

**LAPORAN TUGAS BESAR PERGUDANGAN DATA
UNIT AKADEMIK**



Oleh :

Kelompok 2

- | | |
|---------------------------------|--------------------|
| 1. Abit Ahmad Oktarian | (122450042) |
| 2. Tesalonika Hutajulu | (123450033) |
| 3. Rahmah Gustriana Deka | (123450102) |
| 4. Sarah Wasti | (123450057) |

**PROGRAM STUDI SAINS DATA
FAKULTAS SAINS
INSTITUT TEKNOLOGI SUMATERA
2025**

DAFTAR ISI

DAFTAR ISI.....	i
RINGKASAN.....	1
BAB I PENDAHULUAN.....	2
1.1 Latar Belakang.....	2
1.2 Tujuan Proyek.....	2
1.3 Ruang Lingkup.....	3
1.4 Metodologi.....	3
BAB II ANALISIS KEBUTUHAN.....	5
2.1 Kebutuhan Bisnis.....	5
2.1.1 Identifikasi <i>Stakeholders</i>	5
2.1.2 Proses Bisnis Utama.....	5
2.1.3 Pertanyaan Analitik (<i>Key Analytic Questions</i>).....	6
2.2 Kebutuhan Fungsional.....	7
2.2.1 Manajemen Data Dimensional (<i>Dimensional Data Management</i>).....	7
2.2.2 Manajemen KPI Akademik.....	7
2.2.3 Integrasi Data (<i>ETL & Data Quality</i>).....	8
2.2.4 Pelaporan dan Visualisasi (<i>Reporting</i>).....	8
2.3 Kebutuhan Non-Fungsional.....	8
2.3.1 Performa (<i>Performance</i>).....	8
2.3.2 Keamanan (<i>Security</i>).....	8
2.3.3 Reliabilitas (<i>Reliability</i>).....	9
2.4 Sumber Data.....	9
BAB III PERANCANGAN.....	10
3.1 Model Konseptual (ERD).....	10
3.2 Model Logis (<i>Dimensional Model/Star Schema</i>).....	11
3.2.1 Penentuan <i>Grain</i> (Level Detail).....	12
3.2.2 Identifikasi Tabel Fakta.....	13
3.2.3 Identifikasi Tabel Dimensi.....	13
3.3 Desain Fisik Basis Data.....	14
3.3.1 Penentuan Struktur Tabel.....	14
3.3.2 Perancangan Indeks dan <i>Constraint</i>	15
3.3.3 Pertimbangan Partisi (Jika Diterapkan).....	16
3.4 Diagram Arsitektur.....	17
3.4.1 Alur Pergerakan Data.....	17
3.4.2 Komponen Infrastruktur.....	18
3.4.3 Integrasi Sistem.....	18
BAB IV IMPLEMENTASI.....	20
4.1 Implementasi Basis Data.....	20
4.1.1 Pembuatan Skema <i>Database</i>	20

4.1.2 Pembuatan Tabel Dimensi dan Fakta.....	20
4.1.3 Penerapan Indeks dan <i>Constraint</i>	20
4.2 Proses ETL.....	21
4.2.1 Proses Ekstraksi.....	21
4.2.2 Proses Transformasi.....	21
4.2.3 Proses Pemuatan (Loading).....	22
4.3 Pengembangan Dashboard.....	22
4.3.1 Penyusunan KPI dan Kebutuhan Laporan.....	22
4.3.2 Desain Tampilan Dashboard.....	22
4.4 Implementasi Keamanan.....	22
4.4.1 Pengaturan Hak Akses dan Peran Pengguna.....	23
4.4.2 Audit Trail dan Logging.....	23
BAB V PENGUJIAN DAN VALIDASI.....	24
5.1 Hasil Uji Kualitas Data.....	24
5.2 Pengujian Performa.....	24
5.3 UAT (<i>User Acceptance Test</i>).....	25
5.4 Perbaikan Bug.....	25
BAB VI DEPLOYMENT DAN OPERASIONAL.....	27
6.1 <i>Deployment</i> Produksi.....	27
6.2 Strategi <i>Backup</i>	27
6.3 Monitoring Sistem.....	27
6.4 Prosedur Pemeliharaan.....	27
BAB VII HASIL DAN PEMBAHASAN.....	28
7.1 Temuan Utama.....	28
7.2 <i>Insight</i> Bisnis.....	28
7.3 Metrik Performa.....	28
7.4 Tantangan dan Solusi.....	28
BAB VIII KESIMPULAN DAN SARAN.....	29
8.1 Ringkasan Pencapaian.....	29
8.2 Implikasi dan Manfaat.....	29
8.3 Keterbatasan Proyek.....	29
8.4 Saran Pengembangan Lanjutan.....	29
DAFTAR PUSTAKA.....	30
LAMPIRAN.....	31
Skrip SQL.....	31
<i>Screenshots Dashboard</i>	31
Hasil Uji.....	32

RINGKASAN

Proyek ini membangun sebuah Data Mart Akademik berbasis pendekatan *Dimensional Modeling* (Kimball) untuk menyediakan *single source of truth* bagi seluruh data akademik, mencakup proses Penerimaan Mahasiswa, Perkuliahan, dan Kelulusan. Sistem ini dirancang untuk mendukung analisis strategis dan operasional, memungkinkan pemantauan performa mahasiswa, evaluasi efektivitas program studi, kualitas pengajaran, serta pengambilan keputusan berbasis data oleh pimpinan institusi. Pengembangan dilakukan melalui tiga fase utama: desain konseptual dan logikal (termasuk penyusunan ERD, *Dimensional Model*, dan *Data Dictionary*), pembangunan fisik dan proses ETL (meliputi pembuatan database, pemuatan data dalam skala besar, indexing dan partitioning, serta pengujian performa dan kualitas data), serta implementasi produksi yang mencakup deployment, penjadwalan ETL, pembuatan dashboard analitik Power BI, dan pengamanan data. Hasilnya adalah *Data Mart* yang mampu meningkatkan efisiensi operasional, memberikan wawasan akademik yang lebih mendalam, dan memperkuat pengambilan keputusan strategis. Rekomendasi lanjutan mencakup penerapan SCD Type 2, pengembangan dashboard tingkat eksekutif, dan otomatisasi penuh proses ETL untuk menjaga keberlanjutan sistem.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Proyek ini bertujuan untuk membangun dan mengimplementasikan *Data Mart* Akademik untuk Unit Akademik Institut. *Data Mart* ini dirancang sebagai platform analitik terpusat yang komprehensif, bertujuan untuk mentransformasi data operasional sehari-hari dari sistem sumber (seperti SIAKAD OLTP) menjadi informasi strategis yang siap digunakan untuk pengambilan keputusan.[1]

Fokus utama proyek *Data Mart* Akademik ini mencakup tiga proses bisnis inti dalam siklus hidup mahasiswa: Penerimaan (*Admission*), Perkuliahan (*Enrollment*), dan Kelulusan (*Graduation*). Dengan menggunakan pendekatan *Dimensional Modeling* atau *Star Schema*, proyek ini menghasilkan tiga tabel fakta utama (Fact_Admission, Fact_Enrollment, dan Fact_Graduation) yang didukung oleh enam tabel dimensi terkait.

Hasil akhir dari proyek ini adalah sebuah *data warehouse* mini yang terstruktur, telah melewati tahap desain fisik, *development* ETL, optimasi performa melalui *indexing* dan *partitioning*, hingga siap diimplementasikan dalam lingkungan produksi untuk mendukung visualisasi dan analisis data melalui *tools* seperti Power BI.

1.2 Tujuan Proyek

Tujuan utama dari pembangunan *Data Mart* Akademik ini adalah:

- Menyediakan wawasan akademik yang mendalam: mendukung analitik akademik secara komprehensif, mulai dari pemantauan performa mahasiswa, efektivitas mata kuliah, hingga kualitas pengajaran.
- Mendukung pengambilan keputusan strategis: menyediakan data yang terstruktur dan konsisten bagi *Decision Makers* (termasuk Rektor/Wakil Rektor, Dekan, dan Ketua Program Studi) untuk mengevaluasi *Key Performance Indicators* (KPI) institusi dan program studi.
- Mengukur efisiensi proses bisnis: mampu menjawab 21 pertanyaan analitik (*Use Cases*) yang dirumuskan, termasuk analisis tren penerimaan, tingkat kelulusan tepat waktu, rata-rata IPK, dan indikator *drop-out*.
- Mencapai kualitas data analitik tinggi: memastikan integritas data melalui implementasi *Data Quality Checks* dan penerapan *Business Rules* yang ketat, serta mengoptimalkan performa kueri analitik.

1.3 Ruang Lingkup

Ruang lingkup proyek *Data Mart Akademik* ini mencakup aspek-aspek berikut:

Aspek	Deskripsi Ruang Lingkup
Domain Bisnis	Pengelolaan akademik yang mencakup Pendaftaran masuk, Aktivitas perkuliahan, dan Kelulusan mahasiswa.
Sumber Data (<i>Source</i>)	Data utama berasal dari Sistem Informasi Akademik (SIKAD OLTP) dan didukung oleh sumber data lain seperti Insightera, Website Fakultas, dan data Manual.
Model Data	Implementasi <i>Dimensional Modeling</i> (Star Schema) dengan fokus pada tiga area fakta: <i>Enrollment</i> (butir: satu baris per mahasiswa per mata kuliah per semester), <i>Admission</i> (butir: satu baris per pendaftar), dan <i>Graduation</i> (butir: satu baris per mahasiswa yang lulus).[1]
Implementasi Fisik	Mencakup <i>Database Setup</i> (SQL Server), pembuatan semua <i>Dimension</i> dan <i>Fact Tables</i> dengan <i>referential integrity</i> (Foreign Key).
Optimasi	Penerapan <i>Indexing Strategy</i> (Non-Clustered Indexes) dan <i>Partitioning</i> pada <i>Fact Tables</i> (berbasis tahun akademik) untuk meningkatkan performa analitik.
ETL	Desain dan implementasi alur ETL (Extract, Transform, Load) menggunakan <i>T-SQL Stored Procedures</i> dan <i>Staging Area</i> untuk memuat data dari sumber ke <i>Data Mart</i> .[3]

1.4 Metodologi

Proyek ini dilaksanakan melalui serangkaian tahapan yang terstruktur, dibagi menjadi tiga misi utama:

Misi 1: Desain Konseptual dan Logikal

1. Analisis Kebutuhan Bisnis (Step 1): Identifikasi *Stakeholders*, analisis *Proses Bisnis*, penetapan *KPIs*, dan perumusan *Analytical Questions*.
2. Identifikasi Sumber Data (Step 2): Analisis sumber dan *Data Profiling*.
3. Conceptual Design (Step 3): Perancangan *Entity-Relationship Diagram* (ERD).

4. Logical Design (Step 4): Perancangan *Dimensional Model* (Star Schema) dan definisi *Fact* dan *Dimension Tables*.

Misi 2: Desain Fisikal dan Development

1. Physical Database Design (Step 1): Pembuatan *Dimension Tables* dan *Fact Tables* di SQL Server, termasuk *Data Generation*.
2. Indexing Strategy (Step 2): Penerapan *Clustered* dan *Non-Clustered Indexes*.
3. Partitioning Strategy (Step 3): Implementasi fungsi dan skema *Partitioning* untuk *Fact Table*.
4. ETL Design (Step 4): Perancangan *Staging Tables* dan *ETL Architecture*.
5. ETL Implementation (Step 5): Pembuatan *Stored Procedures* untuk proses *Load Dimensions* dan *Load Facts*.
6. Data Quality Assurance (Step 6): Pengecekan konsistensi, kelengkapan, dan akurasi data.
7. Performance Testing (Step 7): Pengujian kinerja Data Mart menggunakan *Analytical Queries*.

Misi 3: Implementasi Produksi

1. Production Deployment (Step 1): *Environment Setup*, *Database Deployment*, *Initial Data Load*, dan *Schedule ETL Jobs*.
2. Dashboard Development (Step 2): Pembuatan *Analytical Views* dan perancangan Dashboard Power BI.
3. Security Implementation (Step 3): Penetapan *User Roles*, *Users*, dan *Data Masking*.
4. Backup and Recovery Strategy (Step 4).
5. User Acceptance Testing (UAT) (Step 5).
6. Documentation (Step 6): Penyusunan dokumen arsitektur, ETL, dan manual pengguna.
7. Final Presentation (Step 7).

BAB II

ANALISIS KEBUTUHAN

2.1 Kebutuhan Bisnis

Analisis kebutuhan bisnis dilakukan untuk menyelaraskan solusi teknis *Data Mart* dengan tujuan strategis institusi pendidikan. Tahap ini bertujuan mengidentifikasi masalah bisnis, menentukan pemangku kepentingan (*stakeholders*), serta memetakan area keputusan krusial yang memerlukan dukungan data akurat (*data-driven decision making*). Analisis kebutuhan bisnis dilakukan untuk mengidentifikasi pemangku kepentingan utama dan area keputusan strategis yang memerlukan dukungan data.

2.1.1 Identifikasi Stakeholders

Identifikasi pemangku kepentingan (*stakeholders*) dilakukan untuk memetakan siapa saja pengguna sistem dan jenis informasi spesifik yang mereka butuhkan. Dalam ekosistem akademik ini, pengguna diklasifikasikan ke dalam tiga tingkatan manajemen, yaitu:

a. Eksekutif (*Strategic Level*)

Kelompok ini terdiri dari Rektor, Wakil Rektor Bidang Akademik, dan Dekan Fakultas. Sebagai pengambil keputusan strategis, mereka membutuhkan laporan bersifat agregat dan ringkas (*executive summary*) untuk memantau kesehatan institusi secara keseluruhan. Kebutuhan informasi mereka berfokus pada tren jangka panjang, seperti tren pertumbuhan jumlah mahasiswa baru, tingkat retensi institusi, dan pencapaian target kelulusan tahunan untuk keperluan akreditasi dan perencanaan strategis universitas.

b. Manajemen Unit (*Tactical Level*)

Kelompok ini meliputi Kepala Unit Akademik, Kepala Program Studi (Kaprodi), dan Kepala Biro Akademik. Mereka berperan sebagai pengelola taktis yang membutuhkan data lebih rinci untuk mengevaluasi efektivitas operasional di unit masing-masing. Kebutuhan utama mereka meliputi pemantauan rata-rata Indeks Prestasi (IPK) mahasiswa per semester, analisis beban kerja dosen, evaluasi kurikulum berdasarkan tingkat kelulusan mata kuliah, serta pemantauan rasio dosen dan mahasiswa di tingkat program studi.

c. Operasional (*Operational Level*)

Kelompok ini mencakup Staf Administrasi Akademik dan Unit Penerimaan Mahasiswa Baru (PMB). Mereka merupakan pengguna yang berinteraksi langsung dengan data harian dan membutuhkan akses detail untuk keperluan pelaporan rutin serta validasi data. Kebutuhan informasi mereka meliputi rekapitulasi data pendaftar harian, status pembayaran registrasi ulang mahasiswa, serta verifikasi data prasyarat yudisium dan wisuda secara akurat dan efisien.

2.1.2 Proses Bisnis Utama

Sistem *Data Mart* ini dikembangkan untuk mengelola data di setiap tahapan masa studi mahasiswa (*Student Lifecycle*). Integrasi data difokuskan pada empat area operasional utama berikut ini:

a. Penerimaan Mahasiswa Baru (*Admission*)

Mencakup rangkaian proses awal masuk perguruan tinggi, dimulai dari pendaftaran calon mahasiswa, pelaksanaan seleksi (baik jalur tes maupun prestasi), hingga tahap registrasi ulang bagi calon mahasiswa yang dinyatakan lulus.

b. Perkuliahan & Pengajaran

Mengelola data aktivitas akademik harian, meliputi penyusunan jadwal mata kuliah, rekapitulasi kehadiran (*presensi*) dosen dan mahasiswa, serta pelaksanaan kegiatan belajar mengajar di kelas.

c. Evaluasi Akademik

Berfokus pada pemantauan prestasi akademik mahasiswa, mencakup proses pemberian nilai akhir semester, perhitungan Indeks Prestasi (IPS dan IPK), serta pembaruan riwayat transkrip nilai secara berkala.

d. Kelulusan (*Graduation*)

Menangani fase akhir studi mahasiswa, mulai dari administrasi pengerjaan Tugas Akhir/Skripsi, penetapan status kelulusan melalui yudisium, hingga pelaksanaan wisuda.

e. Mutasi/Drop-out/Cuti Akademik

Menangani perubahan status studi mahasiswa selama masa perkuliahan, mencakup proses perpindahan program studi (mutasi), penghentian studi karena tidak memenuhi ketentuan akademik atau administratif (*drop-out*), serta pengajuan izin tidak aktif sementara dalam periode tertentu (cuti akademik). Setiap proses diatur melalui prosedur resmi untuk memastikan data dan status akademik mahasiswa tercatat dengan benar.

2.1.3 Pertanyaan Analitik (*Key Analytic Questions*)

Kebutuhan bisnis diterjemahkan ke dalam pertanyaan analitik utama, antara lain:

a. Penerimaan & Registrasi Mahasiswa

1. Bagaimana tren penerimaan mahasiswa dalam 5 tahun terakhir per program studi?
2. Berapa persentase pendaftar yang melakukan registrasi ulang (yield rate)?
3. Jalur penerimaan mana yang paling efektif (SNBP, SNBT, Mandiri)?
4. Program studi mana yang mengalami peningkatan/penurunan pendaftar paling signifikan?

b. Kinerja Akademik

1. Program studi mana yang memiliki rata-rata IPK tertinggi/terendah?
2. Bagaimana distribusi nilai dan IPK per semester?
3. Bagaimana korelasi antara kehadiran dan nilai akhir?
4. Mata kuliah mana yang paling banyak gagal?
5. Apakah terdapat pola performa akademik untuk mahasiswa berprestasi atau mahasiswa bermasalah?

c. Retensi & Drop-out

1. Indikator apa yang memprediksi drop-out?
2. Berapa tingkat drop-out per semester?
3. Pada semester ke berapa drop-out paling banyak terjadi?
4. Apa alasan utama mahasiswa drop-out?

d. Kelulusan

1. Berapa persen mahasiswa yang lulus tepat waktu (4 tahun)?
2. Apa tren kelulusan per program studi selama 5 tahun terakhir?
3. Berapa rata-rata lama studi per angkatan?
4. Berapa mahasiswa yang lulus dengan predikat cum laude per tahun?

e. Monitoring Operasional

1. Berapa jumlah mahasiswa aktif per semester?
2. Tren pengambilan SKS per semester.
3. Tingkat kehadiran mahasiswa per kelas dan per dosen.
4. Monitoring keterlambatan registrasi.

2.2 Kebutuhan Fungsional

Sistem *Data Mart* harus menyediakan serangkaian kapabilitas teknis untuk mendukung seluruh rantai pemrosesan data, mulai dari penyimpanan, integrasi, hingga penyajian informasi.

2.2.1 Manajemen Data Dimensional (*Dimensional Data Management*)

Sistem harus mampu menyimpan dan mengelola data historis dalam struktur Skema Bintang yang terorganisir. Hal ini mencakup pengelolaan tabel fakta untuk transaksi akademik (*Fact Enrollment*, *Fact Admission*, *Fact Graduation*) serta tabel dimensi pendukung (*Dim Student*, *Dim Course*, *Dim Instructor*, *Dim Date*) untuk memungkinkan analisis *drill-down* dan *slicing-dicing* data .

2.2.2 Manajemen KPI Akademik

Sistem harus memiliki logika komputasi untuk menghitung metrik kinerja utama (*Key Performance Indicators*) secara otomatis dan akurat, yang meliputi:

- a. KPI Tingkat Institusi
 - Jumlah mahasiswa aktif per tahun akademik
 - Tren penerimaan mahasiswa baru per tahun
 - Drop-out rate per angkatan
- b. KPI Tingkat Program Studi
 - Rata-rata IPK per semester dan angkatan
 - Tingkat kelulusan tepat waktu
 - Rasio dosen:mahasiswa per prodi
 - Persentase mahasiswa yang mengambil beban studi minimal/optimal

c. KPI Operasional Akademik

- Kehadiran dosen dan mahasiswa
- Distribusi nilai tiap mata kuliah (A–E)
- Jumlah mahasiswa yang mengulang mata kuliah
- Waktu tempuh studi (lama studi)

2.2.3 Integrasi Data (ETL & Data Quality)

Sistem harus menyediakan mekanisme *Extract, Transform, Load* (ETL) yang handal untuk:

- Melakukan sinkronisasi data dari *staging area* ke tabel dimensi dan fakta menggunakan *Stored Procedure* .
- Mendukung mekanisme *Merge* untuk menangani pembaruan data dimensi (*Slowly Changing Dimension*) .
- Menjalankan pengecekan kualitas data (*Data Quality Checks*) otomatis, seperti validasi kelengkapan data (*completeness*), konsistensi referensi (*orphan checks*), dan validitas nilai numerik .

2.2.4 Pelaporan dan Visualisasi (*Reporting*)

Sistem harus menyediakan (*Dashboard*) yang interaktif untuk berbagai tingkatan pengguna:

- Executive Summary: Menyajikan ringkasan makro seperti total mahasiswa aktif, rata-rata IPK institusi, dan tren total *enrollment* .
- Academic Performance Dashboard: Memvisualisasikan distribusi nilai, tingkat kelulusan mata kuliah (*pass rate*), dan daftar mahasiswa berprestasi .
- Financial & Admission Dashboard: Menyajikan analisis corong penerimaan (*Funnel Admission*) dan kontribusi program studi terhadap kelulusan .

2.3 Kebutuhan Non-Fungsional

Untuk menjamin kualitas operasional sistem, kebutuhan non-fungsional berikut diterapkan:

2.3.1 Performa (*Performance*)

- Implementasi strategi indeks (*Non-Clustered Index* dan *Columnstore Index*) untuk mempercepat kueri analitik pada tabel fakta besar seperti Fact_Enrollment.
- Penerapan partisi tabel (*Partitioning Strategy*) berdasarkan Tahun Akademik (PF_AcademicYear) untuk manajemen data historis yang efisien .

2.3.2 Keamanan (*Security*)

- Role-Based Access Control (RBAC): Pembagian hak akses menjadi db_executive, db_analyst, db_viewer, dan db_etl_operator .
- Dynamic Data Masking: Perlindungan data sensitif (Email dan Nomor Telepon) pada tabel Dim_Student agar tidak terlihat oleh pengguna yang tidak berwenang .

- Audit Trail: Pencatatan aktivitas *INSERT*, *UPDATE*, dan *DELETE* pada tabel fakta untuk keperluan audit .

2.3.3 Reliabilitas (Reliability)

- Strategi pencadangan rutin meliputi *Full Backup* mingguan, *Differential Backup* harian, dan *Transaction Log Backup* setiap 6 jam.

2.4 Sumber Data

Sumber data utama berasal dari Staging Area yang berisi data akademik sintetis mencakup:

Tabel 2.1 Sumber Data

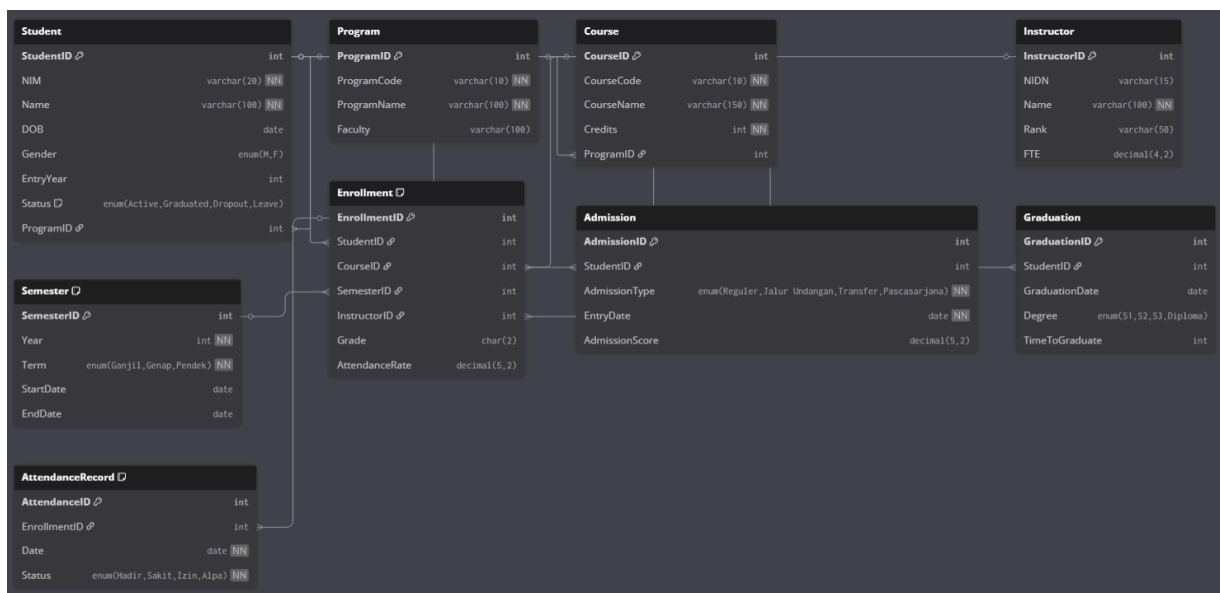
Sumber	Tipe	Contoh Struktur	Volume (perkiraan)	Frekuensi Update
SIAKAD (OLTP)	SQL Server	tabel: Students, Enrollments, Courses, Grades	100k rows	Real-time / harian
Insightera	API/DB	Pendaftar, nilai seleksi	Sedang	Real-time (Tertentu)
Fakultas Website	Website	Catata khusus	Sedang	Bulanan
Manual Excel	Excel	Catatan khusus (Manual Data Sintesis)	kecil	Sporadis

Semua file *staging* diproses melalui ETL sebelum masuk ke Data Warehouse.

BAB III PERANCANGAN

3.1 Model Konseptual (ERD)

Pada tahap ini disusun *Entity Relationship Diagram (ERD)* sebagai model konseptual yang menggambarkan struktur utama data akademik yang akan dikelola. ERD mendefinisikan entitas inti seperti mahasiswa, dosen, program studi, mata kuliah, kelas, semester, dan nilai, beserta atribut serta hubungan antar entitas. Penyusunan ERD dilakukan untuk memastikan bahwa seluruh kebutuhan informasi dari Unit Akademik tercakup secara menyeluruh. Berikut ini ERD yang dibangun.

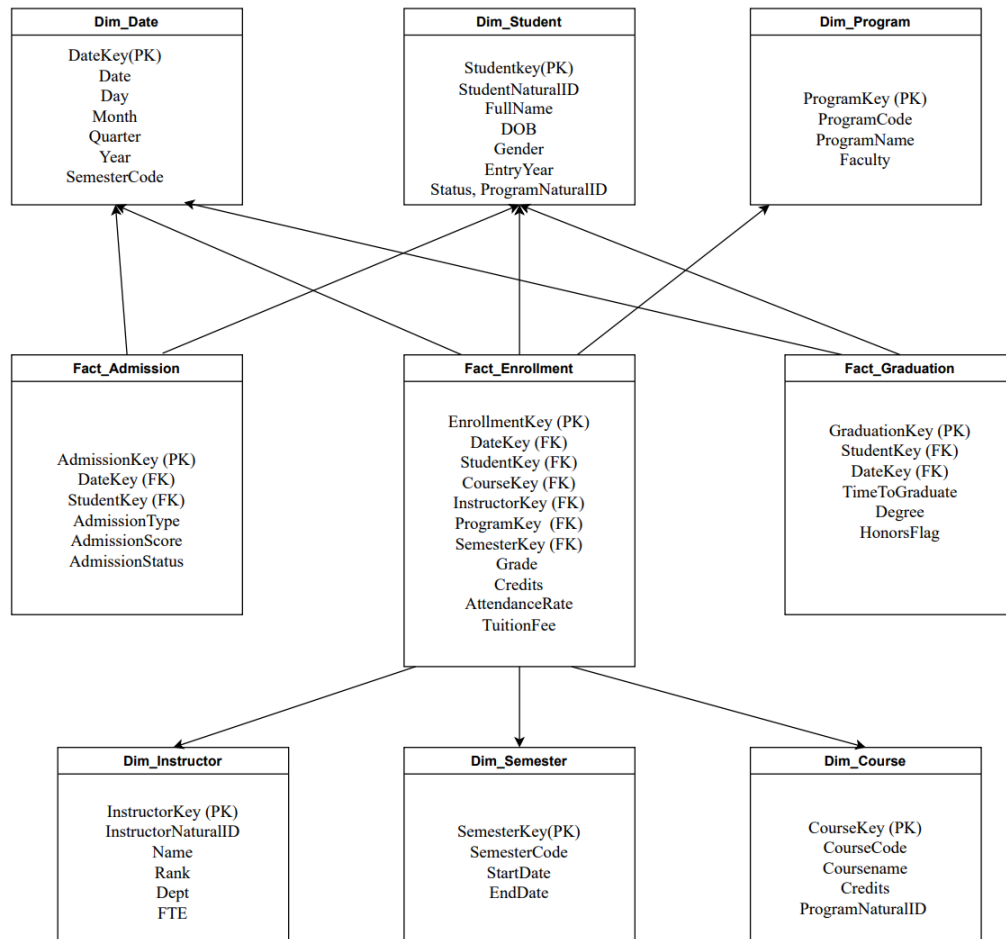


Gambar 3.1 ERD Unit Akademik

Diagram tersebut menggambarkan struktur basis data akademik yang saling terhubung untuk mengelola informasi mahasiswa, program studi, mata kuliah, dosen, serta aktivitas akademik mereka. Entitas *Student* menyimpan data pribadi dan status mahasiswa, yang terhubung ke *Program*, *Admission*, *Enrollment*, dan *Graduation*. Setiap mahasiswa dapat mengikuti beberapa mata kuliah melalui entitas *Enrollment*, yang juga mengaitkan dosen (*Instructor*) dan semester (*Semester*). Kehadiran mahasiswa dicatat dalam *AttendanceRecord*. Sementara itu, *Course* terhubung ke *Program* dan menjadi bagian dari proses pengambilan mata kuliah. Dengan hubungan-hubungan tersebut, sistem ini mampu melacak perjalanan akademik mahasiswa dari saat masuk (*Admission*), menjalani perkuliahan (*Enrollment* dan *AttendanceRecord*), hingga lulus (*Graduation*).

3.2 Model Logis (Dimensional Model/Star Schema)

Model logis dirancang menggunakan pendekatan *dimensional modeling* dengan arsitektur *star schema* untuk mendukung kebutuhan analitik akademik. Pada tahap ini ditetapkan tabel fakta dan tabel dimensi yang merepresentasikan proses bisnis inti seperti penerimaan mahasiswa, aktivitas perkuliahan, dan kelulusan. Berikut ini adalah gambar dari star schema.



Gambar 3.2 Star Schema unit akademik

Diagram tersebut menunjukkan rancangan *data warehouse* dengan skema *star schema* yang terdiri dari tabel fakta dan tabel dimensi untuk mendukung analisis data akademik. Tabel fakta—*Fact_Admission*, *Fact_Enrollment*, dan *Fact_Graduation*—menyimpan peristiwa utama seperti penerimaan mahasiswa, pengambilan mata kuliah, dan kelulusan, masing-masing terhubung ke berbagai tabel dimensi. Tabel dimensi seperti *Dim_Student*, *Dim_Program*, *Dim_Instructor*, *Dim_Semester*, *Dim_Course*, dan *Dim_Date* menyediakan konteks detail yang digunakan untuk melakukan analisis berdasarkan karakteristik mahasiswa, program studi, dosen, semester, mata kuliah, maupun waktu. Melalui hubungan-hubungan ini, sistem mampu memberikan gambaran komprehensif mengenai perjalanan akademik mahasiswa mulai dari penerimaan, proses pembelajaran, hingga kelulusan, serta memungkinkan pelaporan dan analisis strategis secara lebih fleksibel dan terstruktur.

3.2.1 Penentuan Grain (Level Detail)

Penentuan *grain* (butiran data) merupakan langkah fundamental dalam perancangan model dimensional untuk memastikan *Data Mart* mampu menyimpan data pada level detail terendah yang atomik. Dalam proyek ini, *grain* ditetapkan pada level transaksi individu untuk menjamin fleksibilitas analisis yang maksimal.

a. Grain Tabel Fakta

1. Fact_Enrollment

Tabel ini dirancang untuk menyimpan data pada tingkat transaksi paling mendasar, yaitu setiap satu kali seorang mahasiswa mengambil satu mata kuliah di semester tertentu. Pendekatan ini memungkinkan sistem untuk mencatat detail performa akademik secara presisi, mulai dari nilai numerik, persentase kehadiran, hingga status kelulusan mata kuliah tersebut bagi setiap individu.

2. Fact_Admission

Tabel ini mencatat setiap satu aplikasi pendaftaran yang diajukan oleh calon mahasiswa. Tingkat detail ini mencakup seluruh riwayat pendaftar, mulai dari jalur seleksi yang dipilih (seperti SNBP, SNBT, atau Mandiri), skor hasil tes, hingga keputusan akhir apakah pendaftar tersebut diterima atau ditolak oleh program studi tujuan.

3. Fact_Graduation

Tabel ini difokuskan untuk merekam peristiwa kelulusan, di mana satu baris data mewakili satu mahasiswa yang telah menyelesaikan masa studinya secara resmi. Pada level ini, sistem dapat menyimpan atribut pencapaian akhir yang krusial dan unik per individu, seperti total SKS yang ditempuh, IPK final, durasi studi, serta predikat kelulusan (*honors*) yang diraih.

b. Dampak Grain Terhadap Kualitas Analisis

Pemilihan *grain* pada level atomik (individu) memberikan dampak signifikan terhadap kualitas dan kedalaman analisis yang dapat dilakukan oleh sistem:

1. Drill-Down

Pengguna eksekutif dapat melihat laporan agregat (misalnya: Rata-rata IPK Universitas), lalu melakukan *drill-down* hingga ke level Fakultas, Program Studi, hingga melihat transkrip nilai individu mahasiswa yang bermasalah.

2. Roll-Up

Data detail memungkinkan sistem untuk melakukan agregasi ke segala dimensi tanpa batasan. Analisis dapat dilakukan berdasarkan dimensi waktu (semester/tahun), dimensi wilayah (fakultas/prodi), atau dimensi demografi (angkatan/jalur masuk) tanpa perlu merancang ulang tabel fakta.

3.2.2 Identifikasi Tabel Fakta

. Berdasarkan analisis kebutuhan, terpilih tiga tabel fakta utama yang merepresentasikan perjalanan siklus hidup mahasiswa (*student lifecycle*), mulai dari pendaftaran hingga kelulusan.

a. Fact_Admission (Fakta Penerimaan)

Tabel ini dipilih untuk merepresentasikan fase awal masuknya mahasiswa ke dalam ekosistem institusi. Fokus utamanya adalah mengukur efektivitas proses seleksi dan kualitas calon mahasiswa yang mendaftar. Atribut numerik (*measures*) yang digunakan dalam tabel ini meliputi TestScore dan InterviewScore untuk mengevaluasi kualitas input akademik calon mahasiswa, HighSchoolGPA sebagai indikator prestasi sebelumnya, serta ProcessingDays untuk mengukur efisiensi durasi waktu yang dibutuhkan staf dalam memproses satu aplikasi pendaftaran .

b. Fact_Enrollment (Fakta Perkuliahan)

Sebagai inti dari kegiatan operasional akademik, tabel ini dipilih untuk mencatat transaksi pengambilan mata kuliah yang terjadi setiap semester. Fakta ini krusial untuk memantau performa belajar mengajar secara berkelanjutan. *Measures* yang disimpan mencakup NumericGrade (nilai angka) untuk perhitungan IPK, Credits (SKS) untuk memantau beban studi, dan AttendanceRate (persentase kehadiran) yang berfungsi sebagai indikator kedisiplinan dan deteksi dini risiko mahasiswa bermasalah

c. Fact_Graduation (Fakta Kelulusan)

Tabel ini ditetapkan untuk mengukur *output* atau hasil akhir dari proses pendidikan. Pemilihannya didasari oleh kebutuhan manajemen untuk mengevaluasi keberhasilan studi dan kepatuhan terhadap masa studi ideal. Atribut ukur yang digunakan meliputi GPA (IPK Akhir) sebagai indikator kualitas lulusan, TotalCredits untuk memastikan pemenuhan syarat kelulusan, ThesisScore untuk menilai kualitas tugas akhir, serta StudyDuration (lama studi dalam bulan) untuk menghitung tingkat kelulusan tepat waktu

3.2.3 Identifikasi Tabel Dimensi

Tabel dimensi berfungsi sebagai penyedia konteks deskriptif untuk pengukuran numerik yang tersimpan dalam tabel fakta. Terdapat enam dimensi utama yang memungkinkan analisis data akademik dilakukan dari berbagai perspektif.

a. Dim_Student (Dimensi Mahasiswa)

Atribut yang terdapat dalam dimensi ini meliputi StudentNaturalID (NIM), FullName, Gender, dan EntryYear (Tahun Angkatan). Peran utama dimensi ini adalah mendukung analisis demografi, pemantauan performa berdasarkan angkatan (*cohort analysis*), serta evaluasi tingkat keberhasilan studi berdasarkan karakteristik mahasiswa.

b. Dim_Program (Dimensi Program Studi)

Atribut kunci pada dimensi ini adalah ProgramName dan Faculty . Dimensi ini berperan krusial dalam analisis komparatif, memungkinkan eksekutif membandingkan kinerja akademik, jumlah pendaftar, dan tingkat kelulusan antar program studi maupun antar fakultas.

c. Dim_Course (Dimensi Mata Kuliah)

Dimensi ini berisi detail kurikulum yang ditawarkan institusi. Atribut utamanya mencakup CourseCode, CourseName, dan Credits (SKS). Dalam analisis, dimensi ini digunakan untuk mengevaluasi tingkat kesulitan mata kuliah (berdasarkan distribusi nilai), mengukur beban studi mahasiswa, serta mengidentifikasi mata kuliah yang memiliki tingkat kegagalan tinggi.

d. Dim_Instructor (Dimensi Pengajar)

Atribut pentingnya meliputi Name, Rank (Jabatan Akademik), dan FTE (*Full-Time Equivalent*) . Dimensi ini memungkinkan analisis beban kerja dosen, evaluasi kinerja pengajaran berdasarkan rata-rata nilai kelas, serta pemetaan rasio dosen terhadap mahasiswa.

e. Dim_Semester (Dimensi Periode Akademik)

Dimensi ini secara spesifik memetakan kalender akademik institusi. Atributnya terdiri dari SemesterCode, StartDate, dan EndDate . Perannya untuk memisahkan analisis berdasarkan periode studi (Semester Ganjil/Genap) dan memantau tren perkembangan akademik dari satu semester ke semester berikutnya.

f. Dim_Date (Dimensi Waktu Kalender)

Dimensi ini menyediakan hierarki waktu kalender. Atributnya mencakup Date, Month, Quarter, dan Year . Dimensi ini digunakan untuk analisis yang berbasis waktu harian atau bulanan, seperti memantau tren pendaftaran harian saat masa penerimaan mahasiswa baru atau durasi pemrosesan administrasi.

3.3 Desain Fisik Basis Data

Desain fisik merupakan tahap penerjemahan model logis ke dalam struktur basis data yang akan diimplementasikan pada sistem manajemen basis data yang digunakan. Pada tahap ini ditentukan tipe data setiap kolom, pemilihan *primary key* dan *surrogate key*, pembuatan indeks, pembatasan (*constraints*), serta pengaturan partisi jika diperlukan.

3.3.1 Penentuan Struktur Tabel

Penentuan struktur tabel merupakan langkah krusial dalam pembangunan fisik *Data Warehouse*. Desain kolom tidak hanya bertujuan untuk menampung data, tetapi juga menjaga kualitas dan validitas informasi tersebut. Seluruh tabel dimensi dan fakta dirancang menggunakan prinsip *dimensional modeling* dengan penerapan batasan (*constraints*) yang ketat. Hal ini mencakup aturan NOT NULL pada kolom-kolom penting agar data tidak boleh kosong, serta penggunaan *Check Constraints* untuk memvalidasi logika bisnis secara

otomatis. Sebagai contoh, nilai NumericGrade dibatasi hanya dalam rentang 0 hingga 4, dan kolom Gender dipastikan hanya menerima karakter 'M' atau 'F'.

Sejalan dengan desain tersebut, strategi penguncian (*key strategy*) diterapkan untuk memisahkan ketergantungan antara gudang data dan sistem operasional. Setiap tabel menggunakan Surrogate Key (kunci pengganti) bertipe INT dengan penomoran otomatis (IDENTITY) sebagai *Primary Key*. Penggunaan kunci buatan ini, seperti StudentKey, bertujuan melindungi struktur gudang data dari perubahan data di sistem sumber sekaligus mempercepat proses penggabungan (*join*) antar tabel. Meskipun demikian, kunci asli (*Natural Key*) seperti NIM atau NIDN tetap disimpan dengan akhiran nama NaturalID untuk keperluan pelacakan data (*traceability*) saat proses integrasi.

Pemilihan tipe data disesuaikan untuk menyeimbangkan efisiensi penyimpanan dan kecepatan akses. Seluruh kunci relasi dan atribut tanggal (DateKey) disimpan menggunakan tipe data INT (format YYYYMMDD). Pendekatan ini dipilih karena pemrosesan data angka jauh lebih cepat dibandingkan format tanggal atau teks, serta mendukung strategi partisi tabel yang lebih baik. Untuk data pengukuran seperti GPA, digunakan tipe DECIMAL guna menjamin akurasi hitungan tanpa kesalahan pembulatan. Sementara itu, atribut dengan kode tetap seperti Gender menggunakan tipe CHAR(1) karena lebih hemat ruang penyimpanan dibandingkan tipe variabel (VARCHAR).

3.3.2 Perancangan Indeks dan Constraint

Untuk menjamin kinerja sistem yang optimal dan validitas data yang tinggi, perancangan fisik basis data dilengkapi dengan strategi pengindeksan (*indexing*) dan penerapan batasan (*constraints*) yang bertujuan untuk menyeimbangkan kebutuhan kecepatan akses data untuk keperluan analitik dengan kebutuhan menjaga integritas data agar tetap konsisten dan akurat.

a. Jenis Indeks yang Digunakan

1. Non-Clustered Index

Jenis indeks ini diterapkan secara ekstensif pada kolom-kolom *Foreign Key* dan kolom yang sering digunakan dalam klausa JOIN atau filter pencarian, seperti StudentKey dan DateKey. Tujuannya adalah untuk mempercepat proses pencarian data spesifik tanpa memindai seluruh tabel (*table scan*).

2. Columnstore Index

Khusus untuk tabel fakta dengan volume data besar seperti Fact_Enrollment, digunakan *Non-Clustered Columnstore Index*. Teknologi ini menyimpan data secara kolumnar yang sangat efisien untuk operasi agregasi dan analitik data besar, sehingga performa *query* pelaporan menjadi jauh lebih cepat dibandingkan indeks baris tradisional.

3. Clustered Index

Secara *default*, indeks ini diterapkan pada *Primary Key* setiap tabel untuk mengatur urutan penyimpanan fisik data, memastikan akses data baris tunggal tetap efisien .

b. Tujuan Penambahan Constraint

1. Menjaga Integritas Referensial

Penggunaan *Primary Key* dan *Foreign Key* memastikan bahwa setiap transaksi pada tabel fakta (seperti pendaftaran atau nilai) selalu terhubung dengan data dimensi yang valid (mahasiswa, dosen, atau mata kuliah yang terdaftar) .

2. Validasi Logika Bisnis Otomatis

Check Constraint diterapkan untuk mencegah masuknya data yang tidak logis atau di luar aturan. Contoh penerapannya meliputi pembatasan nilai *NumericGrade* agar selalu berada di rentang 0 hingga 4, serta memastikan persentase kehadiran (*AttendanceRate*) berada di antara 0 hingga 100 .

3. Mencegah Duplikasi

Kombinasi strategi pengindeksan dan penerapan *constraint* memberikan dampak signifikan terhadap operasional sistem *Data Warehouse*, terutama dalam aspek kecepatan dan keandalan data. Penggunaan indeks yang tepat, khususnya *Columnstore*, terbukti mampu mengakselerasi waktu respon dengan memproses jutaan baris data transaksi dalam hitungan detik, yang secara langsung mendukung terciptanya *dashboard* yang responsif . Selain itu, efisiensi penyimpanan dan I/O juga meningkat berkat pemilihan tipe data yang presisi, sehingga beban *disk* berkurang dan *server* mampu menangani lebih banyak permintaan analisis secara bersamaan. Terakhir, penerapan *constraint* yang ketat menjamin konsistensi data dengan meminimalisir risiko kesalahan laporan akibat masuknya data yang tidak valid (*garbage in, garbage out*) sejak awal proses pemuatan data .

3.3.3 Pertimbangan Partisi (Jika Diterapkan)

Strategi partisi (*partitioning strategy*) diterapkan secara selektif pada tabel fakta dengan volume data terbesar, khususnya tabel *Fact_Enrollment*, guna mengantisipasi pertumbuhan data historis jangka panjang. Keputusan ini didasari oleh kebutuhan untuk menjaga stabilitas performa sistem di tengah akumulasi jutaan baris data transaksi pengambilan mata kuliah seiring berjalannya waktu. Tanpa mekanisme ini, setiap kueri analitik berisiko memindai keseluruhan tabel (*table scan*) yang membebani sumber daya I/O secara berlebihan. Oleh karena itu, partisi diperlukan untuk memecah tabel fisik besar menjadi unit-unit logis yang lebih kecil sehingga sistem dapat melakukan penskalaan secara efisien (*scale efficiently*) .

Metode teknis yang digunakan dalam perancangan ini adalah Range Partitioning (Partisi Berdasarkan Rentang) yang berfokus pada kolom *DateKey* sebagai kunci partisi. Implementasinya menggunakan fungsi partisi bernama *PF_AcademicYear*, yang menetapkan batas-batas spesifik berdasarkan awal tahun akademik baru, yaitu setiap tanggal 1 Agustus

(misalnya batas nilai 20200801, 20210801, dan seterusnya). Skema partisi PS_AcademicYear kemudian digunakan untuk memetakan setiap baris data transaksi ke dalam partisi yang relevan secara otomatis berdasarkan tahun ajarannya .

Penerapan strategi ini memberikan dampak ganda yang signifikan bagi operasional *Data Warehouse*. Dari sisi performa, mekanisme ini mengaktifkan kemampuan *Partition Elimination*, di mana mesin basis data secara cerdas mengabaikan partisi yang tidak relevan saat mengeksekusi kueri berbasis waktu, sehingga waktu respon laporan menjadi jauh lebih cepat. Sementara dari sisi manajemen data, administrator diberikan kemudahan untuk mengelola siklus hidup data, seperti melakukan pengarsipan atau pemeliharaan khusus pada data tahun ajaran lama tanpa mengganggu akses pengguna terhadap data tahun ajaran yang sedang aktif.

3.4 Diagram Arsitektur

Diagram arsitektur menjelaskan alur keseluruhan sistem mulai dari sumber data operasional, proses ekstraksi, transformasi, dan pemuatan (ETL), penyimpanan dalam data warehouse, hingga pemanfaatan data melalui dashboard analitik.

3.4.1 Alur Pergerakan Data

Alur data dimulai dengan proses ekstraksi dari berbagai sumber data yang heterogen, baik dari segi platform maupun format penyimpanannya. Sumber data utama adalah sistem SIAKAD berbasis SQL Server yang menyediakan data operasional inti mahasiswa dan perkuliahan. Selain itu, data juga ditarik dari Insightera (via API/Database) untuk data pendaftaran, Website Fakultas untuk informasi spesifik, serta data Manual (Excel) untuk melengkapi catatan historis yang tidak tersedia secara digital. Keberagaman format ini menuntut proses standarisasi awal agar data dapat diolah secara seragam dalam tahap selanjutnya.

Data yang telah diekstraksi kemudian dipindahkan ke dalam area transit yang disebut Staging Area. Dalam implementasi teknisnya, sebuah skema khusus bernama stg dibuat dalam basis data (seperti stg.Student, stg.Enrollment) untuk menampung data mentah ini sementara waktu. Pada tahap ini, data dimuat "apa adanya" (*as-is*) dari sumber tanpa transformasi bisnis yang kompleks. Tujuannya adalah untuk meminimalisir beban pada sistem operasional sumber dan menyediakan ruang isolasi untuk proses pembersihan data (*cleansing*) serta validasi sebelum data tersebut diizinkan masuk ke tabel produksi utama .

Tahap terakhir adalah integrasi data dari *staging* menuju tabel dimensi dan fakta di *Data Warehouse*. Proses ini dijalankan secara otomatis menggunakan T-SQL Stored Procedures yang diorkestrasi dalam satu paket ETL utama. Untuk tabel dimensi (seperti Dim_Student), digunakan logika MERGE yang mampu menangani *Slowly Changing Dimension* (SCD) sederhana: sistem akan menyisipkan data baru jika belum ada, atau memperbarui atribut data jika ditemukan perubahan pada *record* yang sudah ada. Sementara untuk tabel fakta (seperti Fact_Enrollment), proses pemuatan melibatkan pencarian kunci pengganti (*surrogate key lookup*) dengan melakukan *join* ke tabel dimensi, memastikan

bahwa setiap transaksi yang dicatat terhubung dengan referensi dimensi yang valid dan menjaga integritas referensial data .

3.4.2 Komponen Infrastruktur

Infrastruktur server operasional terdiri dari lingkungan basis data transaksional (OLTP) yang berfungsi sebagai sumber data utama. Sistem SIAKAD, yang menyimpan data inti mahasiswa dan akademik, berjalan di atas platform SQL Server yang menangani transaksi harian dengan volume tinggi. Selain server basis data utama, infrastruktur sumber juga mencakup titik akhir (*endpoints*) lain seperti API untuk sistem Insightera dan penyimpanan fail sistem untuk data manual berupa Excel/CSV yang dikelola oleh unit administrasi .

Untuk lingkungan *Data Warehouse*, infrastruktur dibangun di atas platform *cloud computing* menggunakan layanan Microsoft Azure Virtual Machine. Server ini dikonfigurasi menggunakan sistem operasi Linux (Ubuntu 24) yang menjalankan instansi SQL Server sebagai mesin basis data analitik. Guna menjamin keamanan akses ke server produksi ini, koneksi tidak dibuka secara langsung ke publik, melainkan melalui mekanisme Azure Bastion Host, yang menyediakan akses RDP/SSH yang aman dan terisolasi tanpa perlu mengekspos alamat IP publik server basis data .

Pada implementasi proses ETL menggunakan T-SQL Stored Procedures yang dijalankan langsung di dalam mesin basis data. Orkestrasi dan penjadwalan proses ETL ini dikelola oleh SQL Server Agent untuk memastikan pembaruan data berjalan otomatis sesuai jadwal. Sementara itu, untuk lapisan visualisasi dan penyajian data kepada pengguna akhir, digunakan Microsoft Power BI (.pbix) yang terhubung langsung ke *Data Warehouse* untuk menyajikan *dashboard* interaktif dan laporan manajerial .

3.4.3 Integrasi Sistem

Integrasi sistem dirancang untuk menghubungkan seluruh komponen arsitektur, mulai dari sumber data hingga tampilan visualisasi, agar dapat bekerja sebagai satu kesatuan yang utuh. Secara konseptual, hubungan antar komponen ini bersifat linier: data dari sistem operasional (seperti SIAKAD dan API Insightera) dialirkan masuk ke dalam lingkungan server *Data Warehouse* di Azure. Di dalam server tersebut, data diproses dan disimpan secara terpusat. Komponen terakhir adalah Power BI, yang terhubung langsung ke *Data Warehouse* untuk menarik data matang tersebut dan menyajikannya dalam bentuk visual kepada pengguna akhir .

Mekanisme sinkronisasi data dilakukan secara otomatis untuk menjamin kesegaran informasi tanpa memerlukan intervensi manual yang berulang. Proses ini dijalankan menggunakan skrip terjadwal (*scheduled jobs*) melalui layanan SQL Server Agent. Agen ini bertugas memicu eksekusi *Stored Procedure* ETL pada waktu-waktu yang telah ditentukan (misalnya setiap malam saat beban server rendah), yang akan memindahkan data baru dari sumber, memasukkannya ke area *staging*, dan memutakhirkan tabel dimensi serta fakta utama .

Dari sisi keamanan jaringan, pola integrasi menerapkan prinsip "akses tertutup" untuk melindungi aset data yang sensitif. Server basis data ditempatkan dalam jaringan privat yang tidak terekspos langsung ke internet publik. Akses administratif untuk pemeliharaan sistem atau pembaruan skrip ETL wajib melalui jalur aman Azure Bastion Host, yang mengenkripsi koneksi dan memverifikasi identitas pengguna sebelum memberikan akses ke server. Pola ini meminimalkan celah keamanan dari serangan luar sekaligus memastikan jalur integrasi data tetap terjaga kerahasiaannya .

BAB IV IMPLEMENTASI

4.1 Implementasi Basis Data

Tahap implementasi basis data mencakup pembuatan struktur database sesuai desain fisik yang telah dirumuskan pada bab sebelumnya. Implementasi dilakukan tanpa menyertakan query pada bab ini, tetapi seluruh skrip ditempatkan dalam lampiran. Fokus utama pada tahap ini mencakup pembuatan database, pembuatan tabel dimensi dan fakta, penerapan *primary key*, *foreign key*, indeks, serta pengaturan struktur penyimpanan data.

4.1.1 Pembuatan Skema Database

Skema data warehouse disusun untuk memastikan alur ETL berjalan terstruktur mulai dari tahap ekstraksi, pembersihan, hingga pemuatan data akhir ke dalam schema produksi. Desain skema mencakup pemisahan area penyimpanan menjadi beberapa lapisan, yaitu schema akademik (dbo) sebagai penyimpanan utama tabel dimensi dan fakta, schema staging (stg) sebagai area penampung sementara hasil ekstraksi sebelum diproses, serta direktori kerja khusus seperti *DW_AKADEMIK_DEPLOY* yang menyimpan seluruh skrip SQL deployment secara berurutan. Pembagian ini memudahkan proses validasi, troubleshooting, dan deployment ke Azure Virtual Machine melalui bastion.

Standarisasi penamaan tabel diterapkan agar konsisten dan mudah dipahami. Tabel dimensi menggunakan format Dim_* (misalnya Dim_Student, Dim_Program), tabel fakta menggunakan Fact_, *sedangkan tabel pada staging menggunakan prefix stg_* sesuai sumber datanya. Penamaan skrip deployment juga mengikuti urutan bernomor (01–08) untuk mencerminkan alur pembuatan database, tabel, indeks, hingga prosedur ETL. Dengan penamaan yang sistematis ini, seluruh proses deployment, verifikasi file, koneksi produksi, serta eksekusi ETL dapat berjalan lebih terkontrol dan minim kesalahan.

4.1.2 Pembuatan Tabel Dimensi dan Fakta

Proses pembuatan tabel dilakukan secara berurutan mulai dari pembuatan database, dilanjutkan dengan tabel dimensi, tabel fakta, indeks, serta prosedur ETL yang mengatur alur pemuatan data. Setelah struktur siap, dilakukan initial load ke tabel dimensi menggunakan prosedur MERGE, yang memastikan data awal dari schema *staging* masuk secara konsisten sambil memperbarui nilai yang sudah ada tanpa duplikasi. Untuk tabel fakta, sinkronisasi dilakukan dengan mekanisme incremental loading, yaitu memasukkan data baru dari staging hanya jika kombinasi kunci tertentu belum pernah muncul sebelumnya. Pendekatan ini menjaga fakta tetap konsisten, menghindari duplikasi transaksi, dan memastikan data warehouse selalu selaras dengan perubahan terbaru dari sistem sumber.

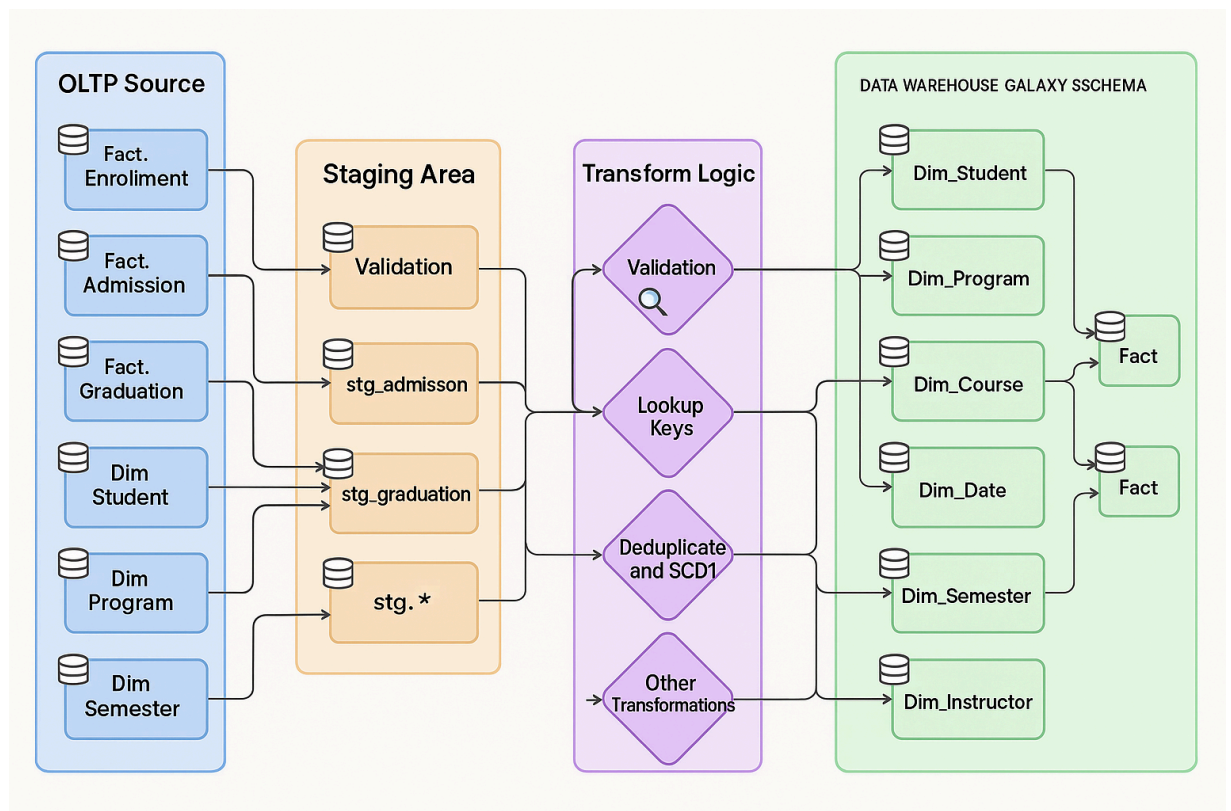
4.1.3 Penerapan Indeks dan Constraint

Penerapan indeks dilakukan terutama pada kolom kunci di tabel dimensi dan fakta, seperti *surrogate key*, *natural key*, serta kolom yang sering digunakan dalam proses join. Indeks yang digunakan meliputi clustered index pada kunci utama untuk mempercepat pengurutan fisik data, serta non-clustered index pada kolom referensi agar query analitik

berjalan lebih efisien. Pemilihan indeks ini didasarkan pada pola akses data warehouse yang dominan melakukan pembacaan besar dan join antar tabel dimensi–fakta, sehingga indeks mampu menurunkan waktu eksekusi secara signifikan. Selain itu, seluruh tabel juga menggunakan constraint seperti *primary key*, *foreign key*, *unique*, dan *check constraint* untuk menjaga integritas data, memastikan tidak ada duplikasi, hubungan antar tabel tetap valid, serta nilai yang masuk sesuai aturan bisnis yang telah ditetapkan.

4.2 Proses ETL

Tahap ETL mencakup proses ekstraksi data dari sistem operasional, transformasi untuk standarisasi dan pembersihan, serta pemuatan ke dalam data warehouse. Semua prosedur teknis seperti log eksekusi, mapping, dan script berada pada lampiran.



Gambar 4.1 proses ETL

4.2.1 Proses Ekstraksi

Data diambil dari berbagai sumber sistem akademik, termasuk data mahasiswa, program studi, dosen, mata kuliah, dan aktivitas akademik. Pengambilan dilakukan melalui akses terjadwal ke schema *staging* yang menampung hasil ekstraksi dari database operasional. Mekanisme ekstraksi berjalan otomatis menggunakan prosedur SQL/ETL terjadwal, dengan frekuensi harian agar data warehouse selalu selaras dengan sistem sumber tanpa membebani transaksi operasional.

4.2.2 Proses Transformasi

Data yang telah diekstraksi kemudian dibersihkan melalui validasi tipe data, standarisasi format kode, serta normalisasi penulisan nama dan label. Atribut tertentu

digabungkan atau dipisahkan sesuai kebutuhan model dimensi, misalnya memecah atribut tanggal atau menggabungkan kode–nama program. Nilai hilang ditangani dengan aturan imputation sederhana atau ditandai sebagai *unknown*, sedangkan data duplikat dihapus melalui proses deduplikasi. Tahap transformasi ini memastikan data sudah konsisten dan siap dimuat ke struktur *star schema*.

4.2.3 Proses Pemuatan (Loading)

Pemuatan dilakukan secara berurutan dimulai dari tabel dimensi, kemudian dilanjutkan ke tabel fakta agar seluruh foreign key dapat direferensikan dengan benar. Dua mekanisme digunakan: *initial full load* untuk pemuatan awal seluruh data, dan *incremental load* untuk menambahkan perubahan baru tanpa memproses ulang seluruh dataset. Setelah pemuatan selesai, dilakukan validasi berupa pengecekan jumlah baris, kesesuaian foreign key, serta verifikasi rekonsiliasi data dengan tabel staging untuk memastikan proses ETL berjalan akurat dan tanpa anomali.

4.3 Pengembangan Dashboard

Dashboard dikembangkan untuk mendukung analisis akademik dan membantu Unit Akademik dalam monitoring kinerja. Pengembangan dilakukan dengan menghubungkan tools visualisasi dengan data warehouse serta menyusun tampilan visual sesuai KPI yang telah ditentukan.

4.3.1 Penyusunan KPI dan Kebutuhan Laporan

KPI utama yang digunakan meliputi jumlah mahasiswa aktif, distribusi IPK, performa kelulusan mata kuliah, beban mengajar dosen, serta tren registrasi per semester. Unit Akademik memprioritaskan laporan yang mendukung monitoring kinerja studi, penentuan intervensi dini, dan evaluasi program. Jenis analisis yang dibutuhkan mencakup analisis tren, segmentasi mahasiswa, serta perbandingan antarperiode dan antarprogram.

4.3.2 Desain Tampilan Dashboard

Desain dashboard disusun secara bertahap dimulai dari struktur halaman yang membedakan ringkasan KPI, rincian analitik, dan halaman khusus untuk masing-masing entitas akademik. Visualisasi yang digunakan mencakup bar chart, line chart, pie/donut, tabel interaktif, dan KPI card. Navigasi dibuat sederhana dengan menu terstruktur dan filter interaktif agar pengguna dapat menelusuri data berdasarkan semester, program studi, atau kategori tertentu tanpa mengubah struktur laporan utama.

4.4 Implementasi Keamanan

Keamanan diterapkan untuk memastikan data akademik terlindungi dari akses tidak sah, sekaligus menjaga integritas dan konsistensi informasi. Pengaturan ini mencakup kontrol hak akses pada database dan dashboard, pengawasan aktivitas pengguna, serta perlindungan terhadap data sensitif.

4.4.1 Pengaturan Hak Akses dan Peran Pengguna

Peran pengguna dibagi berdasarkan fungsi kerja seperti admin database, staf akademik, analis data, dan pengguna dashboard biasa. Setiap peran diberikan akses terbatas pada tabel, view, atau prosedur yang relevan, sehingga data sensitif hanya dapat diakses oleh pihak berwenang. Kontrol akses pada dashboard juga diatur agar setiap user hanya dapat melihat laporan sesuai kewenangannya.

4.4.2 Audit *Trail* dan *Logging*

Setiap aktivitas penting dicatat melalui mekanisme audit trail, termasuk login pengguna, perubahan data, dan eksekusi prosedur ETL. Log ini digunakan untuk memantau anomali, menelusuri riwayat perubahan, dan mendukung proses pelaporan audit internal. Dengan sistem pencatatan yang terpusat, potensi kesalahan maupun penyalahgunaan dapat dideteksi lebih cepat dan ditindaklanjuti secara tepat.

BAB V

PENGUJIAN DAN VALIDASI

5.1 Hasil Uji Kualitas Data

Pengujian kualitas data dilakukan untuk memastikan bahwa data yang dimuat ke dalam data warehouse berada dalam kondisi lengkap, konsisten, dan valid. Seluruh data diperiksa dari aspek kelengkapan nilai, kesesuaian antar tabel, integritas referensial, serta keberadaan duplikasi dan anomali. Hasil evaluasi ditampilkan melalui tabel ringkasan dan visualisasi sederhana bila diperlukan untuk memperjelas pola atau temuan penting.

Pada tahap pemeriksaan kelengkapan data, seluruh tabel dimensi dan fakta dianalisis untuk mengetahui tingkat keberhasilan pemuatan dan mengidentifikasi nilai kosong pada atribut tertentu. Atribut-atribut yang dianggap kritis, seperti identitas mahasiswa, kode akademik, dan data waktu, diperiksa secara khusus untuk memastikan tidak ada informasi penting yang hilang. Jika ditemukan ketidakterisian, jumlah dan proporsinya dicatat pada tabel hasil pemeriksaan.

Validasi integritas dan konsistensi dilakukan dengan mengevaluasi hubungan antar tabel melalui foreign key serta memeriksa kesesuaian kode akademik dan atribut referensial lainnya. Konsistensi alur data akademik—mulai dari informasi mahasiswa hingga nilai—diamati untuk memastikan tidak ada ketidaksesuaian antar proses. Diagram hubungan atau tabel pemeriksaan keselarasan dapat digunakan untuk menggambarkan hasil evaluasi ini.

Deteksi anomali dan duplikasi dilakukan dengan mengidentifikasi baris yang muncul lebih dari satu kali serta nilai-nilai yang berada di luar kisaran wajar, baik untuk variabel numerik maupun kategorikal. Temuan yang bersifat tidak lazim dicatat untuk dianalisis lebih lanjut. Grafik distribusi dapat digunakan untuk menunjukkan pola anomali secara visual.

5.2 Pengujian Performa

Pengujian performa dilakukan untuk memastikan bahwa data warehouse mampu mendukung kebutuhan analitik dengan waktu respon yang optimal dan kinerja sistem yang stabil. Evaluasi dilakukan terhadap kecepatan eksekusi query, durasi proses ETL, serta kemampuan sistem ketika digunakan oleh banyak pengguna secara bersamaan.

Pengujian waktu eksekusi query berfokus pada query analitik yang paling sering digunakan. Setiap query dijalankan beberapa kali untuk mendapatkan estimasi waktu respon yang stabil. Query yang memiliki waktu eksekusi melebihi batas wajar dicatat dan dianalisis penyebabnya, kemudian diberikan rekomendasi optimasi. Grafik perbandingan waktu eksekusi dapat digunakan untuk menunjukkan perbedaan performa antarquery.

Pada pengujian performa ETL, dilakukan pengukuran durasi proses ekstraksi, transformasi, dan pemuatan. Tahap yang memakan waktu paling lama dicermati untuk mengidentifikasi bottleneck. Setelah perbaikan dilakukan, proses yang sama dijalankan ulang untuk melihat sejauh mana optimasi berdampak pada kecepatan pemrosesan. Ringkasan waktu proses dapat ditampilkan dalam bentuk tabel.

Pengujian beban sistem dilakukan dengan mensimulasikan penggunaan oleh beberapa pengguna secara bersamaan. Selama pengujian, penggunaan CPU, memori, dan I/O dipantau untuk menilai seberapa stabil sistem bekerja ketika terjadi peningkatan aktivitas. Grafik pemantauan resource dapat digunakan untuk memberikan gambaran perubahan beban selama pengujian berlangsung.

5.3 UAT (*User Acceptance Test*)

Pengujian UAT dilakukan bersama Unit Akademik sebagai pihak pengguna utama untuk memastikan bahwa seluruh fitur dan kemampuan analitik yang disediakan telah sesuai dengan kebutuhan operasional mereka. Pengujian ini mencakup evaluasi terhadap alur fungsi, kesesuaian laporan dan KPI, serta kemudahan penggunaan dashboard.

Skenario pengujian disusun berdasarkan proses akademik yang sebenarnya, seperti pengelolaan kelas, pencatatan aktivitas perkuliahan, dan pemantauan nilai. Setiap skenario dijalankan oleh pengguna untuk melihat apakah informasi yang ditampilkan sudah sesuai dengan ekspektasi dan apakah visualisasi mudah dipahami. Jika diperlukan, daftar skenario dapat ditampilkan dalam bentuk tabel.

Hasil UAT kemudian dievaluasi untuk mengidentifikasi temuan yang muncul selama sesi pengujian. Pengguna memberikan umpan balik terkait tampilan, kelengkapan informasi, kecepatan akses, dan alur penggunaan dashboard. Rangkuman tingkat penerimaan pengguna dicatat sebagai indikator apakah sistem dapat digunakan tanpa revisi mayor atau masih memerlukan penyempurnaan.

Tahap akhir UAT adalah proses persetujuan pengguna, yaitu konfirmasi formal bahwa sistem telah memenuhi kebutuhan operasional mereka. Persetujuan tersebut menjadi dasar bahwa sistem siap digunakan dalam lingkungan kerja sehari-hari. Ringkasan hasil UAT serta pernyataan penerimaan pengguna dituangkan dalam bentuk dokumentasi akhir.

5.4 Perbaikan Bug

Perbaikan bug dilakukan berdasarkan seluruh temuan dari pengujian kualitas data, pengujian performa, dan UAT. Setiap masalah yang ditemukan dianalisis untuk mengetahui penyebabnya, kemudian diperbaiki melalui penyesuaian skrip ETL, perbaikan struktur data, atau penyempurnaan logika visualisasi.

Identifikasi bug dilakukan dengan mengelompokkan setiap temuan sesuai tingkat keparahannya. Bug yang berdampak langsung pada kebenaran data atau menghalangi proses analitik diprioritaskan untuk ditangani terlebih dahulu. Catatan mengenai jenis bug, lokasi, dan dampaknya dapat disusun dalam bentuk tabel agar lebih mudah dilacak.

Proses perbaikan dilakukan melalui revisi skrip ETL yang bermasalah, penyesuaian struktur tabel atau view, serta perbaikan logika perhitungan pada dashboard. Setelah perbaikan dilakukan, pengujian ulang diterapkan untuk memastikan bahwa masalah tidak muncul kembali dan hasil data telah sesuai dengan standar. Bila diperlukan, diagram alur

proses sebelum dan sesudah perbaikan dapat digunakan untuk menjelaskan perubahan secara visual.

BAB VI

DEPLOYMENT DAN OPERASIONAL

6.1 Deployment Produksi

Deployment produksi merupakan proses menempatkan sistem Data Warehouse ke lingkungan yang digunakan oleh pengguna akhir. Pada tahap ini dilakukan konfigurasi server, pengaturan koneksi ke database produksi, aktivasi pipeline ETL, serta penyesuaian parameter keamanan dan akses. Seluruh komponen sistem dipastikan berjalan stabil dan sesuai standar operasional agar layanan dapat digunakan tanpa gangguan.

6.2 Strategi Backup

Strategi backup digunakan untuk memastikan seluruh data dan konfigurasi sistem tetap aman apabila terjadi kegagalan, kerusakan, atau kehilangan data. Proses ini mencakup penjadwalan backup rutin, penyimpanan salinan di lokasi terpisah, serta pengujian pemulihan (restore) secara berkala. Dengan strategi yang tepat, sistem dapat dipulihkan dengan cepat ketika terjadi insiden.

6.3 Monitoring Sistem

Monitoring sistem dilakukan untuk memantau kesehatan, performa, dan ketersediaan Data Warehouse secara berkelanjutan. Proses ini meliputi pengawasan penggunaan sumber daya server, aktivitas ETL, performa query, dan notifikasi otomatis jika terjadi anomali. Monitoring yang baik membantu mendeteksi masalah sejak dini sehingga dapat ditangani sebelum mengganggu pengguna.

6.4 Prosedur Pemeliharaan

Prosedur pemeliharaan bertujuan menjaga agar sistem tetap optimal dalam jangka panjang. Kegiatan ini mencakup pembaruan perangkat lunak, pembersihan data sementara, evaluasi struktur penyimpanan, serta pengujian berkala terhadap keamanan sistem. Pemeliharaan dilakukan secara terjadwal untuk memastikan operasi Data Warehouse tetap efisien dan aman.

BAB VII

HASIL DAN PEMBAHASAN

7.1 Temuan Utama

Analisis terhadap data yang telah dimuat ke dalam Data Warehouse menghasilkan sejumlah temuan penting terkait pola, tren, dan hubungan antarvariabel. Hasil implementasi menunjukkan bagaimana integrasi berbagai sumber data mampu memberikan gambaran yang lebih lengkap dibandingkan data operasional awal. Temuan-temuan yang muncul mencakup identifikasi kecenderungan tertentu, anomali yang berhasil disaring melalui proses kualitas data, serta verifikasi bahwa struktur dimensional yang digunakan mampu menangkap kebutuhan analitik yang dirancang pada tahap perancangan.

7.2 *Insight* Bisnis

Hasil pengolahan data memberikan sejumlah *insight* yang relevan secara bisnis, khususnya untuk mendukung proses pengambilan keputusan. Informasi yang dihasilkan dashboard memperlihatkan area yang perlu perhatian, tren yang dapat dimanfaatkan untuk perencanaan, serta indikator kinerja yang menggambarkan kondisi nyata dalam periode tertentu. *Insight* ini memberi dasar yang lebih objektif dalam merumuskan strategi, meningkatkan efektivitas proses, atau mengoptimalkan penggunaan sumber daya.

7.3 Metrik Performa

Evaluasi performa dilakukan menggunakan beberapa metrik, seperti waktu eksekusi query, efisiensi proses pemuatan data, tingkat konsistensi antar-tabel, serta keberhasilan pembersihan data pada tahap ETL. Pengukuran performa menunjukkan seberapa baik sistem merespons permintaan analitik dan bagaimana optimasi desain berkontribusi pada kecepatan akses. Perbandingan metrik sebelum dan sesudah proses tuning memberikan gambaran mengenai stabilitas dan efektivitas sistem dalam kondisi beban tertentu.

7.4 Tantangan dan Solusi

Selama pengembangan dan implementasi, terdapat sejumlah tantangan yang muncul, baik dari sisi kualitas data, ketidaksesuaian struktur sumber, konfigurasi ETL, maupun keterbatasan lingkungan server. Setiap masalah dianalisis, dan pendekatan penyelesaian diterapkan untuk memastikan sistem tetap dapat berjalan sesuai kebutuhan. Proses debugging, penyesuaian transformasi, perbaikan model, serta optimalisasi query menjadi bagian dari rangkaian solusi yang diterapkan hingga sistem mencapai performa yang stabil.

BAB VIII

KESIMPULAN DAN SARAN

8.1 Ringkasan Pencapaian

Pengembangan *Data Warehouse* telah melalui seluruh tahapan mulai dari perancangan konseptual, pembangunan struktur dimensional, implementasi ETL, hingga pengujian dan penyajian dashboard analitik. Seluruh komponen tersebut berhasil diintegrasikan sehingga menghasilkan sistem yang dapat menampung, memproses, dan menyajikan informasi secara konsisten. Hasil akhir menunjukkan bahwa tujuan utama proyek, yaitu menyediakan platform analitik yang handal dan mudah diakses, telah tercapai.

8.2 Implikasi dan Manfaat

Implementasi *Data Warehouse* memberikan manfaat langsung berupa peningkatan kualitas informasi dan efisiensi dalam analisis data. Konsolidasi data dari berbagai sumber memungkinkan pemangku kepentingan untuk memahami kondisi operasional dan akademik secara lebih menyeluruh. Sistem juga mendukung proses evaluasi dan perencanaan jangka panjang dengan menyediakan indikator kinerja yang lebih terstruktur, akurat, dan mudah dievaluasi.

8.3 Keterbatasan Proyek

Meskipun berhasil dibangun dengan fungsionalitas utama yang lengkap, proyek ini masih memiliki beberapa keterbatasan, seperti cakupan data yang terbatas pada sumber tertentu, ketergantungan pada struktur data operasional, serta belum diterapkannya beberapa metode analitik lanjutan. Batasan teknis seperti spesifikasi server dan kapasitas penyimpanan juga menjadi faktor yang perlu diperhatikan untuk pengembangan berikutnya.

8.4 Saran Pengembangan Lanjutan

Pengembangan lanjutan dapat diarahkan pada perluasan sumber data, penerapan data enrichment, serta peningkatan mekanisme keamanan dan audit. Selain itu, integrasi dengan sistem bisnis lainnya dan penggunaan teknik analitik prediktif dapat menambah nilai strategis *Data Warehouse*. Penyempurnaan desain dashboard dan otomatisasi proses ETL juga dapat meningkatkan kenyamanan pengguna dan efisiensi pengelolaan data.

DAFTAR PUSTAKA

- [1] W. H. Inmon, *Building the Data Warehouse*, 4th ed. Hoboken, NJ, USA: Wiley, 2005.
- [2] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.
- [3] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA, USA: Morgan Kaufmann, 2012.

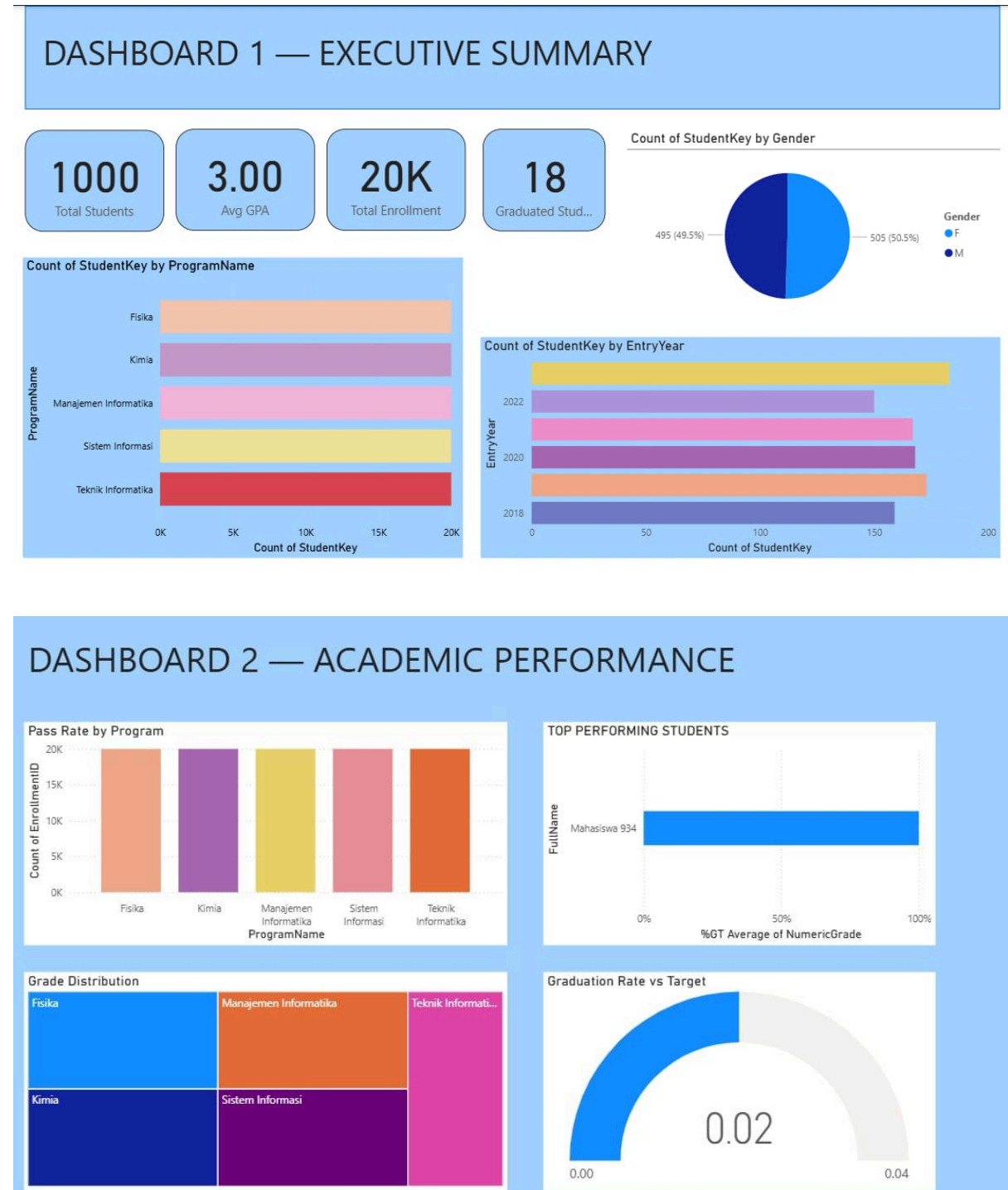
LAMPIRAN

Skrip SQL

Skrip SQL dapat diakses di :

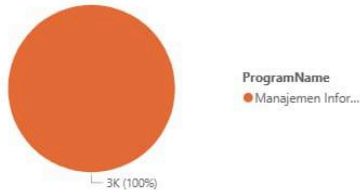
https://github.com/sains-data/Kelompok02_Akademik.git

Screenshots Dashboard

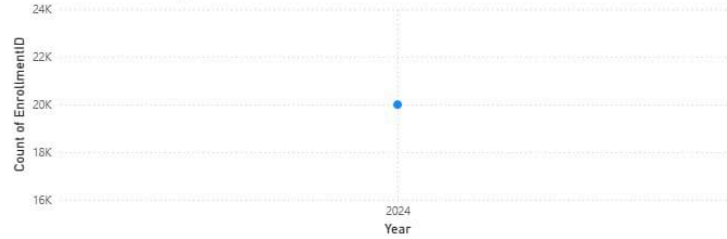


DASHBOARD 3 — FINANCIAL ANALYSIS

Count of GraduationID by ProgramName



Count of EnrollmentID by Year



Count of AdmissionID by AdmissionStatus



Hasil Uji

SQLQuery1....ongo (110)*

```

1  SET STATISTICS TIME ON;
2  SET STATISTICS IO ON;
3  GO
4
5  -- Query 1: Total mahasiswa & rata-rata nilai per program per tahun
6  SELECT
7    p.ProgramName,
8    d.[Year],
9    COUNT(DISTINCT f.StudentKey) AS TotalStudents,
10   AVG(f.NumericGrade) AS AvgGrade
11  FROM dbo.Fact_Enrollment f
12  JOIN dbo.Dim_Program p ON f.CourseKey = f.CourseKey
13  JOIN dbo.Dim_Date d ON f.DateKey = d.DateKey
14  GROUP BY p.ProgramName, d.[Year]
15  ORDER BY d.[Year], TotalStudents DESC;
16  GO
  
```

75 % 14 0 75 % Ln: 12, Ch: 55 SPC CRLF Windows 1252

ProgramName	Year	TotalStudents	AvgGrade
1 Teknik Sistem Energi - Energi Terbarukan	2024	31	2.957450
2 Teknik Industri - Logistik	2024	31	2.957450
3 Teknik Sipil	2024	31	2.957450
4 Teknik Sistem Energi	2024	31	2.957450
5 Sains Aktuaria	2024	31	2.957450
6 Farmasi	2024	31	2.957450
7 Meteorologi Terapan	2024	31	2.957450
8 Teknologi Industri Pertanian	2024	31	2.957450
9 Teknik Fisika	2024	31	2.957450
10 Teknik Informatika	2024	31	2.957450
11 Teknik Sipil - Konstruksi	2024	31	2.957450
12 PWK - Transportasi	2024	31	2.957450
13 Sains Data & Kecerdasan Buatan	2024	31	2.957450
14 Teknik Biosistem	2024	31	2.957450
15 Teknik Geomatika	2024	31	2.957450
16 Biologi	2024	31	2.957450
17 Teknologi Pangan	2024	31	2.957450
18 Teknik Lingkungan - Sanitasi Daerah	2024	31	2.957450
19 Kimia Analitik	2024	31	2.957450
20 Rekayasa Tata Kelola Air Terpadu	2024	31	2.957450
21 Teknik Kelautan	2024	31	2.957450
22 Sains Atmosfer dan Keplanetan	2024	31	2.957450
23 DKV - Animasi & Media Digital	2024	31	2.957450
24 PWK - Lingkungan & Permukiman	2024	31	2.957450
25 Teknik Kimia	2024	31	2.957450
26 Perencanaan Wilayah dan Kota	2024	31	2.957450

Query executed successfully. BITLANCKA\MSSQLSERVER01 (16... BITLANCKA\Pongo (110) akademik 00:00:00 Row: 1, Col: 1 53 rows

File Edit View Query Git Project Tools Extensions Window Help Search

DM_AKADEMIK Execute

Object Explorer

Connect BITLANCKA\MSSQLSERVER01 (SQL Server 16.0.1000.6)

Databases

- System Databases
- Database Snapshots
- akademik
- DM_AKADEMIK
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Dim_Course
 - dbo.Dim_Date
 - dbo.Dim_Instructor
 - dbo.Dim_Program
 - dbo.Dim_Semester
 - dbo.Dim_Student
 - dbo.Fact_Admission
 - dbo.Fact_Enrollment
 - dbo.Fact_Graduation
 - Dropped Ledger Tables
 - Views
 - External Resources

SQLQuery1....Pongo (67)*

```

1  -- Query 2: Trend enrollment bulanan
2  SELECT
3      d.[Year],
4      d.[Month],
5      COUNT(*) AS TotalEnrollments
6  FROM dbo.Fact_Enrollment f
7  JOIN dbo.Dim_Date d ON f.DateKey = d.DateKey
8  GROUP BY d.[Year], d.[Month]
9  ORDER BY d.[Year], d.[Month];
10 GO

```

91% 10 0

Results Messages

	Year	Month	TotalEnrollments
1	2024	6	20000

File Edit View Query Git Project Tools Extensions Window Help Search

DM_AKADEMIK Execute

Object Explorer

Connect BITLANCKA\MSSQLSERVER01 (SQL Server 16.0.1000.6)

Databases

- System Databases
- Database Snapshots
- akademik
- DM_AKADEMIK
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Dim_Course
 - dbo.Dim_Date
 - dbo.Dim_Instructor
 - dbo.Dim_Program
 - dbo.Dim_Semester
 - dbo.Dim_Student
 - dbo.Fact_Admission
 - dbo.Fact_Enrollment
 - dbo.Fact_Graduation
 - Dropped Ledger Tables
 - Views
 - External Resources

SQLQuery1....Pongo (67)*

```

1  -- Query 3: Funnel Admission + Enrollment + Graduation
2
3
4  ;WITH Adm AS (
5      SELECT
6          a.ProgramKey,
7          d.[Year] AS AdmissionYear,
8          COUNT(DISTINCT a.StudentKey) AS TotalApplicants
9      FROM dbo.Fact_Admission a
10     JOIN dbo.Dim_Date d ON a.AdmissionDateKey = d.DateKey
11     GROUP BY a.ProgramKey, d.[Year]
12  ),
13  Enr AS (
14      SELECT
15          c.ProgramNaturalID,
16          d.[Year] AS EnrollmentYear,
17          COUNT(DISTINCT f.StudentKey) AS TotalEnrolled
18      FROM dbo.Fact_Enrollment f
19      JOIN dbo.Dim_Course c ON f.CourseKey = c.CourseKey
20      JOIN dbo.Dim_Date d ON f.DateKey = d.DateKey
21      GROUP BY c.ProgramNaturalID, d.[Year]

```

91% 39 0

Results Messages

	ProgramName	AdmissionYear	TotalApplicants	TotalEnrolled	TotalGraduated	Conv_Adm_to_Enroll	Conv_Adm_to_Grad
1	Sistem Informasi	2021	1	0	0	0.00000000000000	0.00000000000000