Juana 1	6
Teknik Informatika	2016	Nasional	Favorit	2
Teknik Informatika	2016	Internasional	Bronze	7
Teknik Informatika	2016	Nasional	Juana 2	6
Teknik Informatika	2016	UB	Juana 2	6
Teknik Informatika	2016	Nasional	Semi Finalis	1
Teknik Informatika	2016	Nasional	Finalis	23
Teknik Informatika	2017	Nasional	Juana 1	4
Teknik Informatika	2017	Nasional	Favorit	2
Teknik Informatika	2017	Regional	Harapan 1	1
Teknik Informatika	2017	Provinsi	Juana 1	1
Teknik Informatika	2017	Regional	Juana 1	1
Teknik Informatika	2018	Nasional	Juana 3	13
Teknik Informatika	2018	Nasional	Juana 1	6
Teknik Informatika	2018	Internasional	Silver	6
Teknik Informatika	2018	Internasional	Gold	15
Teknik Informatika	2018	Internasional	Bronze	2
Teknik Informatika	2018	Internasional	Juana 2	10
Teknik Informatika	2018	Regional	Juana 3	1
Teknik Informatika	2018	Internasional	Finalis	7
Teknik Informatika	2018	Nasional	Finalis	15
Teknik Informatika	2019	Nasional	Juana 1	1
Teknik Informatika	2019	Internasional	Gold	5
Teknik Informatika	2019	Nasional	Juana 2	1
Teknik Informatika	2019	Nasional	Juana 3	2
Teknik Komputer	2016	UB	Juana 2	1
Teknik Komputer	2016	Nasional	Juana 1	4
Teknik Komputer	2016	Nasional	Finalis	9
Teknik Komputer	2016	Regional	Juana 1	3
Teknik Komputer	2016	Nasional	Juana 2	7
Teknik Komputer	2017	Nasional	Harapan 1	1
Teknik Komputer	2017	Nasional	Juana 3	3
Teknik Komputer	2017	Nasional	Finalis	1
Teknik Komputer	2017	Internasional	Gold	1
Teknik Komputer	2017	Nasional	Juana 2	3
Teknik Komputer	2018	Internasional	Bronze	1
Teknik Komputer	2018	Internasional	Silver	1
Teknik Komputer	2018	Internasional	Gold	1
Teknik Komputer	2019	Nasional	Juana 3	4
Teknik Komputer	2019	Nasional	Juana 1	1
Teknologi Informasi	2018	Nasional	Juana 3	1
Teknologi Informasi	2018	Nasional	Finalis	1

69 record(s) selected.

Lampiran 2 Pengujian validasi fakta Prestasi



Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya

Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya

Repository  
Repository  
Repository  
Repository  
Repository  
Repository

PRODI	ANGKATAN	TAHUN	SMT	JUMLAH_LULUSAN
Ilmu Komputer	2010	2016	1	5
Ilmu Komputer	2010	2016	2	25
Ilmu Komputer	2011	2016	1	5
Ilmu Komputer	2011	2016	2	9
Ilmu Komputer	2011	2017	1	5
Ilmu Komputer	2011	2017	2	17
Ilmu Komputer	2011	2018	1	1
Pendidikan Teknologi Informasi	2015	2018	2	21
Pendidikan Teknologi Informasi	2015	2019	1	9
Sistem Informasi	2011	2016	1	3
Sistem Informasi	2011	2016	2	2
Sistem Informasi	2011	2017	2	19
Sistem Informasi	2012	2016	1	34
Sistem Informasi	2012	2016	2	6
Sistem Informasi	2012	2017	1	2
Sistem Informasi	2012	2017	2	3
Sistem Informasi	2012	2018	1	3
Sistem Informasi	2012	2018	2	9
Teknik Informatika	2013	2018	1	45
Teknik Informatika	2013	2018	2	25
Teknik Informatika	2013	2019	1	3
Teknik Informatika	2014	2016	2	47
Teknik Informatika	2014	2017	1	37
Teknik Informatika	2014	2017	2	79
Teknik Informatika	2014	2018	1	97
Teknik Informatika	2014	2018	2	58
Teknik Informatika	2014	2019	1	10
Teknik Informatika	2015	2016	2	4
Teknik Informatika	2015	2017	1	2
Teknik Informatika	2015	2018	1	51
Teknik Informatika	2015	2018	2	191
Teknik Informatika	2015	2019	1	45
Teknik Informatika	2016	2017	2	2
Teknik Informatika	2016	2018	1	5
Teknik Informatika	2016	2019	1	1
Teknik Informatika	2017	2018	2	5
Teknik Informatika	2017	2019	1	3
Teknik Komputer	2011	2016	1	1
Teknik Komputer	2011	2016	2	2
Teknik Komputer	2011	2017	2	10
Teknik Komputer	2012	2016	1	16
Teknik Komputer	2012	2016	2	25
Teknik Komputer	2012	2017	1	3
Teknik Komputer	2012	2017	2	8
Teknik Komputer	2012	2018	1	7
Teknik Komputer	2012	2018	2	6
Teknik Komputer	2013	2016	1	5
Teknik Komputer	2013	2016	2	58
Teknik Komputer	2013	2017	1	39
Teknik Komputer	2013	2017	2	15
Teknik Komputer	2013	2018	1	33
Teknik Komputer	2013	2018	2	12
Teknik Komputer	2013	2019	1	8
Teknik Komputer	2014	2016	1	1
Teknik Komputer	2014	2016	2	14
Teknik Komputer	2014	2017	1	11
Teknik Komputer	2014	2017	2	19
Teknik Komputer	2014	2018	1	40
Teknik Komputer	2014	2018	2	22
Teknik Komputer	2014	2019	1	8

109 record(s) selected.

### Lampiran 3 Pengujian validasi fakta Lulusan

Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya

Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya  
Repository Universitas Brawijaya

Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository  
Repository



**DAFTAR REFERENSI**

- Amo, W. C. (2000). *Microsoft SQL Server OLAP developer's guide*. Retrieved from <https://www.books-by-isbn.com/0-7645/0764546430-SQL-Server-7-OLAP-Developer-s-Guide-With-CD-ROM-0-7645-4643-0.html>
- Ardista, N., Purbandini, P., & Taufik, T. (2017). Rancang Bangun Data Warehouse Untuk Pembuatan Laporan dan Analisis pada Data Kunjungan Pasien Rawat Jalan Rumah Sakit Universitas Airlangga Berbasis Online Analytical Processing (OLAP). *Journal of Information Systems Engineering and Business Intelligence*, 3(1), 40. <https://doi.org/10.20473/jisebi.3.1.40-51>
- Bowen, J. (2012). Getting Started with Talend Open Studio for Data Integration. Retrieved February 25, 2020, from <https://www.talend.com/resources/get-started-talend-open-studio-data-integration/>
- Connolly, T., & Begg, C. (2015). *Database Systems* (6th ed.). Retrieved from [www.pearsonglobaleditions.com/connolly](http://www.pearsonglobaleditions.com/connolly)
- F. Azizah, A., Agus Hendrawan, R., & Aulia Vinarti, R. (2012). Pembuatan Data Warehouse Penjualan Produk dan Penerapan Dalam Studi Kasus Divisi Greenscope Energy – PT. Tustika Nagata Surabaya. *JURNAL TEKNIK POMITS*, 1(1), 1–5. Retrieved from <http://digilib.its.ac.id/public/ITS-paper-24337-5208100032-Paper.pdf>
- Firmansyah, M. (2017). *Pembangunan Data Warehouse Keuangan Pada Instansi Pendidikan (Studi Kasus: Fakultas XYZ)*.
- Hammergren, T. (1989). *Data Warehousing on the Internet: Accessing the Corporate Knowledge Base*.
- Han, J., & Kamber, M. (2006). *Designing Data-Intensive Web Applications*. Retrieved from <http://ebooks.bharathuniv.ac.in/gdlc1/gdlc1/Software Engineering/Data Mining - Concepts and Techniques - 2nd Edition - Impressao.pdf>
- Humdiana, & Indrayani, E. (2008). *SISTEM INFORMASI MANAJEMEN*. Retrieved from <https://humdiana.files.wordpress.com/2018/03/book-sim-hum-evi.pdf>
- Inmon, W. H. (2005). *Building the Data Warehouse, Fourth Edition* (4th ed.). Retrieved from [http://homes.dcc.ufba.br/~mauricio052/Material Artigo/Building The Data Warehouse \(2005\) Fourth Edition-Inmon-Wiley.pdf](http://homes.dcc.ufba.br/~mauricio052/Material Artigo/Building The Data Warehouse (2005) Fourth Edition-Inmon-Wiley.pdf)
- Jedox. (2019). Enterprise Performance Management and BI unified in one seamless solution. Retrieved February 25, 2020, from <https://www.jedox.com/en/>
- Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit 2nd Ed.* (2nd ed.; Robert Elliott, Ed.). Retrieved from <http://users.itk.ppke.hu/~szoer/DW/Kimball & Ross - The Data Warehouse Toolkit 2nd Ed %5BWiley 2002%5D.pdf>



Kusnawi, (2010). Multidimensional Data Warehouse dengan Menggunakan MySQL. *DASI*, 11(3).

McCreadie, M., & Rice, R. E. (1999). Trends in analyzing access to information. Part I: cross-disciplinary conceptualizations of access. *Information Processing & Management*, 35(1), 45–76. [https://doi.org/10.1016/S0306-4573\(98\)00037-5](https://doi.org/10.1016/S0306-4573(98)00037-5)

McLeod, R., & Schell, G. P. (2007). *Management information systems*. Retrieved from <https://www.worldcat.org/title/management-information-systems/oclc/64453371>

Mujiono, & Musdholifah, A. (2016). IJCCS (Indonesian Journal of Computing and Cybernetics Systems). *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 10(1), 1–10. Retrieved from <https://jurnal.ugm.ac.id/ijccs/article/view/11184/8424>

Nielsen, J. (2009). Powers of 10: Time Scales in User Experience. Retrieved April 4, 2020, from <https://www.nngroup.com/articles/powers-of-10-time-scales-in-ux/>

Oktavia, T. (2011). *Perancangan Model Data Warehouse Dalam 2011*(semnasIF), 93–100. Retrieved from <http://jurnal.upnyk.ac.id/index.php/semnasif/article/view/1283>

Oracle. (2017). Introduction to Data Warehousing Concepts. Retrieved December 17, 2019, from <https://docs.oracle.com/database/121/DWHSG/concept.htm#DWHSG8073>

Ponniah, P. (2001). *DATA WAREHOUSING FUNDAMENTALS Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals*. Retrieved from <https://anuradhasrinivas.files.wordpress.com/2013/03/data-warehousing-fundamentals-by-paulraj-ponniah.pdf>

Prasetyo, A., Soedijono, B., & Amborowati, A. (2017). Perancangan Data Warehouse Untuk Mendukung Perencanaan Pemasaran Perguruan Tinggi. *Jurnal Telematika*, 10(1), 1–22.

Taleb, D. T. M. J. Al, & Saedi, M. M. K. Al. (2017). Data Warehouse System for Higher Education Student Information System. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 19(5), 47–53. Retrieved from <http://www.iosrjournals.org/iosr-jce/papers/Vol19-issue5/Version-3/H1905034753.pdf>

Umam, Q., Wicaksono, S. A., & Purnomo, W. (2019). *Analisis Dan Perancangan Data Warehouse Menggunakan Pendekatan ( Studi Pada Dinas Komunikasi dan Informatika Kabupaten Sidoarjo )*. 3(2), 1824–1833.

Vassiliadis, P., Simitsis, A., & Skiadopoulos, S. (2002). Conceptual modeling for ETL processes. *ACM International Workshop on Data Warehousing and OLAP (DOLAP)*, 14–21. <https://doi.org/10.1145/583890.583893>

