

APLIKASI OLAP PROFIL MAHASISWA DAN LULUSAN

Arie Nugroho

Fakultas Teknik, Program Studi Sistem Informasi
Universitas Nusantara PGRI Kediri
Email: arienugroho@unpkediri.ac.id

ABSTRAK

Profil mahasiswa dan lulusan adalah salah satu data yang sangat dibutuhkan suatu universitas dalam proses akreditasi. Dengan data tersebut universitas dapat mengetahui informasi tentang kondisi akademik mahasiswa. Data-data yang bisa didapat adalah jumlah mahasiswa, lulusan dan kelompok mahasiswa berdasarkan *Indeks Prestasi kumulatif (IPK)* dalam periode tertentu. Data mahasiswa yang belum terintegrasi menyebabkan kesulitan dalam menampilkan data – data yang dibutuhkan, sehingga dibutuhkan alat bantu berupa aplikasi *OLAP* profil mahasiswa dan lulusan dengan data-data yang diperoleh dari program studi. Data – data yang digunakan diambil dari Program Studi Sistem Informasi Fakultas Teknik Universitas Nusantara PGRI Kediri. Pada penelitian ini dilakukan tahap analisa dan pemecahan masalah, perancangan tampilan aplikasi, tahap uji coba aplikasi dan implementasi. Hasil dari penelitian ini adalah aplikasi *OLAP* untuk menampilkan prosentase IPK lulusan berdasarkan *range* tertentu dan menampilkan IPK minimum, rata-rata dan maksimum.

Kata kunci: *OLAP*, profil, mahasiswa, lulusan, IPK.

ABSTRACT

Profile of students and graduates is one of the data needed by a university in the accreditation process. With these data universities can find information about student academic conditions. The data that can be obtained is the number of students, graduates and student groups based on the Com- putative Achievement Index (GPA) within a certain period. Student data's that has not been integrated causing difficulties in displaying the required data's, so it takes tools in the form of OLAP application of student profiles and graduates with data's obtained from the study program. In this research is done analysis and problem solving phase, design of application display, application test phase and implementation. The result of this research is OLAP application to show the percentage of GPA of graduates based on certain range and display minimum GPA, average and maximum.

Keywords: *OLAP, profile, students, graduates.*

1. PENDAHULUAN

Akreditasi perguruan tinggi digunakan untuk mengetahui *standart* mutu perguruan tinggi. Akreditasi perguruan tinggi diselenggarakan oleh Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT). Dalam proses akreditasi perguruan tinggi diharuskan mengisi borang akreditasi sesuai dengan institusi yang akan diajukan dalam proses akreditasi. Berdasarkan pengumuman no : 609/BAN-PT/Edaran/III/2009 perihal Pemberlakuan Borang Baru untuk Akreditasi Program Studi Sarjana. Borang akreditasi program studi mempunyai tujuh standart yang harus dipenuhi. Standart satu membahas tentang visi, misi, tujuan dan sasaran, serta strategi pencapaian. *Standart* dua membahas tentang tata pamong, kepemimpinan, sistem pengelolaan, dan penjaminan mutu. *Standart* tiga membahas tentang mahasiswa dan lulusan. *Standart* empat membahas tentang sumber daya manusia *Standart* lima membahas tentang kurikulum, pembelajaran, dan suasana akademik. *Standart* enam membahas tentang pembiayaan, sarana dan prasarana, serta sistem informasi. *Standart* tujuh membahas tentang penelitian, pelayanan/pengabdian kepada masyarakat, dan kerjasama.

Dalam penelitian ini *standart* yang dibahas hanya *standart* tiga yang membahas tentang mahasiswa dan lulusan. *Standart* ini membahas tentang jumlah total mahasiswa, jumlah mahasiswa regular, jumlah mahasiswa baru, jumlah lulusan, jumlah mahasiswa berdasarkan IPK dan prosentasenya. Selain itu juga membahas jumlah mahasiswa regular tujuh tahun terakhir setiap tahunnya. Dalam pengisian borang akreditasi program studi untuk *standart* tiga yang membahas tentang profil mahasiswa dan lulusan, program studi sistem informasi Fakultas Teknik Universitas Nusantara PGRI Kediri mengalami kesulitan karena

data mahasiswa yang semakin bertambah, belum terintegrasi sehingga akan memerlukan waktu yang lama untuk menampilkan jumlah mahasiswa berdasarkan isian borang.

Dari pemaparan masalah di atas, muncul gagasan bagi penulis untuk membuat suatu alat bantu berupa aplikasi *OLAP* profil mahasiswa dan lulusan yang dapat mengatasi permasalahan tersebut. Alat bantu ini diharapkan dapat mendukung pengisian borang akreditasi program studi *standart* tiga tentang profil mahasiswa dan lulusan di Universitas Nusantara PGRI Kediri Fakultas Teknik Program studi Sistem Informasi.

Dari Latar Belakang yang telah dijelaskan sebelumnya maka dapat diidentifikasi permasalahannya adalah proses pengisian borang akreditasi tentang profil mahasiswa dan lulusan mengalami kesulitan karena data mahasiswa yang semakin bertambah, belum terintegrasi sehingga akan memerlukan waktu yang lama untuk menampilkan jumlah mahasiswa berdasarkan isian boring. Dalam melaksanakan perencanaan dan pembuatan sistem yang akan dibuat pada tugas akhir ini, permasalahan yang muncul adalah Bagaimana merancang dan membangun aplikasi *OLAP* Profil mahasiswa dan lulusan?.

Tujuan dari penelitian ini adalah membuat aplikasi *OLAP* profil mahasiswa dan lulusan pada Program Studi Sistem Informasi Fakultas Teknik Universitas Nusantara PGRI Kediri. Manfaat dari penelitian ini adalah Program studi Sistem Informasi dapat mengetahui informasi tentang profil mahasiswa dan lulusan serta program studi Sistem Informasi lebih mudah dan cepat dalam pengisian borang akreditasi *standart* 3 tentang profil mahasiswa dan lulusan.

1.1 Data Warehouse

Pada dasarnya *Data warehouse* adalah databsase dan merupakan pusat data yang dibentuk dari hasil penggabungan dan pengolahan data dari beragam sumber data yang digunakan untuk keperluan pelaporan dan analisis data [1]. *Data warehouse* adalah sebuah sistem yang mengambil dan menyatukan data secara periodik dari sistem sumber menuju ke penyimpanan data dimensional atau penyimpanan data normalisasi [2]. Biasanya data yang tersimpan didalamnya merupakan data sejarah (*history data*) yang digunakan untuk melakukan analisa untuk mendukung proses pengambilan keputusan. Selain itu data diperbarui secara berkelompok bukan setiap saat ketika proses transaksi berjalan pada sistem sumber. Menurut Han dan Kamber (2008), suatu basis data dapat digolongkan sebagai *data warehouse* jika memiliki karakteristik sebagai berikut :

- a. Berorientasi Subjek
Data diorganisasikan oleh subjek secara rinci, misalnya : berdasarkan pelanggan, jenis, dan lain-lain.
- b. Terintegrasi
Basis data mencakup data dari kebanyakan atau semua aplikasi operasional organisasi dan data tersebut dibuat secara konsisten.
- c. *Time-Variant*
Data tidak menyediakan status saat ini, data tersebut disimpan untuk periode lima atau sepuluh tahun bahkan lebih dan digunakan untuk *tren*, peramalan, dan perbandingan. Waktu merupakan dimensi penting yang harus didukung oleh semua *data warehouse*.
- d. *NonVolatile*
Data yang dimasukkan ke dalam *data warehouse* adalah data yang *read-only*, yang tidak dapat dirubah. Data yang sudah lama dihapus dan perubahan direkam sebagai data yang baru.
- e. Ringkas.
Jika diperlukan, data operasional dimasukkan ke dalam ringkasan data. Dalam *data warehouse* biasanya tidak dinormalisasi sehingga masih terjadi redudansi atau duplikasi data.
- f. Sumber.
Semua sumber data tersedia dalam *data warehouse*, baik *internal* maupun *eksternal*.
- g. Metadata
Metadata mengacu pada data tentang data yang menguraikan struktur dan beberapa arti tentang data, dengan demikian mendukung penggunaan yang *efektif* atau tidak *efektif* dari data.

1.2 Online Transactional Processing

Online Transactional Processing (OLTP) adalah teknologi untuk mengelola aplikasi yang berorientasi pada transaksi [1]. Karena teknologi ini berhubungan dengan *database*, setiap *database* yang berhubungan dengan *OLTP* disebut *database OLTP*. *Database OLTP* adalah *database* yang umum digunakan pada aplikasi yang berorientasi pada transaksi, yaitu aplikasi yang cenderung lebih banyak melakukan proses *insert,update* dan *delete* secara *real-time* dan umumnya ditujukan untuk aplikasi yang tergolong *mission*

critical application, yaitu aplikasi yang jika terjadi masalah atau gangguan bisa menyebabkan proses bisnis terganggu.

1.3 OnLine Analytical Processing

OnLine Analytical Processing (OLAP) adalah teknologi untuk menjawab kebutuhan analitik [1]. Seperti juga *OLTP*, *OLAP* berhubungan dengan *database*. Setiap *database* yang berhubungan dengan *OLAP* disebut *database OLAP*. *Database OLAP* adalah *database* yang dimaksimalkan untuk kecepatan dalam pembacaan (*select query*). *OLAP* mengandung dua tipe dasar, yaitu *measures* dan *dimension*. *Measures* adalah data bilangan yang terukur, misalkan kuantitas (*quantity*), harga (*price*), nilai rata-rata (*averages*) dari kelompok nilai tertentu, jumlah (*sum*) dari kelompok nilai tertentu. *Dimension* mengacu pada kategori yang digunakan untuk mengatur *measures*. Biasanya data dikelompokkan dalam bentuk bertingkat (*level*). Dimensi yang umumnya hampir selalu ada adalah dimensi waktu (*time dimension*). Pada dimensi ini, hierarki yang disusun biasanya tahun, kuartal, bulan dan hari. Bisa juga jika diperlukan ada minggu di antara bulan dan hari. *OLAP* merupakan perpaduan dinamis analisis dan gabungan dari data *multi dimensional* dalam jumlah yang besar [3]. *OLAP* merupakan kumpulan aturan yang menyediakan sebuah kerangka *dimensional* untuk mendukung pengambilan keputusan. *OLAP* juga merupakan sebuah pendekatan secara cepat menyediakan jawaban-jawaban terhadap *query* analitik yang *multi dimensi*. *OLAP* adalah bagian dari kategori yang lebih *global* dari pemikiran bisnis, yang juga merangkum hubungan antara pelaporan dan penggalian data. Di dalam inti sebaran sistem *OLAP* merupakan konsep dari sebuah kubus *OLAP* (disebut juga sebagai kubus *multi dimensi* atau *hyper cube*) yang terdiri dari *numeric fact* yang disebut ukuran dan dikategorikan sebagai dimensi. Kubus metadata secara khusus terbuat dari sebuah skema bintang atau skema kristal salju dari tabel di dalam sebuah *Database* yang berhubungan. Ukuran diturunkan dari record dalam *fact table* dan dimensi-dimensi yang diturunkan dari tabel-tabel dimensi. Dengan menggunakan metode *OLAP*, data-data yang terkait dengan kondisi lulusan institusi pendidikan dapat dikumpulkan kedalam suatu *data warehouse*. *Data warehouse* ini berfungsi sebagai sumber data untuk menghasilkan informasi histori lulusan, sehingga pihak manajemen dapat memanfaatkan informasi tersebut sebagai bahan dalam pengambilan keputusan, agar peningkatan kompetensi lulusan selalu menghasilkan pencapaian maksimal [4]. Pembangunan aplikasi *OnLine Analytical Processing (OLAP) Data Warehouse* merupakan salah satu cara untuk mengekstrak informasi penting dari beberapa sistem informasi yang berbeda. Di mana data yang sudah terintegrasi dapat dimanfaatkan untuk kegiatan penyampaian informasi yang dapat ditinjau dari berbagai sudut pandang atau dimensi dan dapat diatur tingkatan rincian datanya [5].

1.4 Data Staging

Data staging menyediakan sebuah tempat dengan satu set fungsi untuk membersihkan, mengubah, menggabungkan, mengkonversi, mencegah duplikasi data, dan menyiapkan data sumber untuk penyimpanan dan penggunaan dalam *data warehouse*. Tahap pembersihan ini dikenal juga dengan istilah *Extract, Transformation and Loading (ETL)* [6].

1.5 Extract, Transform, Load

Extract, Transform Load (ETL) adalah sekumpulan proses untuk mengambil dan memproses data dari satu atau banyak sumber menjadi sumber baru [1]. Sumber bisa dari beberapa sumber data, tidak hanya dari *OLTP* saja, bisa juga dari *website*, *file* teks, *spreadsheet* dan lain sebagainya. *Extract* adalah suatu proses yang mengidentifikasi seluruh sumber data yang relevan dan kemudian mengambil data dari sumber-sumber data tersebut [7]. Proses *extract* contohnya membaca *file* dari *excel*, dari *database* dan lain-lain. *Transform* adalah suatu proses yang memiliki peran dalam melakukan perubahan dan integrasi skema data serta struktur yang berbeda ke dalam skema dan struktur yang telah didefinisikan sebelumnya oleh *data warehouse*. Fungsi-fungsi dalam proses *transform* dapat berupa pemindahan data, validasi data sesuai aturan yang ditetapkan, modifikasi (isi, tipe atau struktur data), integrasi atau penggabungan data dari sumber-sumber lain, perhitungan dan lain sebagainya. *Load* adalah proses pemindahan data secara fisik dari *system* operasional ke dalam *data warehouse*. Contoh proses *load* adalah proses *transform* yang kemudian disimpan ke *database OLAP*.

2. METODOLOGI PENELITIAN

Penelitian ini dilaksanakan melalui 4 (empat) proses, sebagai berikut :

- a. Tahap Analisa Pada tahap ini, dilakukan analisa dan pemecahan masalah dalam pembuatan program aplikasi dengan mencari informasi-informasi tentang profil mahasiswa dan lulusan di program studi sistem informasi Universitas Nusantara PGRI Kediri. Sumber lain didapatkan dari internet berupa jurnal dan dari buku- buku yang berkaitan dengan penelitian ini.

- b. Tahap Perancangan, Pada tahap ini peneliti membuat rancangan dan tampilan aplikasi *OLAP* yang akan digunakan.
- c. Tahap Uji Coba, Pada tahap ini, peneliti melakukan uji coba dan evaluasi tampilan dan cara kerja aplikasi *OLAP* yang akan diimplementasikan di sistem operasi *Windows*.
- d. Tahap Implementasi, Pada tahap ini akan dilakukan pengimplementasian hasil rancangan dengan menggunakan program Embarcadero Delphi 10.2 Tokyo dengan database MySQL

3. HASIL DAN PEMBAHASAN

3.1 Proses Extract

Program studi Sistem Informasi Universitas Nusantara PGRI Kediri dalam mengolah data nilai mahasiswa dilakukan oleh dosen dengan menggunakan format excel. Setiap dosen yang mampu matakuliah tertentu mengikuti format pengisian nilai yang telah disediakan oleh program studi. Format *excel* yang telah diisi nilai oleh dosen kemudian diserahkan kembali kepada petugas admin program studi. Contoh data nilai mahasiswa dalam format *excel* ditunjukkan pada gambar 1

DAFTAR NILAI MAHASISWA SEMESTER GANJIL 2015 / 2016								
PROGRAM STUDI : SISTEM INFORMASI								
KELAS : II D								
MATA KULIAH	: Basis Data							
DOSEN PEMBINA	: Rini Indriati, M. Kom							
NO	NPM	NAMA	NILAI					
			P	T	UTS	UAS	NA	NH
1	14.1.03.03.0065	Amanda Retno P	80	70	70	50	66	C+
2	14.1.03.03.0030	Nunik Boru Sinulingga	90	70	75	50	69	C+
3	14.1.03.03.0116	Esti Agus Triana	100	80	80	60	78	B
4	14.1.03.03.0156	Winda Pujaning Ayu	90	70	75	50	69	C+
5	14.1.03.03.0099	Galuh Prahadi Gumelar	100	80	80	50	75	B
6	14.1.03.03.0020	Angga Widiyantoro	100	80	80	60	78	B

Gambar 1. Contoh Data Nilai Mahasiswa Dalam Format *Excel*

Gambar 1 menunjukkan contoh format nilai mahasiswa yang telah diisi oleh dosen pengampu setiap matakuliah. Data nilai mahasiswa dalam format *excel* terdapat data kelas, nama matakuliah, nama dosen Pembina, tabel nilai yang berisi no, NPM, nama mahasiswa, nilai presensi(P), nilai tugas(T), nilai ujian tengah semester(UTS), nilai ujian akhir semester(UAS), nilai akhir(NA) yang diperoleh dari perhitungan semua data sesuai kebijakan program studi dan nilai huruf berdasarkan *range* nilai akhir(NA) yang telah dihitung sebelumnya. Langkah berikutnya adalah melakukan proses *extract* dari data nilai mahasiswa dari gambar 1. Hasil proses *extract* ditunjukkan dalam gambar 2.

No.	NPM	Nama mahasiswa	NILAI HURUF	kode_mk
1	14.1.03.03.0065	Amanda Retno P	C+	MKK-2004
2	14.1.03.03.0030	Nunik Boru Sinulingga	C+	MKK-2004
3	14.1.03.03.0116	Esti Agus Triana	B	MKK-2004
4	14.1.03.03.0156	Winda Pujaning Ayu	C+	MKK-2004
5	14.1.03.03.0099	Galuh Prahadi Gumelar	B	MKK-2004
6	14.1.03.03.0020	Angga Widiyantoro	B	MKK-2004

Gambar 2. Hasil Proses Extract Data Nilai Mahasiswa

Gambar 2 menunjukkan hasil proses *extract* dari gambar 1. Hasil proses *extract* data nilai mahasiswa terdapat data no, NPM, nama mahasiswa, nilai huruf dan kode_mk. Proses *extract* tersebut mengambil nilai huruf dari nilai akhir dari setiap nilai mahasiswa dalam matakuliah dan mengambil kode matakuliah (kode_mk) sesuai dengan nama matakuliahnya. Langkah berikutnya adalah melakukan proses *extract* untuk *grade* nilai matakuliah. *Grade* dibutuhkan untuk mengetahui konversi nilai dari setiap matakuliah. Hasil proses *extract grade* nilai ditunjukkan pada gambar 3.

Huruf	Bobot
A	4
B+	3.5
B	3
C+	2.5
C	2
D	1
E	0

Gambar 3. Hasil Proses *Extract Grade* Nilai

Gambar 3 menunjukkan hasil proses *extract grade* nilai. Hasil proses *extract grade* nilai ditampilkan huruf beserta bobotnya. *Range* nilai yang digunakan mulai A dengan bobot 4 sampai dengan E dengan bobot 0. Langkah berikutnya adalah melakukan proses *extract* untuk data matakuliah. Sks dalam data matakuliah dibutuhkan untuk menentukan pembagi dari total *indeks* prestasi mahasiswa. Contoh hasil proses *extract* data matakuliah ditunjukkan pada gambar 4.

KODE_MK	MATA_KULIAH	SKS	semester
MPK - 1003	ke-PGRI-an	2	1
MKK - 2006	Konsep Sistem Informasi	4	1
MKK - 2001	Logika dan Algoritma	4	1
MKK - 2005	Dasar Akuntansi	4	1
MKK - 2002	Aljabar Linear	2	1
MKK - 2015	Statistik	2	1
MKK - 2012	Pengantar Manajemen	2	1
MPK - 1001	Bahasa Inggris	2	2
MPK - 1006	Pancasila & Kewarganegaraan	3	2
MKK - 2016	Sistem Operasi	2	2
MKK - 2013	Sistem Informasi Manajeme	2	2
MKK - 2010	Matematika Diskret	2	2

Gambar 4. Contoh Hasil Proses Extract Data Matakuliah

Gambar 4 menunjukkan contoh hasil proses *extract* data matakuliah. hasil proses *extract* ditampilkan matakuliah ada data kode matakuliah, nama matakuliah, sks dan semester. *Indeks* prestasi kumulatif dihitung dari jumlah sks dikalikan dengan bobot nilai sesuai *grade* kemudian dibagi dengan jumlah sks matakuliah yang telah diambil. Langkah berikutnya adalah proses *extract* data lulusan yang akan dihitung *indeks* prestasi kumulatifnya. Contoh hasil *extract* data lulusan ditunjukkan dalam tabel 5.

npm	nama	Thn_Akademik	tahun_masuk	Ket
07.1.03.03.0002	CINDY SIMBARA	2010/2011	2007	ts-5
07.1.03.03.0003	M. NAJIBULLOH MUZAKI	2010/2011	2007	ts-5
07.1.03.03.0004	RIZKI ARIF FAUZI	2010/2011	2007	ts-5
08.1.03.03.0005	DESYNTA MEGA DEWI	2012/2013	2008	ts-3
08.1.03.03.0007	HENDRA PANJI PERMANA	2012/2013	2008	ts-3
08.1.03.03.0008	INTAN RAMADHAN	2012/2013	2008	ts-3
08.1.03.03.0009	LILIS SETYONINGSIH	2012/2013	2008	ts-3
08.1.03.03.0011	MOH. HAFIDH ALUFIAN	2012/2013	2008	ts-3
08.1.03.03.0013	NURUL HASANAH	2012/2013	2008	ts-3
08.1.03.03.0014	REBECCA MAULINA HANIF	2012/2013	2008	ts-3
08.1.03.03.0015	RIZKY ANDA SWARDANI	2012/2013	2008	ts-3
08.1.03.03.0016	TRI YUDHITA W	2012/2013	2008	ts-3
08.1.03.03.0017	YUNITA EFI LESTARI	2012/2013	2008	ts-3

Gambar 5. Contoh Hasil Extract Data Lulusan

Gambar 5 menunjukkan contoh hasil proses *extract* data lulusan. Hasil proses *extract* data lulusan ditampilkan data NPM, nama mahasiswa, tahun akademik, tahun masuk dan keterangan tahun sekarang(TS).

3.2 Proses Transform

Langkah berikutnya adalah proses *transform*. Proses *transform* dilakukan untuk perubahan dan integrasi data yang telah didapatkan dari proses *extract*. Proses *transform* yang dibutuhkan dalam penelitian ini adalah untuk menghitung *indeks* prestasi kumulatif(IPK). Proses *transform* dilakukan dengan cara mengintegrasikan data lulusan, data matakuliah, *grade* nilai dan nilai mahasiswa yang disimpan dalam tabel *data_nilai*. Contoh hasil proses *transform* IPK ditunjukkan dalam gambar 6

NPM	Nilai_huruf	Kode_MK	Bobot
13.1.03.03.0002	A	MKB-3010	4
13.1.03.03.0002	B+	MKB-3011	3,5
13.1.03.03.0002	B+	MKB-3012	3,5
13.1.03.03.0002	A	MKB-3015	4
13.1.03.03.0002	C+	MKB-3016	2,5
13.1.03.03.0002	A	MKB-3017	4
13.1.03.03.0002	A	MKB-3018	4
13.1.03.03.0002	B+	MKB-3019	3,5
13.1.03.03.0002	B+	MKB-3020	3,5
13.1.03.03.0002	C+	MKB-3027	2,5
13.1.03.03.0002	B	MKB-3029	3
13.1.03.03.0002	B	MKB-3030	3
13.1.03.03.0002	B	MKB-3031	3

Gambar 6. Contoh Hasil Proses Transform IPK

Gambar 6 menunjukkan contoh hasil proses *transform* IPK. Hasil proses *transform* IPK menampilkan data NPM, nilai_huruf, kode_mk(kode matakuliah) dan bobot. Proses menghitung IPK diperlukan jumlah sks sebagai pembagiannya. Langkah berikutnya adalah membuat *function* untuk menghitung jumlah sks lulusan. *Function* menghitung jumlah sks lulusan ditunjukkan pada gambar 7.

```
CREATE DEFINER='root'@'localhost' FUNCTION `jum_sks_mhs`(fnpm VARCHAR(50)) RETURNS int(11)
BEGIN
RETURN(
SELECT SUM(mk.sks)FROM
data_nilai
INNER JOIN mk ON REPLACE(mk.KODE_MK, ' ', '')=REPLACE(data_nilai.Kode_MK, ' ', '')
WHERE NPM=fnpm
GROUP BY NPM
);
END
```

Gambar 7 Function Untuk Menghitung Jumlah Sks Lulusan.

Gambar 7 menunjukkan *function* untuk menghitung jumlah sks lulusan yang digunakan untuk menghitung IPK lulusan. Langkah berikutnya adalah dibuat *function* untuk menghitung IPK. *Function* untuk menghitung IPK ditunjukkan pada gambar 8.

```
CREATE DEFINER='root'@'localhost' FUNCTION `fc_hitung_ipk`(fnpm VARCHAR(255) ) RETURNS double
BEGIN
RETURN(
SELECT data_nilai.NPM,
( SUM(Bobot * SKS) / jum_sks_mhs(NPM) ) AS ipk
FROM data_nilai
INNER JOIN mk ON REPLACE(mk.KODE_MK, ' ', '')=REPLACE(data_nilai.Kode_MK, ' ', '')
WHERE data_nilai.NPM=fnpm
GROUP BY data_nilai.NPM
);
END
```

Gambar 8 Function Untuk Menghitung IPK Lulusan

Gambar 8 menunjukkan *function* untuk menghitung IPK lulusan. Data diambil dari data_nilai (hasil proses *transform*) pada gambar 6. Contoh hasil dari perhitungan IPK lulusan ditunjukkan pada gambar 9

NPM	ipk
12.1.03.03.0382	3.19
12.1.03.03.0358	3.42
12.1.03.03.0321	3.37
12.1.03.03.0390	3.19
12.1.03.03.0279	3.12
12.1.03.03.0044	3.35
12.1.03.03.0394	3.4
12.1.03.03.0024	3.26
12.1.03.03.0213	3.32
12.1.03.03.0273	3.33
12.1.03.03.0207	3.25

Gambar 9 Contoh Hasil Perhitungan IPK Lulusan

Gambar 9 menunjukkan contoh hasil perhitungan IPK. Hasil perhitungan IPK lulusan menampilkan NPM lulusan dan IPK. Langkah berikutnya adalah proses *transform* untuk menampilkan IPK minimum, IPK rata-rata dan IPK maksimum. *Query* untuk menampilkannya ditunjukkan pada gambar 10.

```
SELECT
SUBSTR(NPM,1,2) as Angkatan,
min(ipk) as minimum,
avg(ipk) as Rata2,
max(ipk) as maksimum
from data_ipk
group by SUBSTR(NPM,1,2)
```

Gambar 10. Query IPK Minimum,Rata-Rata dan Maksimum

Gambar 10 menunjukkan *query* untuk menampilkan IPK minimum,rata-rata dan maksimum. Hasil dari *query* pada gambar 10 akan digunakan untuk mengisi IPK lulusan berdasarkan angkatan, IPK minimum, IPK rata-rata dan IPK maksimum.

3.3 Proses Load

Langkah berikutnya adalah proses *load*. Proses *load* dalam penelitian ini dilakukan langkah untuk memasukkan data ke tabel fakta prosentase IPK. *Procedure* untuk memasukkan data ke tabel fakata prosentase IPK ditunjukkan dalam gambar 11.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `prosentase_ipk`(fthn VARCHAR(255) )
BEGIN
INSERT INTO dim_ipk(angkatan,ket_ipk,jumlah_siswa,prosentase)
SELECT fthn,'>3.5',jumlah_mhs(fthn), ((COUNT(*)/jumlah_mhs(fthn)) * 100 ) FROM data_ipk
WHERE (ipk >3.5) AND SUBSTR(NPM,1,2)=fthn;

INSERT INTO dim_ipk(angkatan,ket_ipk,jumlah_siswa,prosentase)
SELECT fthn,'2.75..3.5',jumlah_mhs(fthn), ((COUNT(*)/jumlah_mhs(fthn)) * 100 ) FROM data_ipk
WHERE (ipk >=2.75) AND (ipk <=3.5) AND SUBSTR(NPM,1,2)=fthn;

INSERT INTO dim_ipk(angkatan,ket_ipk,jumlah_siswa,prosentase)
SELECT fthn,'<2.75',jumlah_mhs(fthn),((COUNT(*)/jumlah_mhs(fthn)) * 100 ) FROM data_ipk
WHERE (ipk < 2.75) AND SUBSTR(NPM,1,2)=fthn;
END
```

Gambar 11. *Procedure* Untuk Memasukkan Data Ke Tabel Fakata Prosentase IPK

Gambar 11 menunjukkan *Procedure* untuk memasukkan data ke tabel fakata prosentase IPK. *Procedure* tersebut digunakan untuk pengelompokkan prosentase IPK mulai dari nilai IPK di bawah 2,75 , nilai IPK antara 2,75 sampai 3,5 dan nilai IPK di atas 3,5. Hasil dari *eksekusi procedure* dari pada gambar 11 ditunjukkan pada gambar 12.

angkatan	ket_ipk	jumlah	prosentase
12	<2.75	0	0
12	2.75..3.5	93	90.2913
12	>3.5	10	9.70874
13	<2.75	71	54.1985
13	2.75..3.5	55	41.9847
13	>3.5	5	3.81679
14	<2.75	90	50
14	2.75..3.5	75	41.6667
14	>3.5	15	8.33333
16	<2.75	9	28.125
16	2.75..3.5	20	62.5
16	>3.5	3	9.375

Gambar 12. Hasil Dari *Eksekusi Procedure* Sebagai Proses *Load*

Gambar 12 menunjukkan hasil eksekusi *Procedure* sebagai proses *load* yang digunakan untuk pengelompokkan prosentase IPK . Hasil pengelompokkan IPK menampilkan data angkatan, ket_ipk(pengelompokkan IPK), jml (jumlah lulusan) dan prosentase.

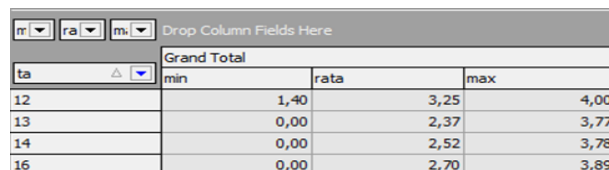
3.4 Aplikasi OLAP

Aplikasi *OLAP* yang dihasilkan menggunakan hasil dari proses *load*. Aplikasi dibuat menggunakan Embarcadero Delphi 10.2 Tokyo dengan *database* MySQL. Tampilan Aplikasi *OLAP* prosentase IPK lulusan ditunjukkan pada gambar 13 dan Aplikasi *OLAP* IPK minimum, rata-rata dan maksimum ditunjukkan pada gambar 14.

angkatan	2.75..3.5	<2.75	>3.5
12	90.29	0.00	9.71
13	41.98	54.20	3.82
14	41.67	50.00	8.33
16	62.50	28.13	9.38

Gambar 13. Aplikasi *OLAP* Prosentase IPK Lulusan

Gambar 13 menunjukkan aplikasi *OLAP* prosentase IPK lulusan berdasarkan *range* IPK yang didapatkan dari hasil proses *load* pada gambar 12. Aplikasi *OLAP* prosentase IPK untuk lulusan menampilkan angkatan, pengelompokkan IPK dan prosentase pada setiap angkatan dan pengelompokkan IPK.



Drop Column Fields Here				
Grand Total				
ta	min	rata	max	
12	1,40	3,25	4,00	
13	0,00	2,37	3,77	
14	0,00	2,52	3,78	
16	0,00	2,70	3,89	

Gambar 14. Aplikasi OLAP IPK Minimum,Rata-Rata Dan Maksimum

Gambar 14 menunjukkan aplikasi OLAP IPK minimum,rata-rata dan maksimum yang didapatkan dari hasil proses load. aplikasi OLAP IPK minimum,rata-rata dan maksimum menunjukkan tahun angkatan(ta), IPK minimum(min), IPK rata-rata(rata) dan IPK maksimum(max).

3.5 Pembahasan

Aplikasi OLAP yang telah dihasilkan dapat digunakan untuk membantu Program Studi Sistem Informasi Fakultas Teknik Universitas Nusantara PGRI Kediri dalam pengisian borang akreditasi program studi untuk standart tiga yang membahas tentang profil mahasiswa dan lulusan. Aplikasi OLAP yang telah dihasilkan adalah aplikasi OLAP prosentase IPK lulusan berdasarkan range IPK dan aplikasi OLAP IPK minimum,rata-rata dan maksimum. Aplikasi OLAP IPK membutuhkan data - data dari Program Studi Sistem Informasi Fakultas Teknik Universitas Nusantara PGRI Kediri yang telah diproses menggunakan ETL.

4. KESIMPULAN

Berdasarkan pembahasan di atas diambil beberapa kesimpulan, yaitu :

- Penerapan metode ETL sangat dibutuhkan dalam pembuatan aplikasi OLAP.
- Proses ETL menggunakan data dari file excel kemudian diolah dan disimpan ke data warehouse OLAP.
- Dengan adanya aplikasi OLAP IPK lulusan ini bisa membantu pihak program studi sistem informasi dalam pengisian IPK lulusan di standart 3 profil mahasiswa dan lulusan.
- Dengan adanya data warehouse suatu perguruan tinggi mempunyai kemudahan untuk mendapatkan informasi yang bersifat analisa untuk data mahasiswa dan lulusan.

UCAPAN TERIMA KASIH

Penelitian ini dapat dilakukan dengan pendanaan penelitian stimulus Yayasan Pembina Lembaga Pendidikan Perguruan Tinggi PGRI Kediri.

DAFTAR PUSTAKA

- [1] M. JRP, 2014. *Pentaho : Solusi Open Source untuk Membangun Data Warehouse*, 1st ed. Yogyakarta: ANDI Yogyakarta.
- [2] V. Rainardi, 2007. *Building Data Warehouse With Examples in SQL Server*. New York: Apress.
- [3] T. M. Connolly and C. E. Begg, 2010. *Database Systems: A Practical Approach to Design, Implementation and Management*, 5th ed. Pearson.
- [4] L. A. Ndoloe, 2012. "Sistem Informasi Lulusan Dengan Metode Online Analytical processing (OLAP) Pada Politeknik Negeri Kupang," *Sist. Inf. Bisnis*, vol. 1, no. 2, p. 84.
- [5] A. Supriyatna, 2016. "SISTEM ANALISIS DATA MAHASISWA MENGGUNAKAN APLIKASI ONLINE ANALYTICAL PROCESSING (OLAP) DATA WAREHOUSE," *Pilar Nusa Mandiri*, vol. XII, no. 1, p. 62.
- [6] S. P. Adithama, I. Wisnubhadra, and B. L. Sinaga, 2013. "ANALISIS DAN DESAIN REAL-TIME BUSINESS INTELLIGENCE UNTUK SUBJEK KEGIATAN AKADEMIK PADA UNIVERSITAS MENGGUNAKAN CHANGE DATA CAPTURE," in *Seminar Nasional Teknologi Informasi dan Komunikasi*.
- [7] M. Y. Pusadan, 2013. *Rancang Bangun Data Warehouse*, 1st ed. Yogyakarta: Graha Ilmu.