



**PERANCANGAN DAN PEMBANGUNAN *DASHBOARD* NILAI SISWA  
BERBASIS WEB MENGGUNAKAN STREAMLIT**

**KERJA PRAKTIK**



UNIVERSITAS  
**Dinamika**

**Oleh:**

**MUHAMMAD HARIS M.K.A**

**21410100029**

---

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2025**

**PERANCANGAN DAN PEMBANGUNAN *DASHBOARD* NILAI SISWA  
BERBASIS WEB MENGGUNAKAN STREAMLIT**

Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana



**Disusun Oleh:**

**Nama : Muhammad Haris M.K.A**

**NIM : 21410100029**

**Program : S1 (Strata Satu)**

**Jurusan : Sistem Informasi**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

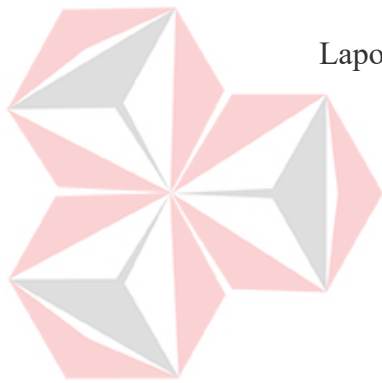
**2025**



“Yang berlalu biarlah berlalu..”

-Haris-

UNIVERSITAS  
**Dinamika**



Laporan Kerja Praktik ini saya persembahkan sepenuhnya dan

saya dedikasikan untuk keluarga, dosen  
pembimbing saya, dan seluruh teman teman saya

**LEMBAR PENGESAHAN**

**PERANCANGAN DAN PEMBANGUNAN *DASHBOARD* NILAI SISWA  
BERBASIS WEB MENGGUNAKAN STREAMLIT**

**Laporan Kerja Praktik**

**oleh:**

**Muhammad Haris M.K.A**

NIM. 21410100029

Telah diperiksa, diuji, dan disetujui

Surabaya, 16 Juli 2025

Disetujui

Dosen Pembimbing

Penyelia,



2025.07.30  
11:23:18  
+07'00'

**Erwin Sutomo, S.Kom., M.Eng.**

NIDN. 0722057501



**Drs. Mujiono, M.M**

NIP. 19671006 19912 1 005

Mengetahui,

Ketua Program Studi S1 Sistem Informasi



Digitally signed by  
Endra Rahmawati  
Date: 2025.08.01  
08:43:02 +07'00'

**Endra Rahmawati, M.Kom.**

NIDN. 0712108701

**PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, Saya :

Nama : **Muhammad Haris Maulana Koliq Abiansyah**  
NIM : **21410100029**  
Program Studi : **SI Sistem Informasi**  
Fakultas : **Fakultas Teknologi dan Informatika**  
Jenis Karya : **Laporan Kerja Praktik**  
Judul Karya : **PERANCANGAN DAN PEMBANGUNAN DASHBOARD NILAI SISWA BERBASIS WEB MENGGUNAKAN STREAMLIT**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 29 Juli 2025



M. Haris M.K.A  
NIM : 21410100029

## ABSTRAK

Pengelolaan nilai siswa di UPT SMP Negeri 16 Gresik yang masih bersifat manual menggunakan spreadsheet menyebabkan proses analisis menjadi lambat, rentan terhadap kesalahan, dan menyulitkan pemantauan perkembangan siswa secara efektif . Untuk mengatasi masalah tersebut, dilakukan perancangan dan pembangunan *dashboard* nilai siswa berbasis web menggunakan bahasa pemrograman Python dengan *framework* Streamlit, yang bertujuan untuk mempermudah visualisasi data nilai agar lebih interaktif dan mudah dipahami oleh guru dan kepala sekolah. Aplikasi yang dihasilkan menyediakan fitur utama seperti *dashboard* interaktif, filter data berdasarkan kelas dan tahun ajaran, serta berbagai jenis grafik (batang, garis, lingkaran, dan *heatmap*) untuk analisis mendalam. Pengujian sistem dilakukan dengan metode *Blackbox*, dan hasilnya menunjukkan bahwa seluruh fungsionalitas yang dirancang telah berjalan dengan baik sesuai harapan. Dengan adanya *dashboard* ini, pihak sekolah dapat memantau prestasi siswa, mengevaluasi kinerja akademik, dan mendukung pengambilan keputusan berdasarkan data yang akurat secara lebih cepat dan efisien.

**Kata kunci:** *Dashboard*, Nilai Siswa , Streamlit, Visualisasi Data

## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, atas limpahan rahmat, hidayah, dan karunia-Nya, penulis dapat menyelesaikan laporan Kerja Praktik ini dengan baik, lancar, dan tepat waktu. Laporan ini disusun sebagai salah satu persyaratan untuk menyelesaikan program studi di Universitas Dinamika, sekaligus sebagai bentuk tanggung jawab akademik dalam menyampaikan hasil pengalaman dan pembelajaran selama menjalani Kerja Praktik.

Dalam proses penyelesaian laporan ini, penulis menghadapi berbagai tantangan yang pada akhirnya dapat dilalui berkat dukungan, motivasi, dan bantuan dari banyak pihak. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang mendalam kepada:

1. Bapak Yoppy Mirza Maulana, S.Kom., M.MT. selaku dosen wali yang telah mengizinkan penulis untuk mengikuti Kerja Praktik
2. Keluarga tercinta, yang selalu memberikan cinta, doa, serta dukungan moral. menjadi pilar utama yang memberikan semangat kepada penulis dalam menyelesaikan setiap tahapan Kerja Praktik dan laporan ini.
3. Teman-teman komunitas BaldFanbase yang selalu memberikan semangat, motivasi, serta inspirasi yang membantu penulis untuk tetap optimis dan berkomitmen selama menjalani proses ini.
4. Bapak Erwin Sutomo, S.Kom., M.Eng. yang telah mengarahkan dan membimbing saat mengerjakan laporan ini.
5. Pihak Mitra yang telah memberikan saya kesempatan untuk kerja praktik di tempat mereka



Penulis menyadari bahwa laporan ini masih memiliki kekurangan dan jauh dari kata sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari semua pihak sebagai masukan untuk perbaikan di masa mendatang. Akhir kata, penulis berharap bahwa laporan ini dapat memberikan manfaat yang sebesar-besarnya bagi pembaca, khususnya bagi mahasiswa yang akan menjalani Kerja Praktik serupa di masa depan. Semoga laporan ini juga dapat memberikan kontribusi positif bagi pengembangan ilmu pengetahuan.

Surabaya, 01 Juni 2025

Penulis



UNIVERSITAS  
**Dinamika**

## DAFTAR ISI

	Halaman
ABSTRAK .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI .....	ix
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN .....	xv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Manfaat .....	3
BAB II GAMBARAN UMUM PERUSAHAAN .....	4
2.1 Profil Perusahaan .....	4
2.2 Visi Perusahaan .....	5
2.3 Misi Perusahaan .....	5
2.4 Struktur Organisasi .....	6
BAB III LANDASAN TEORI .....	7
3.1 Software Development Life Cycle .....	7
3.1.1 Perencanaan .....	8
3.1.2 Analisis .....	8
3.1.3 Desain .....	8

3.1.4	Implementasi.....	8
3.1.5	Testing.....	8
3.1.6	Maintenance.....	9
3.2	Waterfall.....	9
3.3	Python .....	10
3.4	Streamlit.....	10
3.5	Blackbox Testing .....	10
BAB IV DESKRIPSI PEKERJAAN .....		11
4.1	Analisis.....	11
4.1.1	Identifikasi Masalah.....	11
4.1.2	Analisis Kebutuhan .....	12
4.2	Desain.....	16
4.2.1	Alur system .....	16
4.2.2	Context diagram.....	24
4.2.3	DFD level 0 .....	25
4.2.4	Pemodelan data .....	26
4.3	Implementation .....	28
4.3.1	Menampilkan Data Akademik .....	28
4.3.2	Tampilan Sidebar .....	29
4.3.3	View Ringkasan & Perbandingan.....	32
4.3.4	View Analisis Tambahan .....	35
4.3.5	View Tabel Siswa .....	37
4.4	Testing.....	39
4.5	Maintenance .....	49

BAB V PENUTUP.....	50
5.1    Kesimpulan .....	50
5.2    Saran.....	51
DAFTAR PUSTAKA .....	53
LAMPIRAN.....	54



UNIVERSITAS  
Dinamika

## DAFTAR TABEL

	<b>Halaman</b>
Tabel 4.1 Tabel Kebutuhan Fungsional .....	13
Tabel 4.2 Tabel Kebutuhan Non-Fungsional .....	14
Tabel 4.3 Kebutuhan Perangkat .....	16
Tabel 4.4 Tabel Skenario Testing .....	41



UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Logo UPT SMP NEGERI 16 Gresik .....	5
Gambar 2.2 Struktur Organisasi.....	6
Gambar 3.1 Tahapan SDLC (Dewa & Azizah, 2024) .....	7
Gambar 3.2 Arsitektur Waterfall.....	9
Gambar 4.1 Alur Dashboard Nilai .....	17
Gambar 4.2 Alur Filter Nilai.....	18
Gambar 4.3 Alur Metriks Nilai .....	19
Gambar 4.4 Alur Diagram Bar Nilai per-Kelas .....	20
Gambar 4.5 Alur Line Chart Tren Nilai Tahunan.....	21
Gambar 4.6 Pie Chart Status Kelulusan.....	22
Gambar 4.7 <i>Heatmap</i> Sebaran Nilai .....	23
Gambar 4. 8 Tabel nilai siswa.....	24
Gambar 4.9 Context Diagram .....	25
Gambar 4.10 DFD Level 0.....	26
Gambar 4.11 Pemodelan data .....	27
Gambar 4.12 Menampilkan Data Akademik .....	29
Gambar 4.13 Sidebar.....	30
Gambar 4.14 Filter Data.....	31
Gambar 4.15 Metrik-Metrik Kunci Nilai Siswa .....	32
Gambar 4.16 Tabs Ringkasan dan Perbandingan .....	32
Gambar 4. 17 Grafik Distribusi Status Kelulusan.....	33
Gambar 4.18 Grafik Rata-Rata Nilai per Kelas .....	34

Gambar 4.19 View Analisis Tambahan .....	35
Gambar 4. 20 Grafik Tren Nilai Per Tahun Ajaran .....	36
Gambar 4. 21 <i>Heatmap</i> Nilai Mapel per Kelas.....	37
Gambar 4.22 View Tabel Siswa.....	38
Gambar 4. 23 Tabel nilai siswa.....	39



UNIVERSITAS  
Dinamika

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Balasan dari Mitra .....	54
Lampiran 2 Form KP 3 .....	55
Lampiran 3 Form KP 3A .....	56
Lampiran 4 FORM KP 4.....	57
Lampiran 5. Form KP-5 .....	58
Lampiran 6. Form KP-6 .....	60
Lampiran 7. Form KP-7 .....	61
Lampiran 8. Form Bimbingan.....	62
Lampiran 9 Surat Adopsi .....	63
Lampiran 10 Biodata.....	64



UNIVERSITAS  
Dinamika



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

UPT SMP Negeri 16 Gresik merupakan salah satu sekolah menengah pertama negeri di bawah naungan Dinas Pendidikan Kabupaten Gresik yang berkomitmen dalam meningkatkan mutu pendidikan. Sebagai lembaga pendidikan formal, sekolah ini memiliki tanggung jawab tidak hanya dalam proses pembelajaran di kelas, tetapi juga dalam tata kelola data akademik siswa yang rapi dan akurat. Salah satu data penting yang harus dikelola dengan baik adalah nilai siswa, yang menjadi dasar dalam mengevaluasi capaian pembelajaran dan pengambilan keputusan oleh pihak sekolah.

Di era digital saat ini, pemanfaatan teknologi informasi dalam dunia pendidikan menjadi hal yang tak terelakkan. Teknologi telah membantu institusi pendidikan dalam mendigitalisasi berbagai aspek manajemen akademik, termasuk sistem nilai (Fahrezy et al., 2020). Sayangnya, sistem visualisasi nilai di UPT SMP Negeri 16 Gresik hingga saat ini masih menggunakan cara manual atau aplikasi spreadsheet sederhana. Hal ini menyebabkan sejumlah kendala seperti keterlambatan rekapitulasi, kesalahan input data, serta kesulitan dalam visualisasi nilai siswa secara menyeluruh.

UPT SMP Negeri 16 Gresik merupakan salah satu sekolah menengah pertama negeri di bawah naungan Dinas Pendidikan Kabupaten Gresik yang berkomitmen dalam meningkatkan mutu pendidikan. Sebagai lembaga pendidikan formal, sekolah ini memiliki tanggung jawab tidak hanya dalam proses

pembelajaran di kelas, tetapi juga dalam tata kelola data akademik siswa yang rapi dan akurat. Salah satu data penting yang harus dikelola dengan baik adalah nilai siswa, yang menjadi dasar dalam mengevaluasi capaian pembelajaran dan pengambilan keputusan oleh pihak sekolah.

Di era digital saat ini, pemanfaatan teknologi informasi dalam dunia pendidikan menjadi hal yang tak terelakkan. Teknologi telah membantu institusi pendidikan dalam mendigitalisasi berbagai aspek manajemen akademik, termasuk sistem nilai (Fahrezy et al., 2020). Sayangnya, sistem visualisasi nilai di UPT SMP NEGERI 16 Gresik hingga saat ini masih menggunakan cara manual atau aplikasi spreadsheet sederhana. Hal ini menyebabkan sejumlah kendala seperti keterlambatan rekapitulasi, kesalahan input data, serta kesulitan dalam visualisasi nilai siswa secara menyeluruh.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang terurai di atas, maka di dapat rumusan masalah yang diangkat yaitu: Bagaimana merancang dan membangun sebuah *dashboard* nilai berbasis web yang efektif untuk memvisualisasikan data akademik, sehingga dapat mempermudah guru dalam mengelola nilai serta membantu siswa, orang tua, dan kepala sekolah dalam memantau kinerja akademik secara efisien dan transparan?

## 1.3 Batasan Masalah

Berdasarkan Latar belakang, batasan-batasan masalah terkait kerja praktik ini sebagai berikut:

1. *Dashboard* dirancang untuk diakses melalui web browser pada komputer

pribadi.

2. Tidak merancang sistem database.

#### 1.4 Tujuan

Menghasilkan sebuah sistem *dashboard* nilai berbasis web yang fungsional dan informatif untuk memvisualisasikan data akademik, guna meningkatkan efisiensi dan transparansi dalam proses pemantauan kinerja siswa di UPT SMP NEGERI 16 GRESIK

#### 1.5 Manfaat

Aplikasi *dashboard* memiliki manfaat sebagai berikut:

1. Meminimalisir kesalahan input data.
2. Mempermudah visualisasi dan pemantauan perkembangan akademik.
3. Mendukung pengambilan keputusan.
4. Menjadi sarana evaluasi dan monitoring siswa



UNIVERSITAS  
Dinamika

## **BAB II**

### **GAMBARAN UMUM PERUSAHAAN**

#### **2.1 Profil Perusahaan**

UPT Sekolah Menengah Pertama Negeri 16 Kedamean berdiri sejak 7 November tahun 1983. Dari tahun ke tahun jumlah siswanya terus meningkat. Tahun pertama hanya 3 rombongan belajar, sekarang menjadi 27 rombongan belajar.

Tahun Pelajaran 2021/2022 UPT SMPN 16 Gresik mempunyai 860 siswa, 47 tenaga guru dan 18 tenaga tata usaha. Memiliki sarana dan prasarana cukup memadai yaitu : 27 Ruang Kelas, Ruang Laboratorium IPA, Ruang Laboratorium Komputer, Ruang Laboratorium Bahasa, Ruang Multimedia, Ruang Musik, Green House, Ruang Kesenian, Kantin Sehat dan Ruang Lingkungan Hidup

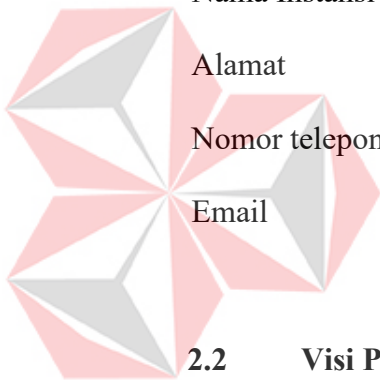
Seluruh tenaga guru berijazah Sarjana (S1) sebanyak 40 guru dan S2 sebanyak 7 guru, serta tenaga kependidikan dan semuanya mempunyai komitmen yang kuat terhadap tugasnya. Hal ini berimbas pada prestasi akademik yang cenderung meningkat dan prestasi non akademik yang membanggakan baik di tingkat Kabupaten, Propinsi maupun Nasional, pada Gambar 2.1 merupakan Logo UPT SMP Negeri 16 Gresik.



Gambar 2.1 Logo UPT SMP NEGERI 16 Gresik

Identitas perusahaan yang diinformasikan meliputi nama perusahaan, alamat, nomor telepon, serta email. Adapun rincian identitasnya adalah sebagai berikut.

Nama Instansi : UPT SMP NEGERI 16 Gresik  
 Alamat : Jl. Raya Kedamean no. 19B Kedamean Gresik  
 Nomor telepon : 031-7911190  
 Email : smpnegeri16gresik@yahoo.com



## 2.2 Visi Perusahaan

Mewujudkan generasi Indonesia yang unggul dan berkarakter profil pelajar Pancasila, yang menggambarkan tekad sekolah dalam mencetak peserta didik yang tidak hanya cerdas dalam hal akademik maupun non-akademik.

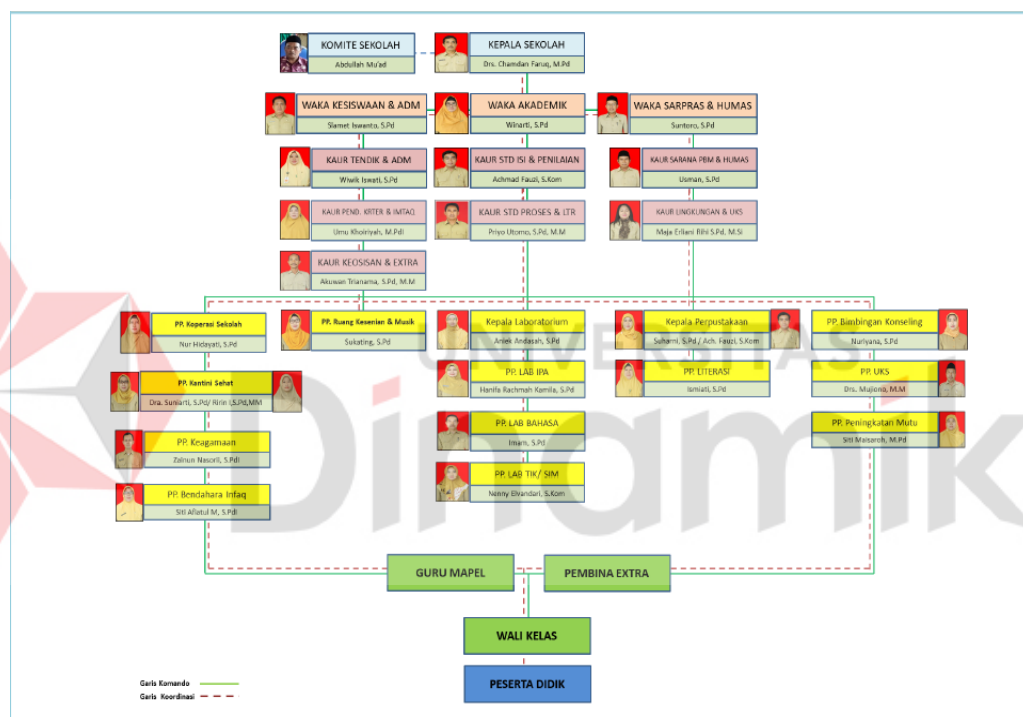
## 2.3 Misi Perusahaan

Misi dari UPT SMP Negeri 16 Gresik adalah :

1. Mewujudkan peserta didik yang mengedepankan karakter profil pelajar Pancasila.
2. Mewujudkan peserta didik yang unggul dibidang literasi dan numerasi.
3. Mewujudkan lingkungan Pendidikan yang kodusif.

## 2.4 Struktur Organisasi

Gambar 2.2 dibawah menggambarkan struktur organisasi UPT SMPN NEGERI Gresik 16 yang terdiri dari Komite Sekolah, diikuti dengan Kepala Sekolah, lalu dibawahnya terdapat Wakil Kepala sekolah Kesiswaan dan Administrasi, Wakil Kepala Akademik, dan Wakil Kepala Sarana prasarana dan Hubungan Masyarakat.



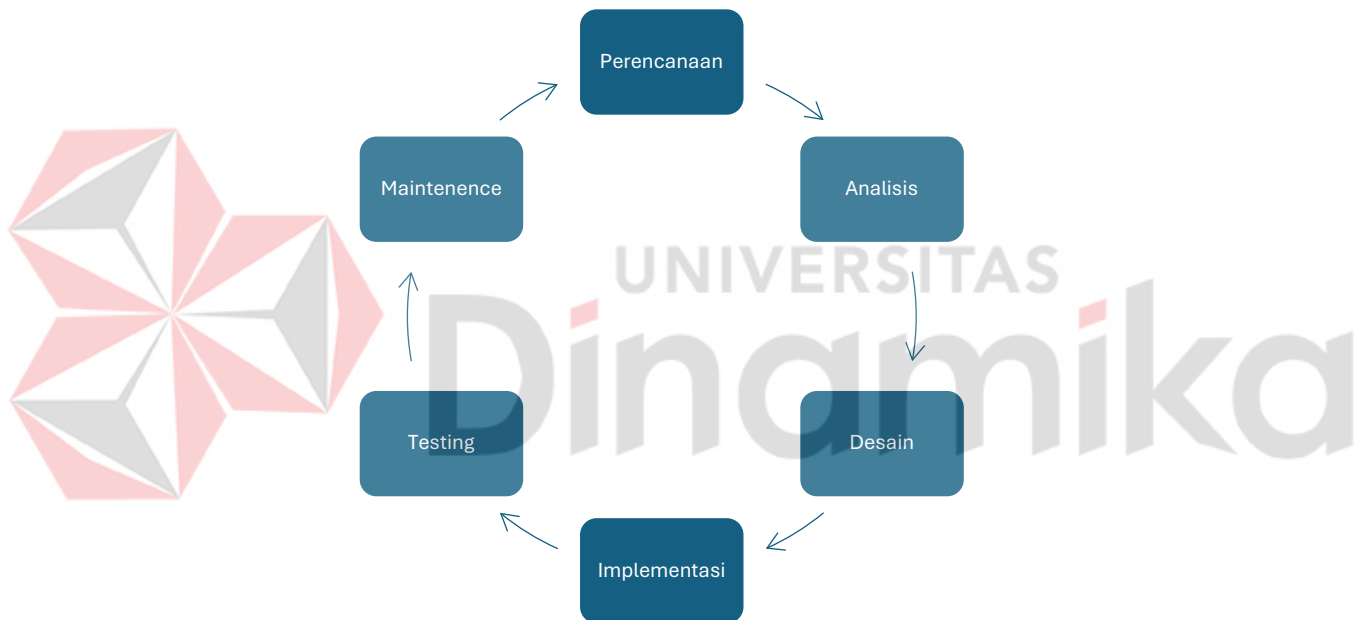
Gambar 2.2 Struktur Organisasi

## BAB III

### LANDASAN TEORI

#### 3.1 *Software Development Life Cycle*

SDLC adalah sebuah kerangka kerja terstruktur yang berisi tahapan-tahapan dalam proses pengembangan perangkat lunak, mulai dari perencanaan hingga pemeliharaan. Penggunaan SDLC membantu memastikan proses pengembangan berjalan sistematis dan terorganisir (A. A. Wahid, 2020).



Gambar 3.1 Tahapan SDLC (Dewa & Azizah, 2024)

### **3.1.1 Perencanaan**

Pada bagian ini dimulainya perencanaan proyek secara menyeluruh. Berawal dengan merencanakan dan mengkalkulasi sumber daya, jadwal, dan anggaran.

### **3.1.2 Analisis**

Tahap kedua ini dilakukan tahap analisis kebutuhan pengguna, apa saja yang diperlukan pengguna di aplikasi tersebut.

### **3.1.3 Desain**

Tahap ketiga adalah tahap desain, di mana kita mulai mendesain aplikasi secara menyeluruh. Pada tahap ini, kita merancang bagaimana alur aplikasi akan berjalan, menyusun struktur database, membuat rancangan tampilan dan fungsi aplikasi, serta menentukan bagaimana data akan mengalir di dalam aplikasi tersebut.

### **3.1.4 Implementasi**

Tahap keempat adalah tahap implementasi, pada tahap ini tahap dilakukannya penulisan kode sesuai dengan desain yang kita buat dan menggunakan bahasa pemrograman dan tools yang dipilih.

### **3.1.5 Testing**

Tahap kelima adalah tahap testing, pada tahap ini dilakukan proses pengujian aplikasi. Proses ini dilakukan pada setiap fungsi dari aplikasi, mulai dari fungsi-fungsi utama hingga detail-detail kecil, untuk mendeteksi adanya bug, kesalahan logika, atau ketidaksesuaian dengan kebutuhan yang di tentukan



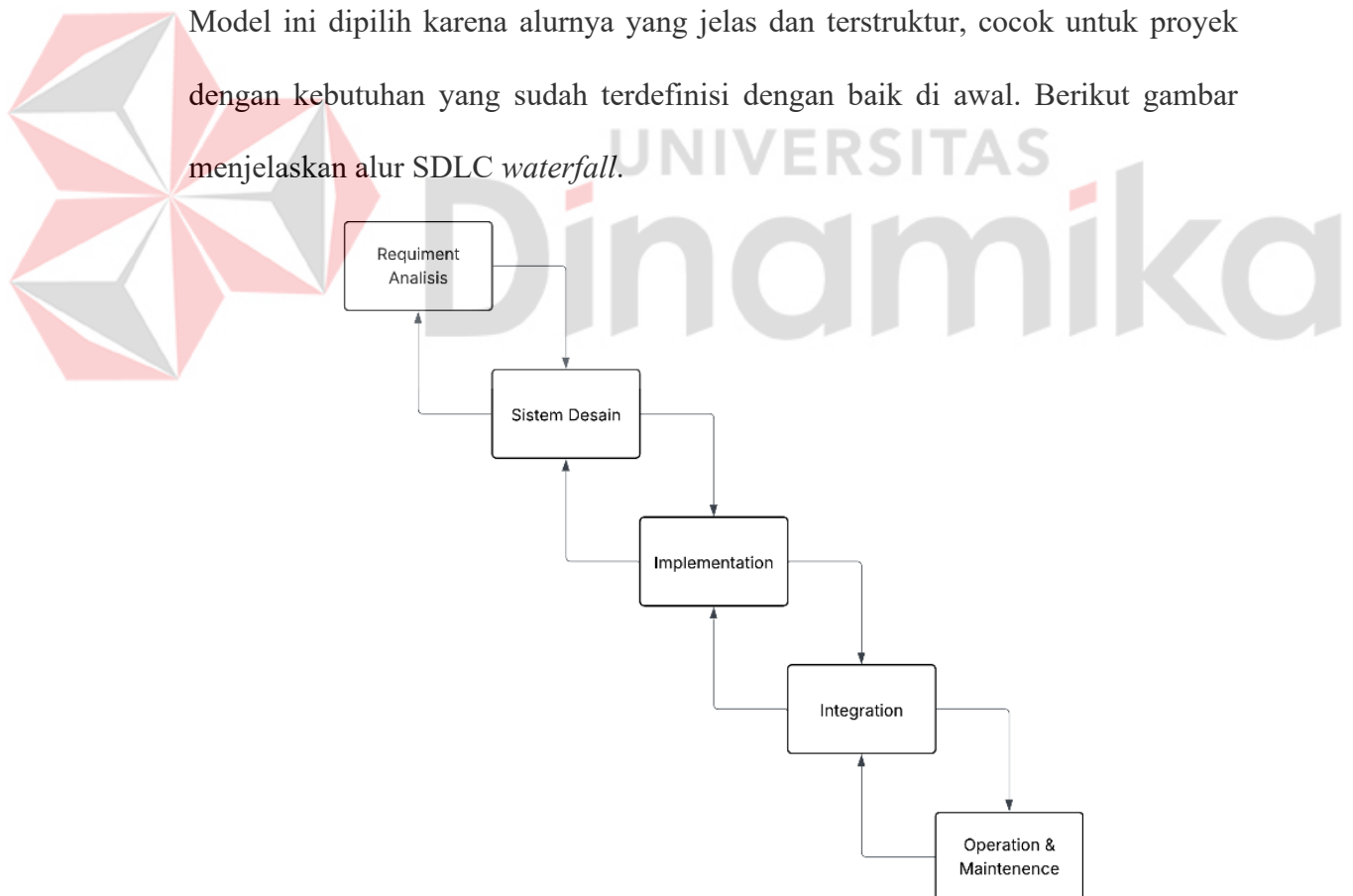
### 3.1.6 Maintenance

Proses keenam adalah tahap maintenance, pada tahap ini merupakan fase lanjutan setelah aplikasi digunakan pengguna. Proses ini dilakukan pemeliharaan secara berkala untuk menjaga performa dan stabilitas sistem. Selain itu, pemeliharaan juga mencakup pengembangan dan pembaruan fitur baru.

## 3.2 Waterfall

*Waterfall* adalah salah satu model SDLC dengan alur kerja yang bersifat sekuensial atau berurutan, di mana satu tahap harus selesai sepenuhnya sebelum lanjut ke tahap berikutnya, layaknya air terjun (Sholeh Kurniandini et al., 2022).

Model ini dipilih karena alurnya yang jelas dan terstruktur, cocok untuk proyek dengan kebutuhan yang sudah terdefinisi dengan baik di awal. Berikut gambar menjelaskan alur SDLC *waterfall*.



Gambar 3.2 Arsitektur *Waterfall*

### 3.3 Python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat *open-source* dan dikenal karena sintaksnya yang sederhana dan mudah dipelajari. Fleksibilitasnya yang tinggi serta dukungan pustaka (*library*) yang sangat lengkap menjadikannya pilihan populer untuk berbagai bidang, terutama dalam analisis data, pengembangan web, dan kecerdasan buatan. Dalam proyek ini, Python menjadi bahasa utama yang digunakan untuk membangun seluruh logika aplikasi.

### 3.4 Streamlit

Streamlit adalah sebuah *framework open-source* berbasis Python yang secara khusus dirancang untuk memudahkan pembuatan antarmuka pengguna (UI) yang interaktif untuk aplikasi data. Keunggulan utama Streamlit adalah kemampuannya untuk mengubah skrip data menjadi aplikasi web hanya dengan beberapa baris kode, tanpa memerlukan pengetahuan mendalam tentang HTML, CSS, atau JavaScript. *Framework* ini sangat cocok untuk pembuatan *dashboard*, visualisasi data, dan prototipe cepat.

### 3.5 Blackbox Testing

*Blackbox* adalah metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas aplikasi tanpa melihat atau menganalisis struktur kode internalnya (Pratama et al., 2023). Prosesnya melibatkan pemberian *input* pada sistem dan pengamatan *output* yang dihasilkan untuk memastikan kesesuaiannya dengan hasil yang diharapkan. Pada proyek ini, seluruh fungsionalitas diuji menggunakan metode *Blackbox* dan dinyatakan berhasil karena hasil aktual dari setiap skenario pengujian sesuai dengan hasil yang diharapkan.

## BAB IV

### DESKRIPSI PEKERJAAN

#### 4.1 Analisis

Tahap pertama adalah mengenali apa saja yang benar-benar dibutuhkan oleh sekolah, khususnya guru dan wali kelas, dalam melihat dan menganalisis nilai siswa. Dari hasil wawancara dan observasi, penulis menemukan bahwa guru masih kesulitan memantau nilai siswa secara menyeluruh karena data hanya tersedia dalam bentuk tabel panjang yang sulit dibaca cepat dan hanya bentuk *dashboard* pada setiap siswa. Maka dari itu, perancang sistem yang bisa menampilkan nilai siswa dalam bentuk grafik dan tampilan visual yang mudah sangat dibutuhkan.

##### 4.1.1 Identifikasi Masalah

Hasil dari wawancara dan observasi bahwa guru memiliki beberapa masalah berikut:

1. Proses pemantauan capaian akademik siswa saat ini tidak efisien karena mengharuskan guru, wali kelas, atau kepala sekolah memeriksa data nilai secara manual satu per satu. Dengan jumlah siswa dan data yang besar, metode ini sangat memakan waktu dan menyulitkan pihak sekolah untuk mendapatkan gambaran performa siswa secara cepat dan komprehensif.
2. Pengambilan keputusan strategis terkait siswa menjadi kurang objektif. Kebijakan penting—seperti penentuan siswa untuk program remedial, evaluasi metode mengajar, atau pelaporan kepada orang tua—sering kali hanya didasarkan pada asumsi atau pengalaman pribadi guru, bukan pada

analisis data nilai yang akurat. Hal ini berisiko menghasilkan intervensi yang tidak tepat sasaran.

#### 4.1.2 Analisis Kebutuhan

Sekolah sering menghadapi kendala dalam menampilkan dan memantau nilai siswa secara ringkas. Data biasanya masih berbentuk tabel panjang dan tidak interaktif, sehingga sulit dianalisis oleh guru dan wali kelas. Untuk mempercepat pengambilan keputusan pengguna membutuhkan sistem visual yang mudah dibaca dan bisa menampilkan informasi penting seperti rata-rata nilai dan kelulusan.

##### A. Kebutuhan Fungsional

Tahap kebutuhan fungsional dilakukan dengan mengumpulkan informasi langsung dari pihak sekolah melalui proses observasi dan wawancara. Dari kegiatan tersebut, diperoleh beberapa kebutuhan utama yang harus dipenuhi oleh sistem, antara lain fitur untuk menampilkan data nilai siswa secara ringkas, pemisahan antara siswa yang memenuhi dan tidak memenuhi KKM, filter berdasarkan tahun ajaran dan kelas, serta tampilan grafik yang memudahkan pembacaan data. Setiap kebutuhan tersebut kemudian dipetakan ke pengguna yang relevan, seperti guru, wali kelas, atau kepala sekolah, agar sistem benar-benar menjawab kebutuhan spesifik mereka. Proses ini sejalan dengan pendekatan *user-centered design*, di mana pengembangan sistem didasarkan pada peran dan kepentingan masing-masing pengguna. Dengan pemetaan ini, sistem tidak hanya dirancang berdasarkan apa yang bisa dibangun secara teknis, tetapi juga pada apa yang memang dibutuhkan dalam konteks operasional sekolah sehari-hari, sebagaimana dirinci dalam tabel 4.1:

Tabel 4.1 Tabel Kebutuhan Fungsional

Id kebutuhan	Nama fungsional	Deskripsi	Aktor	Prioritas
F-01	Menampilkan <i>dashboard</i> nilai	Sistem harus mampu mengambil, Menggabungkan, dan menampilkan data nilai siswa dari database.	Admin sistem	Tinggi
F-02	Filter nilai	Pengguna harus dapat memfilter data yang ditampilkan berdasarkan Tahun Ajaran, Kelas, dan nilai KKM.	Kepala sekolah	Tinggi
F-03	Menampilkan Nilai siswa	Sistem harus menampilkan ringkasan data kunci, yaitu jumlah total siswa dan rata-rata nilai untuk setiap mata pelajaran.	Kepala sekolah, guru mata pelajaran.	Tinggi
F-04	Visualisasi rata-rata nilai per kelas	Sistem harus menyajikan visualisasi dalam bentuk diagram batang ( <i>bar chart</i> ) yang menunjukkan rata-rata nilai siswa per kelas.	Kepala sekolah, guru mata pelajaran.	Sedang
F-05	Visualisasi tren Nilai	Sistem harus menyajikan visualisasi diagram garis ( <i>line chart</i> ) yang menampilkan tren rata-rata nilai di setiap tahun dan tingkat.	Kepala sekolah, guru mata pelajaran.	Sedang
F-06	Visualisasi status kelulusan	Sistem harus menyajikan visualisasi diagram pie ( <i>pie chart</i> ) yang menampilkan status kelulusan di setiap tahun dan tingkat.	Kepala sekolah, guru mata pelajaran.	Sedang

Id kebutuhan	Nama fungsional	Deskripsi	Aktor	Prioritas
F-07	Visualisasi <i>heatmap</i> Nilai mata pelajaran	Sistem harus menyajikan visualisasi diagram panas ( <i>heatmap</i> ) yang menampilkan rata rata nilai per kelas.	Kepala sekolah, guru mata pelajaran.	Sedang
F-08	Menampilkan Tabel data Siswa	Sistem harus dapat menyajikan table data siswa nilainya.	Kepala sekolah, guru mata pelajaran.	Tinggi

#### B. Kebutuhan Non-Fungsional

Selain fitur utama yang harus tersedia, sistem juga perlu memenuhi kebutuhan non-fungsional agar dapat digunakan secara nyaman dan optimal dalam lingkungan sekolah. Berdasarkan hasil pengamatan, pengguna menginginkan sistem yang cepat diakses, tidak membingungkan saat digunakan, dan mampu berjalan dengan lancar meskipun menampilkan banyak data nilai sekaligus. Oleh karena itu, perhatian khusus diberikan pada aspek kecepatan respons sistem, kemudahan navigasi, serta tampilan antarmuka yang sederhana namun informatif. Tampilan visual dirancang dengan warna kontras yang jelas, tombol dan filter diberi label yang mudah dipahami, serta grafik ditampilkan dengan gaya visual yang bersih. Kebutuhan ini sesuai dengan 4 kriteria *usability*, *Performance*, *Reliability*, dan *compatibility*. Kriteria tersebut diuraikan dalam tabel 4.2:

Tabel 4.2 Tabel Kebutuhan Non-Fungsional

Jenis Kebutuhan	Deskripsi Kebutuhan	Aktor	Analisis kebutuhan
<i>Ussability</i>	Aplikasi harus memiliki	Kepala sekolah & Guru.	Pengguna harus bisa mengakses

Jenis Kebutuhan	Deskripsi Kebutuhan	Aktor	Analisis kebutuhan
	antarmuka yang intuitif, mudah dipelajari, dan tidak membingungkan.		informasi dengan cepat tanpa memerlukan pelatihan teknis mendalam, sehingga adopsi sistem berhasil.
<i>Performance</i>	Sistem harus memuat data dan menampilkan visualisasi secara cepat dan responsif.	Kepala sekolah & Guru.	Nilai yang lambat akan menghambat efisiensi kerja dan membuat pengguna frustrasi, sehingga aplikasi bisa ditinggalkan.
<i>Reliability</i>	Data dan hasil kalkulasi (rata-rata, persentase) yang ditampilkan harus 100% akurat dan konsisten.	<i>Programmer</i> , Kepala sekolah & Guru.	<b>Kepala sekolah &amp; guru:</b> Membutuhkan data yang bisa dipercaya untuk membuat keputusan yang benar. <b>Programmer:</b> Bertanggung jawab untuk memastikan logika program bebas dari kesalahan.
<i>Compatibility</i>	Data dan hasil kalkulasi (rata-rata, persentase) yang ditampilkan harus 100% akurat dan konsisten.	<i>Programmer</i> , Kepala sekolah & Guru.	<b>Kepala sekolah &amp; guru:</b> Mendapatkan jaminan bahwa aplikasi dapat diakses dari perangkat kerja mereka. <b>Programmer:</b> Harus memastikan kode yang dibuat sesuai dengan standar web.



UNIVERSITAS  
Dinamika

Kebutuhan fungsional dan non-fungsional terpenuhi selanjutnya perlu identifikasi kebutuhan perangkat apa saja yang dibutuhkan dalam proses pembuatan *Dashboard*. Detail rinciannya bisa dilihat pada tabel 4.3:

Tabel 4.3 Kebutuhan Perangkat

Software	Hardware
1. Visual Studio Code	1. Jaringan Internet (kuota, wi-fi)
2. Internet Browser	2. Hard disk minimal 256 GB
3. Streamlit	3. RAM minimal 4GB

## 4.2 Desain

Pada tahap desain kita akan merancang bagaimana alur sistem tersebut bekerja. Mulai dari bagaimana data tersebut mengalir, bagaimana alur aplikasi, serta darimana data tersebut berasal sesuai dengan apa yang di dapat dalam sub bab analisis.

### 4.2.1 Alur system

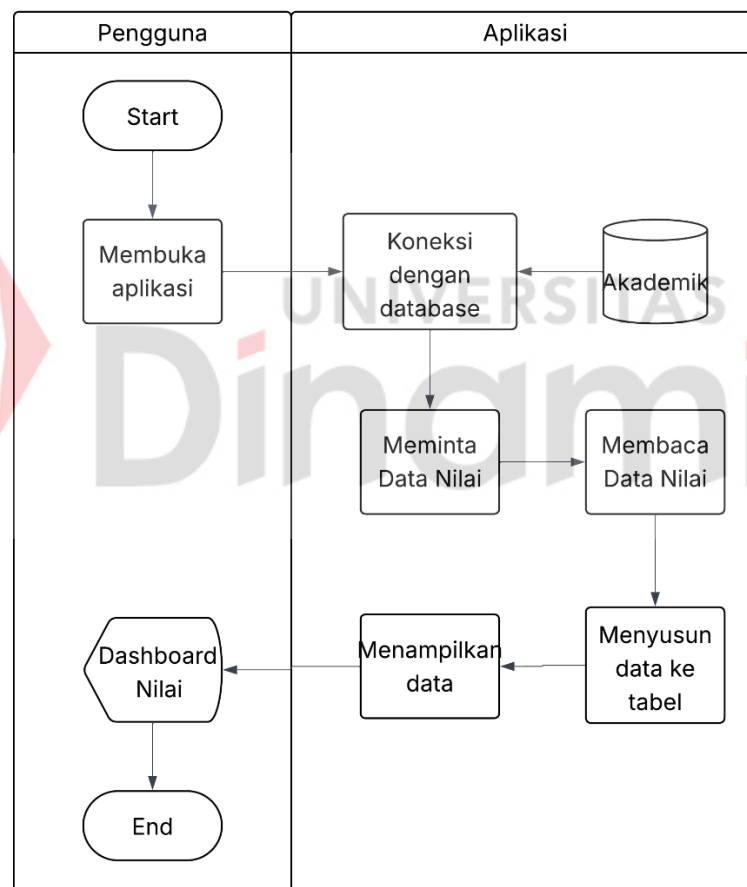
Alur sistem menggambarkan langkah-langkah yang dilakukan pengguna, mulai dari membuka dashboard, memilih filter, hingga melihat visualisasi data yang ditampilkan. Tujuannya agar sistem dapat digunakan dengan alur logis dan mudah dipahami oleh pengguna non-teknis seperti guru dan wali kelas.

#### A. Alur Dashboard Nilai

Pada Alur system proses yang terjadi saat pengguna pertama kali membuka aplikasi hingga *dashboard* utama ditampilkan. Alur ini dimulai saat pengguna membuka aplikasi, yang secara otomatis memicu serangkaian proses di



sisi aplikasi. Sistem akan langsung melakukan koneksi dengan *database* Akademik untuk mengakses sumber data. Setelah terhubung, aplikasi akan meminta data nilai, kemudian membaca informasi tersebut, dan dilanjutkan dengan proses penting yaitu menyusun data ke dalam format tabel agar terstruktur dan siap untuk ditampilkan. Terakhir, data yang sudah rapi ini akan ditampilkan oleh aplikasi, sehingga pengguna dapat melihat hasil akhirnya berupa Dashboard Nilai yang informatif. Untuk diagram alur system dapat dilihat pada Gambar 4.1:

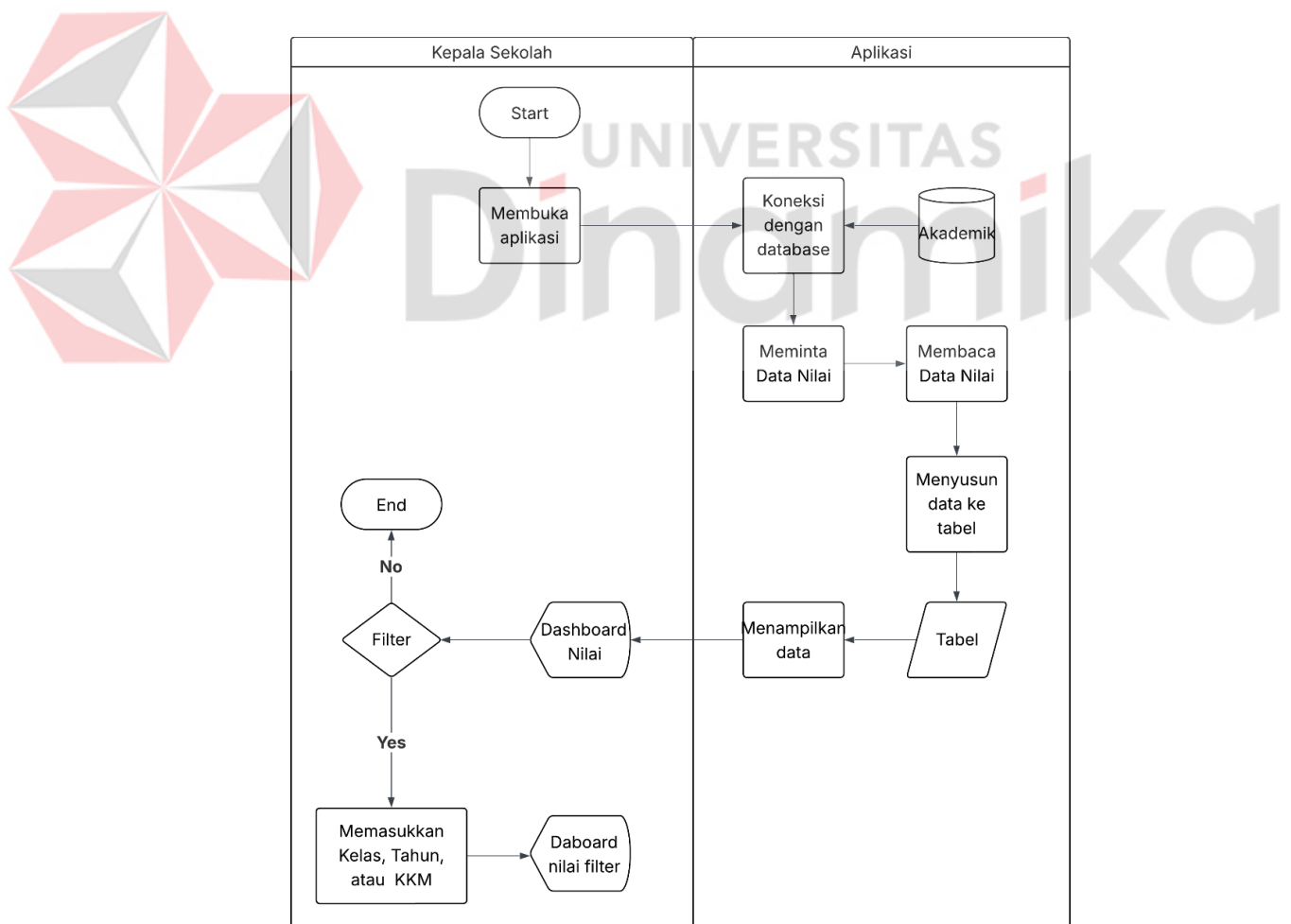


Gambar 4.1 Alur Dashboard Nilai

#### B. Alur Filter Nilai

Alur sistem berkerja saat pengguna berinteraksi dengan fitur filter untuk melihat data yang lebih spesifik. Proses dimulai saat pengguna membuka aplikasi,

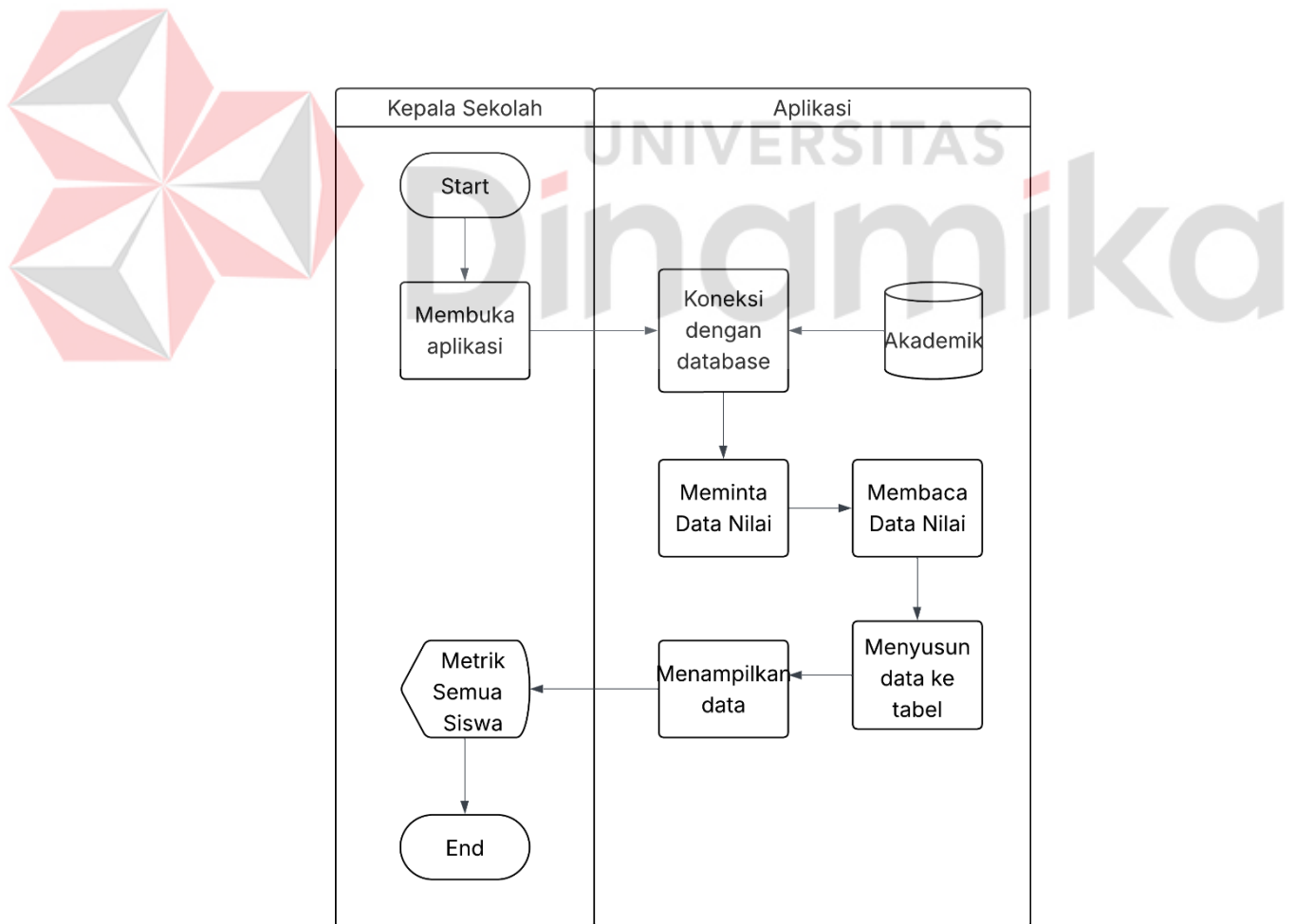
yang memicu sistem untuk melakukan proses di belakang layar: melakukan koneksi dengan database, meminta dan membaca data nilai, menyusun data ke dalam tabel, dan akhirnya menampilkan dashboard awal. Dari dashboard ini, pengguna diberikan pada pilihan untuk memfilter data. Jika pengguna memilih akan memasukkan kriteria yang diinginkan seperti Kelas, Tahun, atau KKM, lalu menekan Enter. Aksi ini memicu aplikasi untuk menjalankan proses filter pada tabel data yang sudah ada. Hasilnya, aplikasi akan menampilkan data yang sudah tersaring, memperbarui tampilan Dashboard Nilai sesuai dengan kriteria yang dipilih pengguna, alur dapat dilihat pada Gambar 4.2:



Gambar 4.2 Alur Filter Nilai

### C. Dashboard Matriks Nilai Siswa

Diagram ini merinci alur kerja sistem saat menampilkan ringkasan data atau metrik kunci untuk seluruh siswa. Proses ini diawali saat Aktor membuka aplikasi. Tindakan ini secara otomatis memicu aplikasi untuk menjalankan serangkaian proses di latar belakang, dimulai dengan melakukan koneksi ke *database* Akademik. Setelah terhubung, aplikasi akan meminta data nilai yang diperlukan, membaca informasi tersebut, lalu menyusun data ke dalam format tabel yang terstruktur. Setelah data siap, aplikasi akan menampilkan data tersebut kepada pengguna dalam bentuk Metrik Semua Siswa, yang menyajikan ringkasan performa akademik secara keseluruhan. Untuk alur dapat dilihat pada Gambar 4.3:

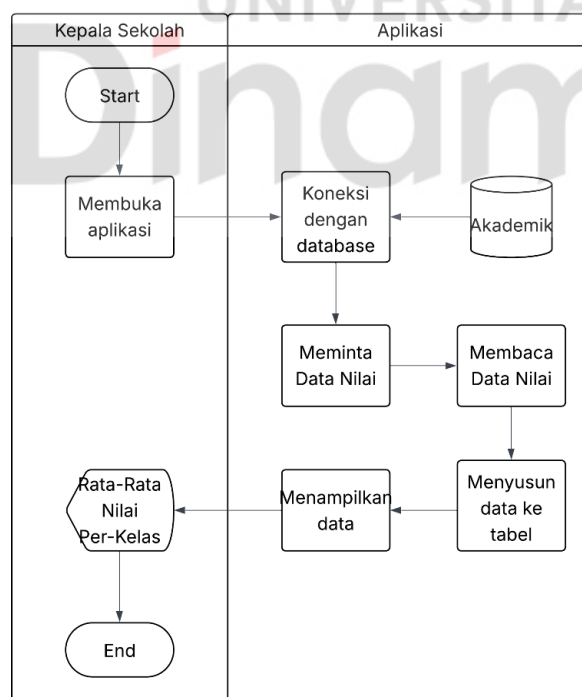


Gambar 4.3 Alur Metriks Nilai

#### D. Diagram Bar Nilai Per-Kelas

Diagram ini menggambarkan proses menampilkan visualisasi perbandingan nilai rata-rata di setiap kelas. Proses ini diawali saat Aktor membuka aplikasi, yang langsung memicu sistem untuk bekerja di belakang layar. Aplikasi akan secara otomatis melakukan koneksi dengan *database* Akademik, kemudian meminta dan membaca data nilai yang diperlukan. Selanjutnya, data tersebut akan disusun ke dalam format tabel yang rapi dan terstruktur. Secara spesifik untuk grafik ini, aplikasi akan mengelompokkan data berdasarkan kelas, lalu menghitung nilai rata-rata untuk setiap kelas tersebut. Hasil perhitungan inilah yang kemudian ditampilkan oleh aplikasi ke Aktor berupa diagram Rata-Rata Nilai Per-Kelas.

Untuk alur dapat dilihat pada Gambar 4.4:

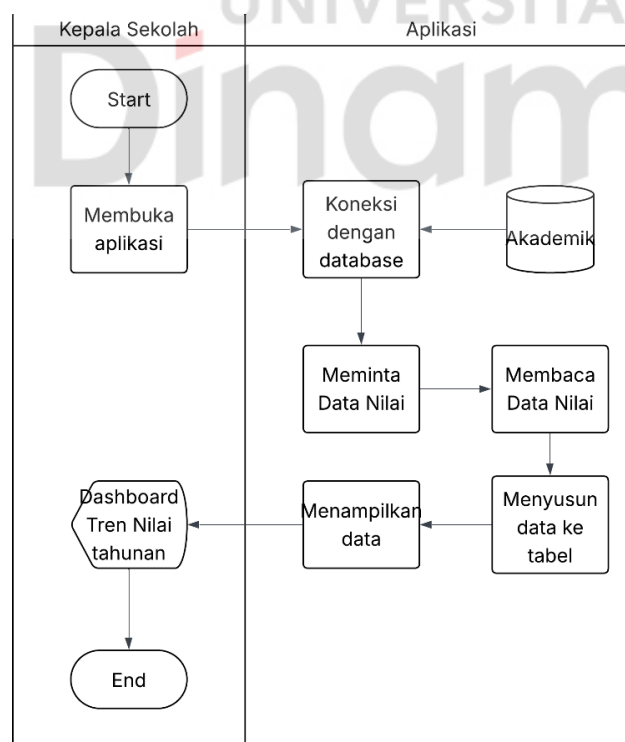


Gambar 4.4 Alur Diagram Bar Nilai per-Kelas

#### E. Line Chart Tren Nilai Tahunan

Diagram ini menjelaskan alur kerja sistem untuk menyajikan visualisasi tren nilai akademik dari tahun ke tahun. Proses ini diawali saat Aktor membuka aplikasi, yang secara otomatis memicu serangkaian proses di belakang layar. Sistem akan melakukan koneksi dengan *database* Akademik, kemudian meminta dan membaca data nilai. Setelah itu, data tersebut akan disusun ke dalam format tabel yang rapi. Untuk menyajikan tren, aplikasi secara khusus akan mengolah data tersebut dengan mengelompokkannya berdasarkan tahun ajaran. Hasil olahan data inilah yang kemudian ditampilkan sebagai Dashboard Tren Nilai Tahunan ke Aktor dalam bentuk visual berupa diagram garis. Untuk alur dapat dilihat pada Gambar

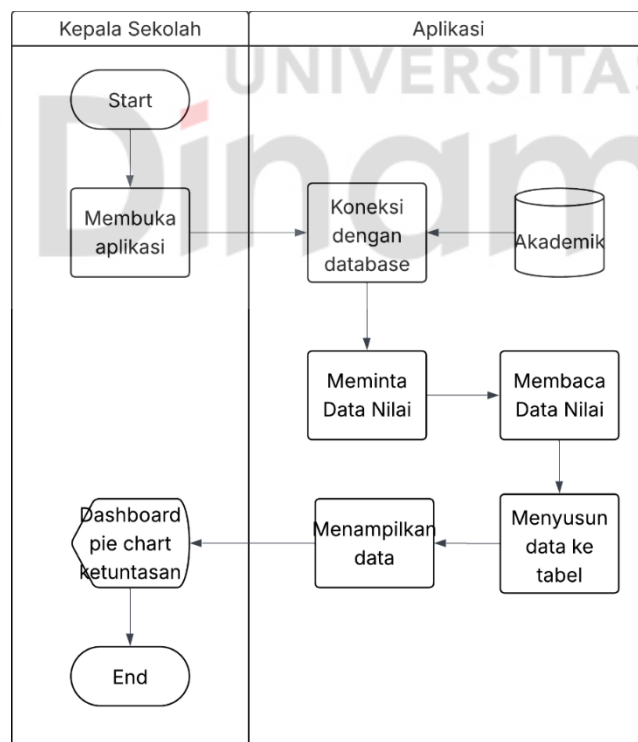
4.5:



Gambar 4.5 Alur Line Chart Tren Nilai Tahunan

#### F. Pie Chart Status Kelulusan

Diagram alur ini merinci proses untuk menampilkan diagram persentase status ketuntasan siswa. Alur ini diawali saat Aktor membuka aplikasi, yang memicu sistem untuk menjalankan serangkaian proses di belakang layar. Aplikasi akan memulai dengan melakukan koneksi dengan *database* Akademik, kemudian meminta dan membaca data nilai. Data yang diterima kemudian akan disusun ke dalam format tabel yang terstruktur. Khusus untuk diagram ini, aplikasi akan menghitung jumlah siswa untuk setiap kategori status lalu mengubahnya menjadi format persentase. Hasil persentase tersebut ditampilkan ke Aktor dalam bentuk visual Dashboard pie chart ketuntasan. Untuk alur dapat di lihat di Gambar 4.6:

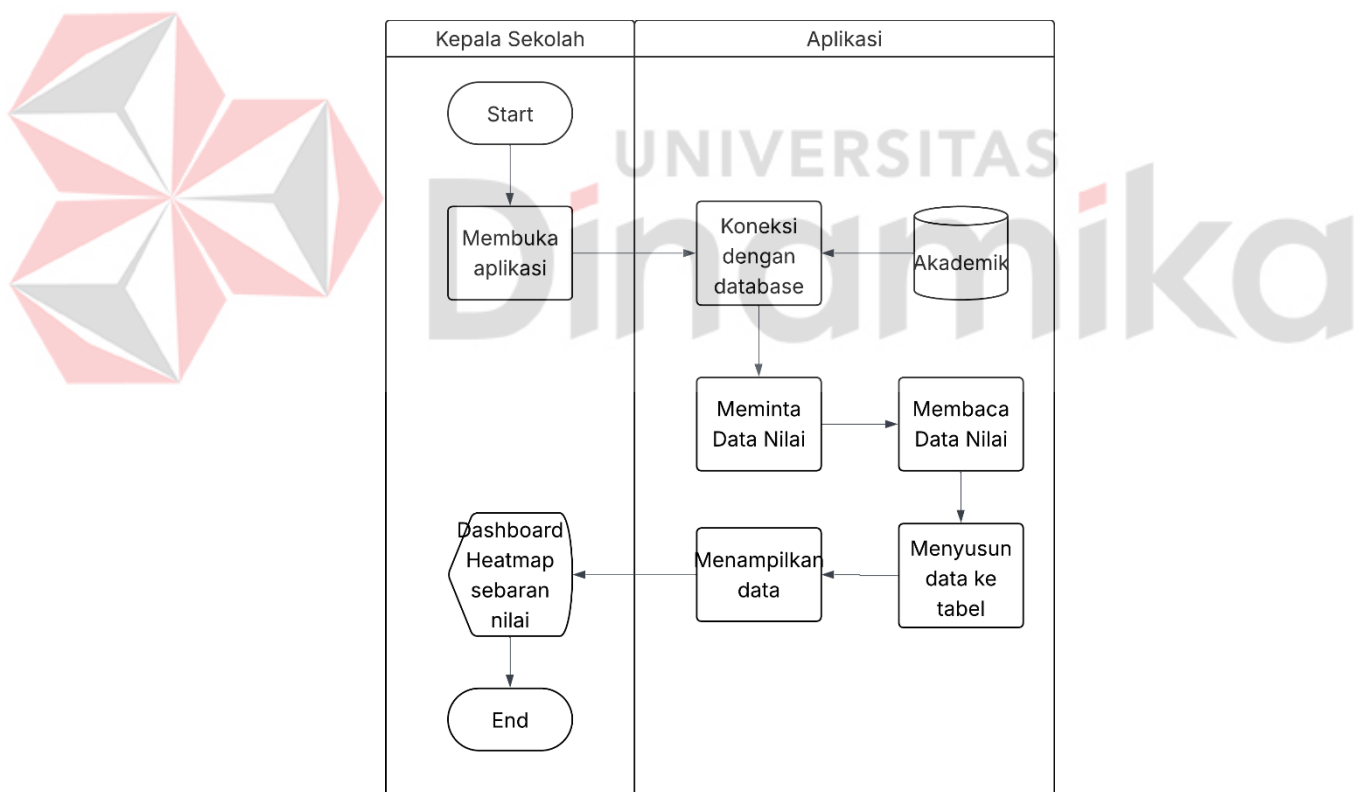


Gambar 4.6 Pie Chart Status Kelulusan

#### G. Heatmap Sebaran Nilai

Diagram ini menjelaskan proses untuk menampilkan visualisasi *heatmap*

yang membandingkan nilai setiap mata pelajaran di semua kelas. Alur kerja ini diawali saat Aktor membuka aplikasi, yang memicu sistem untuk menjalankan proses di belakang layar, yaitu melakukan koneksi dengan *database*, meminta dan membaca data nilai, lalu menyusun data ke dalam format tabel. Untuk membuat *heatmap*, aplikasi melakukan pengolahan data yang lebih kompleks dengan mengubah tabel tersebut menjadi format matriks. Matriks inilah yang kemudian ditampilkan aplikasi dalam bentuk Dashboard *Heatmap* sebaran nilai, di mana warna yang berbeda di setiap sel merepresentasikan tingkat nilai yang berbeda. Untuk alurnya dapat dilihat pada Gambar 4.7

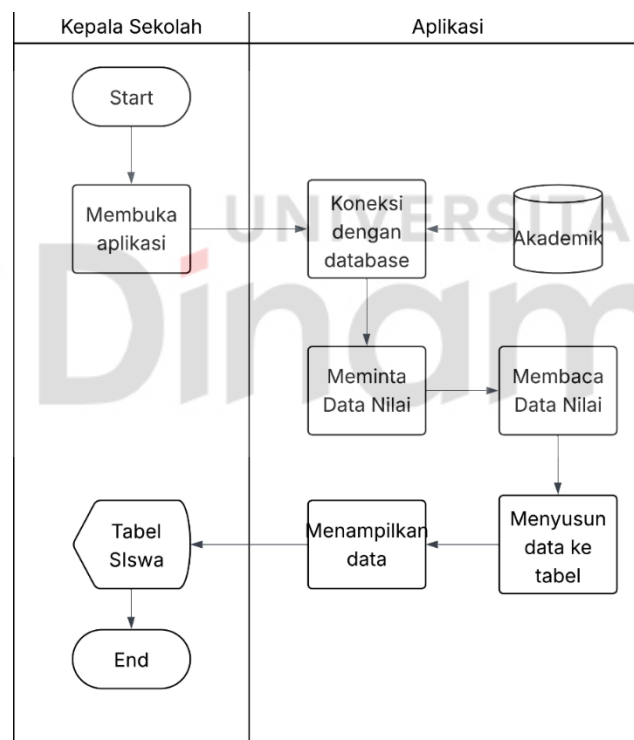


Gambar 4.7 *Heatmap* Sebaran Nilai

#### H. Alur sistem Tabel nilai siswa

Diagram ini menggambarkan alur kerja dasar untuk menampilkan data

siswa dalam bentuk tabel. Prosesnya dimulai ketika Aktor membuka aplikasi, yang langsung memicu serangkaian tindakan di sisi aplikasi. Sistem akan secara otomatis melakukan koneksi dengan *database* Akademik untuk mengakses sumber data, kemudian meminta dan membaca data nilai yang diperlukan. Setelah informasi diperoleh, data tersebut akan disusun ke dalam format tabel yang rapi dan terstruktur. Terakhir, tabel yang sudah siap ini akan ditampilkan oleh aplikasi sehingga Aktor dapat melihat Tabel nilai siswa secara rinci. Untuk alur sistem dapat dilihat pada Gambar 4.8:



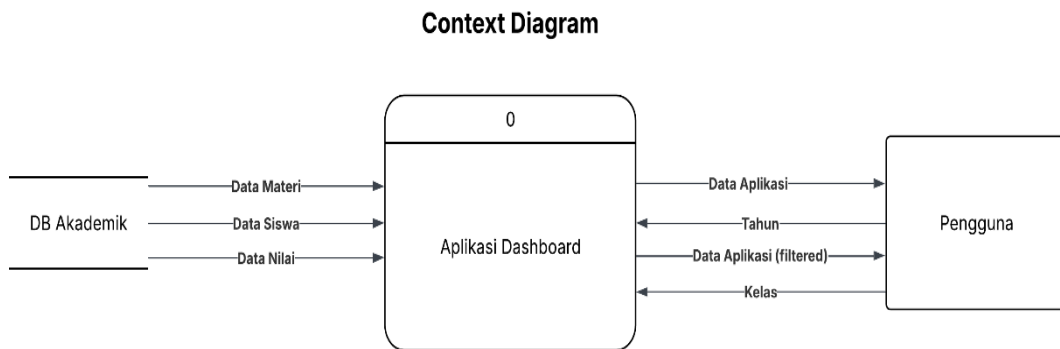
Gambar 4. 8 Tabel nilai siswa

#### 4.2.2 Context diagram

*Context diagram* menggambarkan alur data dalam aplikasi, mulai dari interaksi pengguna hingga pemanfaatan dataset yang digunakan. Diagram ini menunjukkan bagaimana data mengalir ke dalam dan ke luar sistem secara



keseluruhan. Alur data tersebut ditampilkan pada Gambar 4.9.



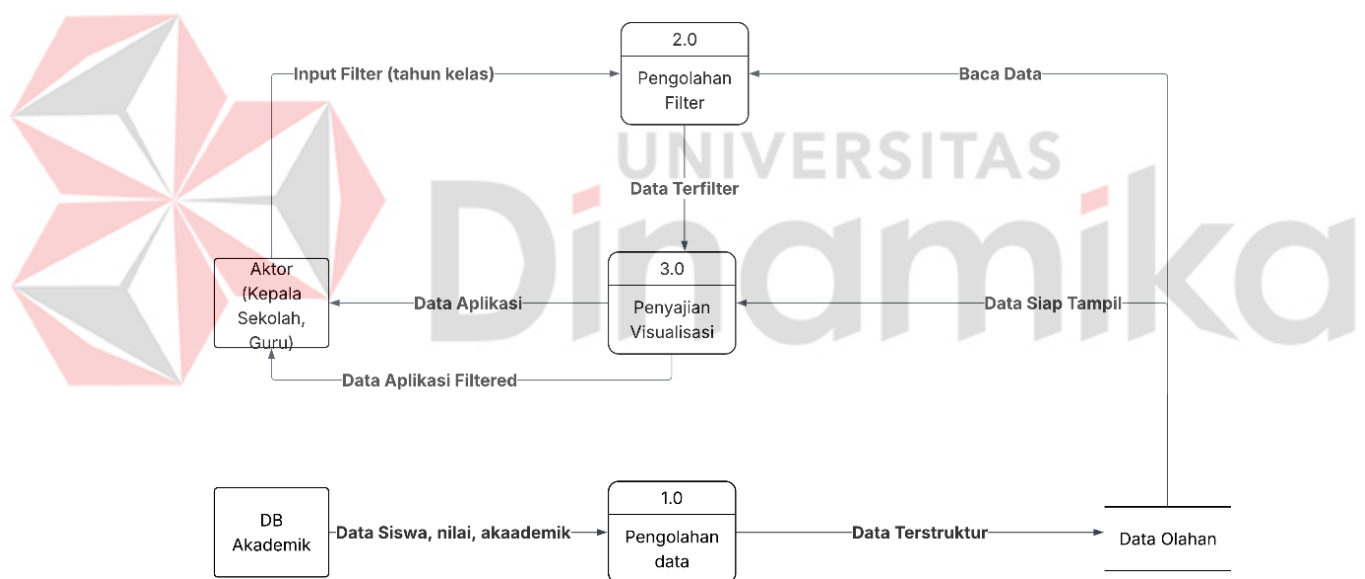
Gambar 4.9 Context Diagram

#### 4.2.3 DFD level 0

Data Flow Diagram (DFD) Level 0 memberikan gambaran yang lebih terperinci tentang bagaimana aliran data terjadi di dalam sistem dashboard nilai siswa. Jika pada *Context Diagram* sistem hanya digambarkan sebagai satu proses utama, maka dalam DFD Level 0 ini proses tersebut dibagi menjadi tiga sub-proses inti, yaitu 1.0 Pengolahan Data, 2.0 Pengolahan Filter, dan 3.0 Penyajian Visualisasi. Diagram ini memperlihatkan bagaimana data bergerak dari sumber, diproses, hingga ditampilkan kepada pengguna. Terdapat dua entitas eksternal yang berinteraksi dengan sistem: DB Akademik sebagai sumber data siswa, nilai, dan akademik; serta Aktor yang mewakili pengguna sistem seperti Kepala Sekolah dan Guru.

Alur dimulai dari DB Akademik yang mengirimkan data mentah ke proses 1.0 Pengolahan Data, yang bertugas menyusun dan menyiapkan data menjadi bentuk terstruktur. Data ini kemudian disimpan sementara di penyimpanan internal (Data Olahan) sebelum masuk ke tahap berikutnya. Pengguna akan memberikan

masukan berupa filter (misalnya memilih tahun ajaran atau kelas) ke proses 2.0 Pengolahan Filter, yang akan membaca dan menyesuaikan data berdasarkan input tersebut. Hasil dari proses ini berupa data yang telah disaring (Data Terfilter), yang kemudian masuk ke 3.0 Penyajian Visualisasi. Pada tahap akhir ini, sistem menampilkan data visual seperti grafik atau tabel, yang dikembalikan kepada Aktor sebagai Data Aplikasi Filtered dan Data Siap Tampil. Keseluruhan proses ini menunjukkan bagaimana sistem memproses data secara bertahap untuk menghasilkan informasi yang mudah dipahami oleh pengguna. Untuk DFD level 0 dapat dilihat pada Gambar 4.10:



Gambar 4.10 DFD Level 0

#### 4.2.4 Pemodelan data

Pemodelan data adalah proses merancang struktur data dalam sebuah sistem untuk memastikan data tersimpan dengan rapi dan mudah dikelola. Dalam penelitian ini, pemodelan data digunakan untuk menggambarkan keterkaitan antara

siswa, nilai, kelas, dan materi agar sistem dashboard dapat menampilkan informasi secara efisien dan akurat. Untuk struktur nya dapat dilihat di Gambar 4.11:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype
---  -
0   student_id  150 non-null   int64
1   name        150 non-null   object
2   grade       150 non-null   object
3   semester    150 non-null   int64
4   year        150 non-null   int64
5   subject     150 non-null   object
6   score       150 non-null   int64
7   status      150 non-null   object
dtypes: int64(4), object(4)
memory usage: 9.5+ KB
```

Gambar 4.11 Pemodelan data

Pada data diatas terdapat 7 properti, masing-masing akan menyimpan sebuah nilai yang memiliki tipe data string, dan ada properti yang tidak memiliki data/*null*. Berikut penjelasan setiap properti pada Gambar 4.11:

1. `student_id`: Properti ini berfungsi sebagai nomor identifikasi unik untuk setiap siswa. Tipe datanya adalah `int64`, yaitu bilangan bulat.
2. `name`: Menyimpan nama lengkap dari siswa. Tipe datanya adalah `object`, yang dalam konteks ini merujuk pada teks atau *string*.
3. `grade`: Menunjukkan tingkatan kelas siswa (contoh: 'VII A', 'VIII B'). Tipe datanya adalah `object` atau teks (*string*).
4. `semester`: Menyimpan data semester saat nilai dicatat. Tipe datanya adalah `int64` atau bilangan bulat.
5. `year`: Menunjukkan tahun ajaran dari data nilai tersebut. Tipe datanya adalah `int64` atau bilangan bulat.

8. int64 atau bilangan bulat.
9. subject: Menyimpan nama mata pelajaran yang dinilai (contoh: 'Matematika', 'Bahasa Indonesia'). Tipe datanya adalah
10. object atau teks (*string*).
11. score: Merupakan nilai atau skor numerik yang diperoleh siswa untuk mata pelajaran tertentu. Tipe datanya adalah
12. int64 atau bilangan bulat.
13. status: Menunjukkan status kelulusan siswa berdasarkan nilai yang diperoleh (contoh: 'Lulus', 'Remedial'). Tipe datanya adalah object atau teks (*string*).

### 4.3 Implementation

Bab ini merinci realisasi teknis dari perancangan sistem dashboard nilai siswa. Implementasi proyek ini dibangun menggunakan bahasa pemrograman Python. Untuk antarmuka pengguna (UI) berbasis web, digunakan framework Streamlit yang memungkinkan pembuatan aplikasi data interaktif dengan sintaks sederhana. Seluruh proses, mulai dari pengambilan data hingga visualisasi, mengandalkan beberapa library pendukung utama.

#### 4.3.1 Menampilkan Data Akademik

Hasil implementasi fungsionalitas ini adalah sebuah sistem yang mampu secara otomatis mengambil, menggabungkan, dan menyajikan data dari berbagai tabel di *database* (nilai, siswa, materi) ke dalam sebuah struktur data terpadu, seperti yang ditunjukkan pada Gambar 4.12. Pembahasan dari sisi teknis, proses ini diwujudkan melalui koneksi ke *database* PostgreSQL menggunakan *library* SQLAlchemy yang kemudian datanya dimuat ke dalam format Pandas DataFrame.

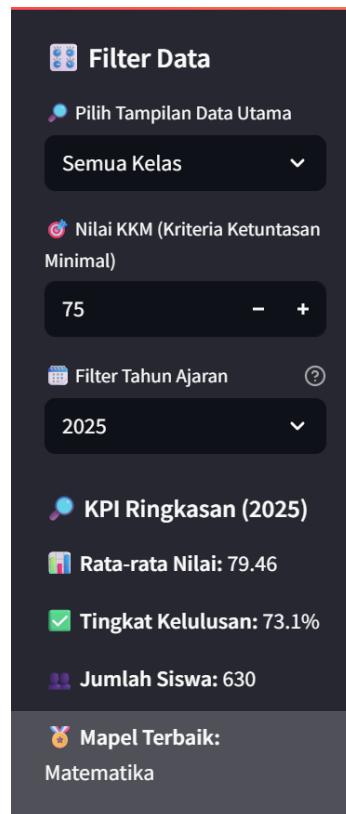
Fungsionalitas ini merupakan fondasi dari seluruh aplikasi dan secara langsung mengatasi masalah inti di sekolah, yaitu proses rekapitulasi data manual yang terfragmentasi dan lambat. Dengan mengotomatisasi proses agregasi, sistem ini sejalan dengan temuan (Fahrezy et al., 2020) yang menyoroti pentingnya digitalisasi untuk monitoring hasil belajar yang lebih akurat dan efisien

	id_siswa	nama_siswa	tahun_ajaran	id_kelas	kelas	id_materi	materi	nilai	status	tingkat
0	1	Siswa VII-A-2023-1	2023	1	VII-A	1	Matematika	87	Tugas	7
1	1	Siswa VII-A-2023-1	2023	1	VII-A	1	Matematika	73	Quiz	7
2	1	Siswa VII-A-2023-1	2023	1	VII-A	1	Matematika	76	UTS	7
3	1	Siswa VII-A-2023-1	2023	1	VII-A	1	Matematika	80	UAS	7
4	1	Siswa VII-A-2023-1	2023	1	VII-A	2	IPA	77	Tugas	7
...	...	...	...	...	...	...	...	...	...	...
52915	1890	Siswa IX-G-2025-30	2025	21	IX-G	6	Bahasa Jawa	63	UAS	9
52916	1890	Siswa IX-G-2025-30	2025	21	IX-G	7	Pendidikan Agama	83	Tugas	9
52917	1890	Siswa IX-G-2025-30	2025	21	IX-G	7	Pendidikan Agama	81	Quiz	9
52918	1890	Siswa IX-G-2025-30	2025	21	IX-G	7	Pendidikan Agama	85	UTS	9
52919	1890	Siswa IX-G-2025-30	2025	21	IX-G	7	Pendidikan Agama	91	UAS	9

Gambar 4.12 Menampilkan Data Akademik

### 4.3.2 Tampilan Sidebar

Pada tampilan dashboard yang dibangun menggunakan Streamlit, sidebar berfungsi sebagai pusat kendali bagi pengguna dalam menyaring dan mengatur data yang akan dianalisis. Gambar 4.13 menunjukkan antarmuka sidebar yang dirancang secara intuitif agar mudah dioperasikan oleh pengguna non-teknis seperti guru dan kepala sekolah. Sidebar ini terbagi menjadi dua bagian utama, yaitu filter data dan metrik-metrik nilai siswa. Kedua komponen ini bekerja secara sinergis dalam mengatur dan menyajikan informasi yang sesuai kebutuhan pengguna, serta menjadi titik awal dalam proses analisis nilai akademik.



Gambar 4.13 Sidebar

A. Filter Data

Hasil dari fungsionalitas ini adalah tersedianya panel filter interaktif pada *sidebar* aplikasi pada Gambar 4.13 yang memungkinkan pengguna untuk menyaring data secara dinamis. Pembahasannya, fitur ini diimplementasikan menggunakan komponen *widget* bawaan dari *framework* Streamlit, seperti `selectbox` dan `number_input`, yang memberikan pengguna kemampuan untuk memfilter data berdasarkan Tahun Ajaran, Kelas, dan nilai KKM. Implementasi ini secara langsung menjawab kebutuhan fungsional F-02 dan secara kualitatif mengubah proses analisis data dari yang semula statis menjadi dinamis. Guru dan kepala sekolah kini dapat melakukan eksplorasi data secara mandiri untuk mendapatkan wawasan spesifik tanpa proses manual yang rumit.

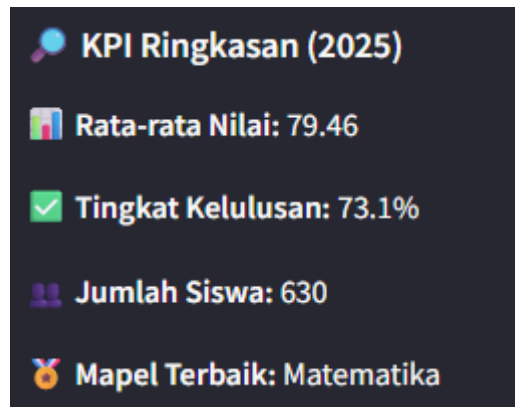


Gambar 4.14 Filter Data

#### B. Metrik-metrik Nilai Siswa

Sistem berhasil menampilkan ringkasan data kunci atau *Key Performance Indicators* (KPI) yang ringkas dan informatif, seperti terlihat pada Gambar 4.14.

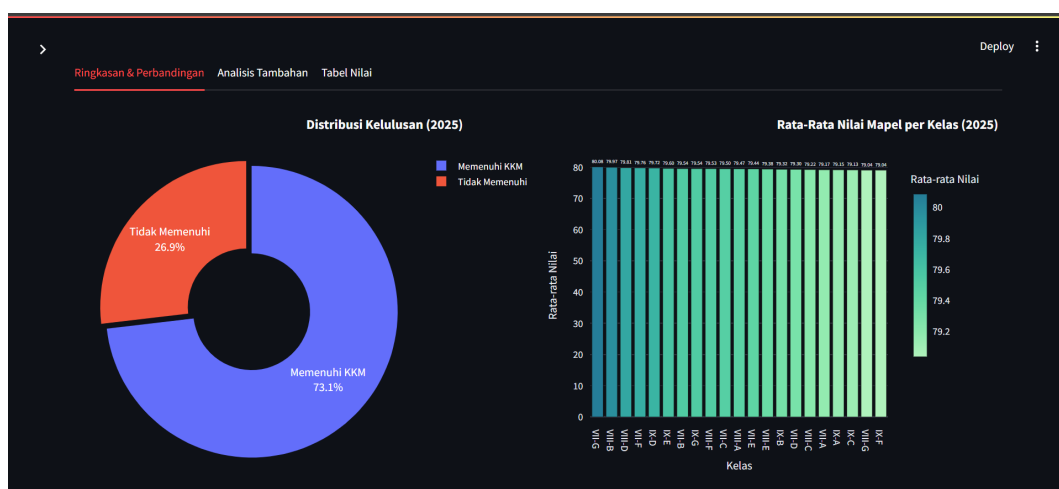
KPI yang disajikan meliputi Rata-rata Nilai, Tingkat Kelulusan, Jumlah Siswa, dan Mapel Terbaik. Pembahasan dari fungsionalitas ini, setiap metrik dikalkulasi secara kuantitatif dan *real-time* dari data yang telah difilter menggunakan fungsi agregasi pada Pandas DataFrame. Fitur ini dirancang untuk memberikan gambaran performa akademik secara cepat kepada kepala sekolah, sehingga mendukung proses pengambilan keputusan yang lebih tanggap dan berbasis data, sesuai dengan tujuan yang dirumuskan.



Gambar 4.15 Metrik-Metrik Kunci Nilai Siswa

### 4.3.3 View Ringkasan & Perbandingan

Tampilan ini merupakan bagian utama dari dashboard yang menyajikan dua komponen visual utama secara berdampingan, yaitu grafik status kelulusan siswa dan grafik rata-rata nilai per kelas. Kedua visualisasi ini ditampilkan dalam tab bernama “Ringkasan & Perbandingan”, seperti ditunjukkan pada Gambar 4.16. Tampilan ini memungkinkan pengguna memperoleh wawasan akademik dari dua sisi: seberapa besar tingkat kelulusan siswa secara agregat dan bagaimana distribusi rata-rata nilai di antara kelas-kelas yang ada.

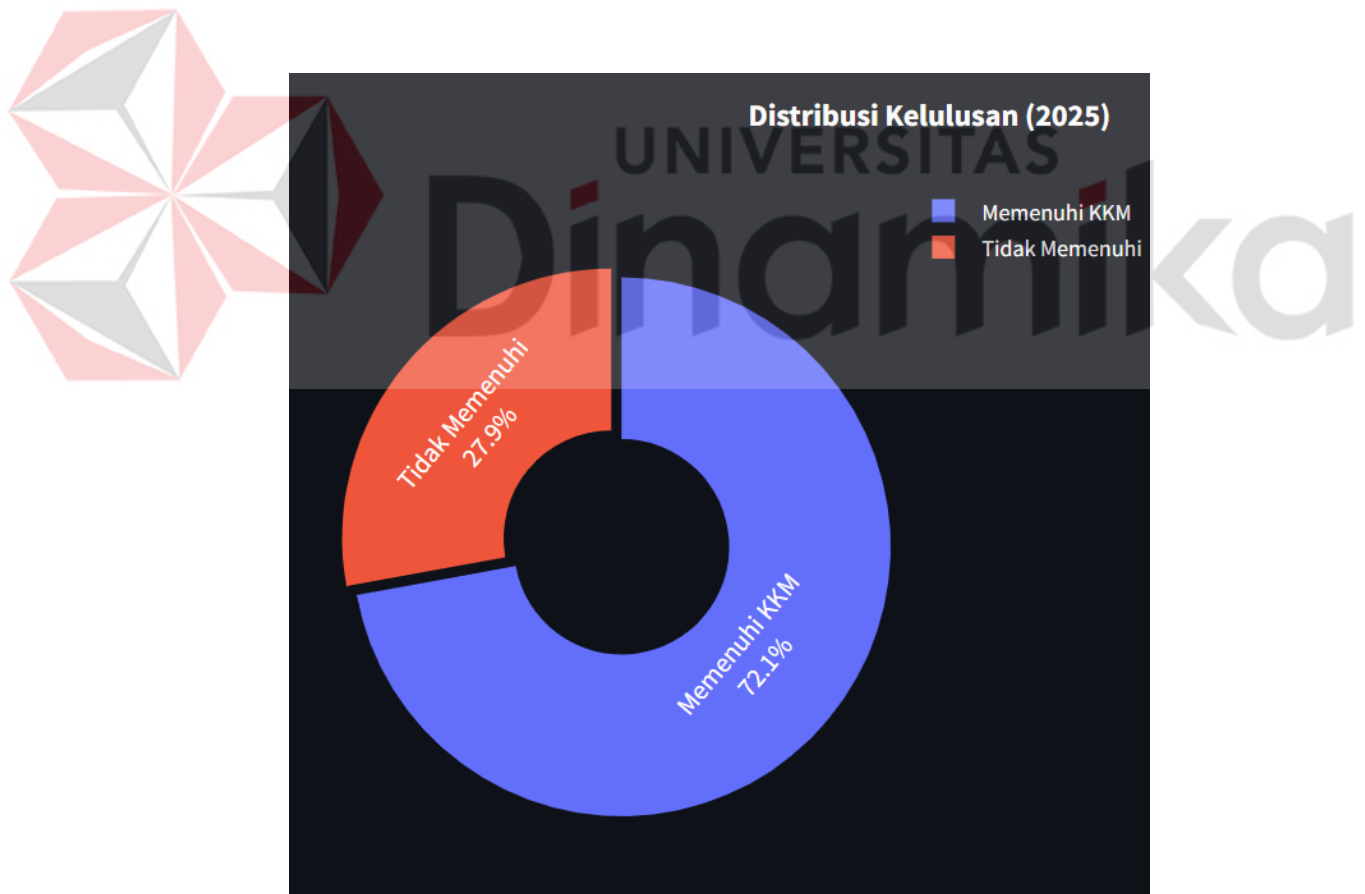


Gambar 4.16 Tabs Ringkasan dan Perbandingan



#### A. Visualisasi Status Kelulusan

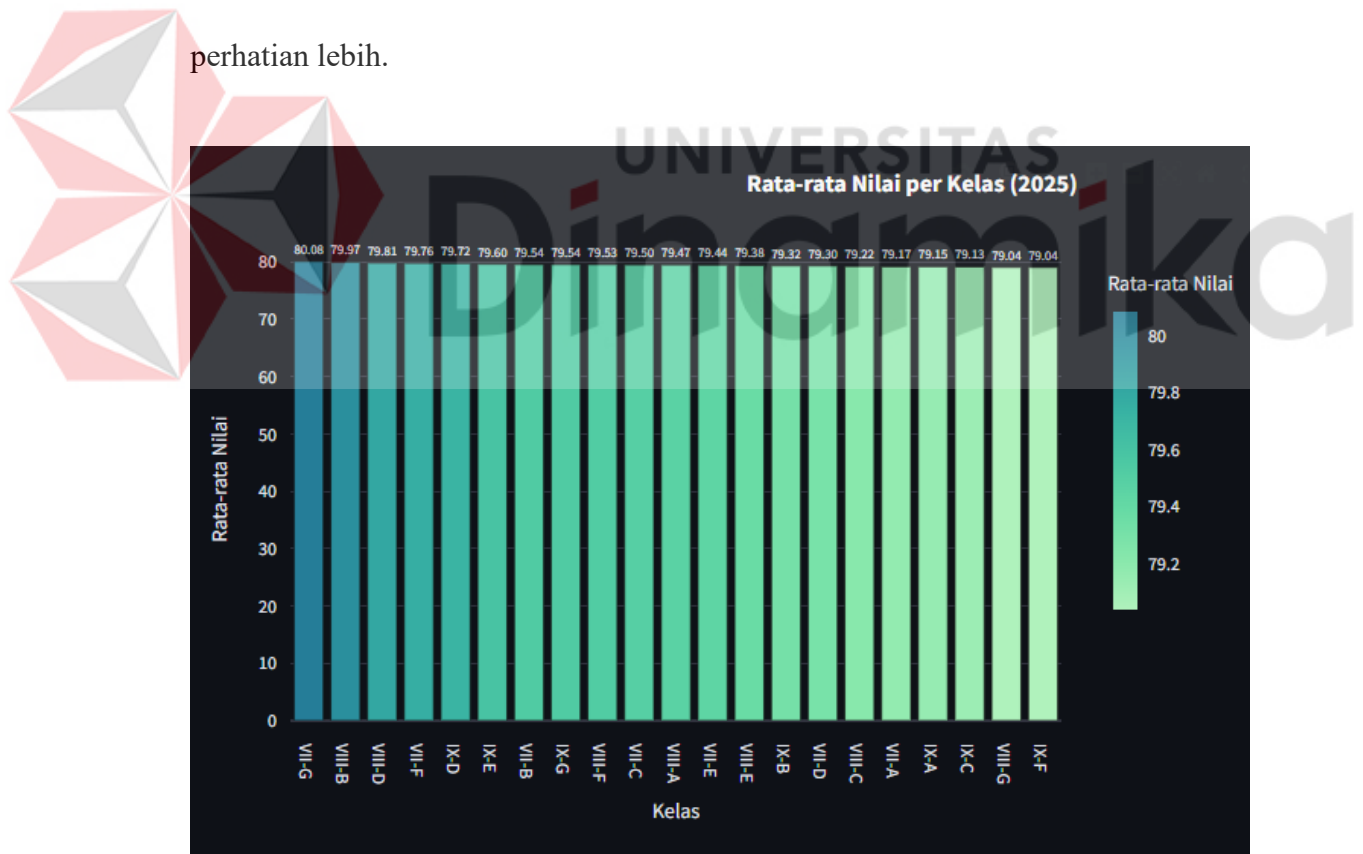
Fungsionalitas ini berhasil diwujudkan dalam bentuk visualisasi diagram lingkaran (*pie chart*) yang menampilkan proporsi status kelulusan siswa di setiap tahun dan tingkat Gambar 4.17. Pembahasan dari implementasi ini adalah dengan menghitung jumlah kemunculan setiap kategori dalam kolom status (misalnya, 'Memenuhi KKM' dan 'Perlu Perbaikan') berdasarkan nilai KKM yang diinput pengguna. Data proporsi ini kemudian digunakan untuk menghasilkan diagram yang mudah dibaca. Fitur ini secara langsung memenuhi kebutuhan fungsional F-06 dengan memberikan umpan balik instan mengenai tingkat ketuntasan belajar siswa secara keseluruhan.



Gambar 4. 17 Grafik Distribusi Status Kelulusan

## B. Visualisasi Rata-Rata Nilai per Kelas

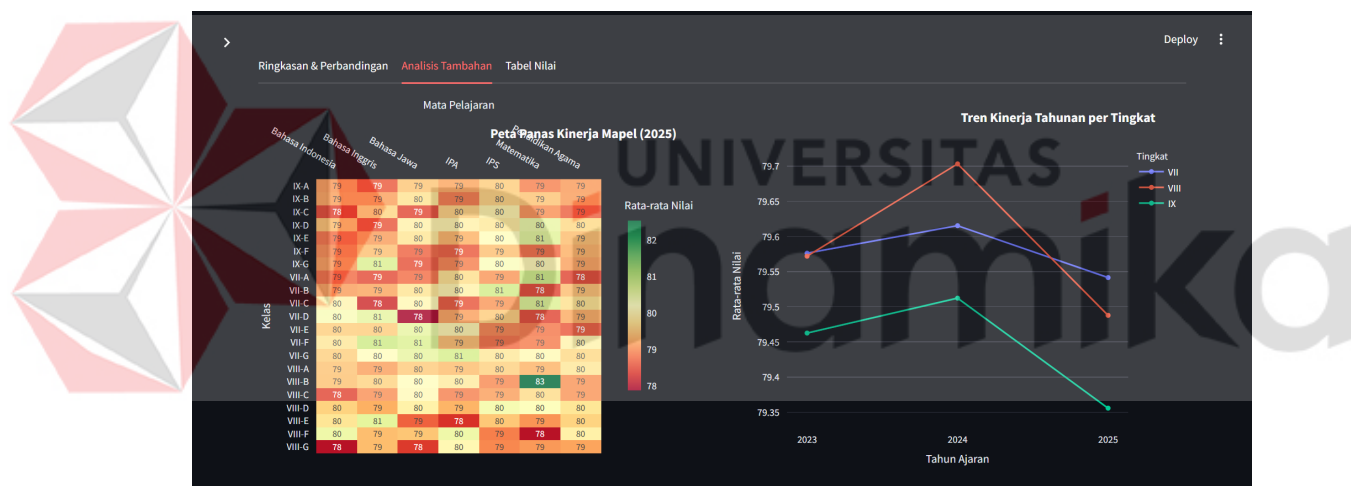
Hasil dari fungsionalitas ini adalah sebuah visualisasi diagram batang yang menyajikan perbandingan rata-rata nilai siswa untuk setiap kelas secara jelas Gambar 4.15. Dalam pembahasannya, visualisasi ini diimplementasikan dengan mengelompokkan data pada DataFrame berdasarkan kolom kelas, kemudian menghitung nilai rata-rata dari kolom skor untuk setiap kelas tersebut. Hasilnya kemudian di-plot ke dalam grafik interaktif. Fitur ini secara efektif menjawab kebutuhan fungsional F-04 dengan menyajikan perbandingan yang mudah dipahami secara visual, memungkinkan guru dan kepala sekolah untuk dengan cepat mengidentifikasi kelas yang berprestasi tinggi atau yang memerlukan perhatian lebih.



Gambar 4.18 Grafik Rata-Rata Nilai per Kelas

#### 4.3.4 View Analisis Tambahan

Tab “Analisis Tambahan” merupakan bagian lanjutan dari dashboard yang menyajikan dua jenis visualisasi yang lebih mendalam: grafik tren nilai akademik dari tahun ke tahun dan diagram panas (*heatmap*) untuk menilai distribusi nilai tiap mata pelajaran di setiap kelas. Kedua visualisasi ini memberikan konteks temporal dan kurikulum yang lebih luas terhadap performa siswa, dan ditampilkan dalam tampilan seperti yang terlihat pada Gambar 4.19. Tab ini memberikan perspektif longitudinal dan multidimensi, sangat berguna bagi kepala sekolah dan guru dalam menyusun kebijakan pembelajaran yang strategis.

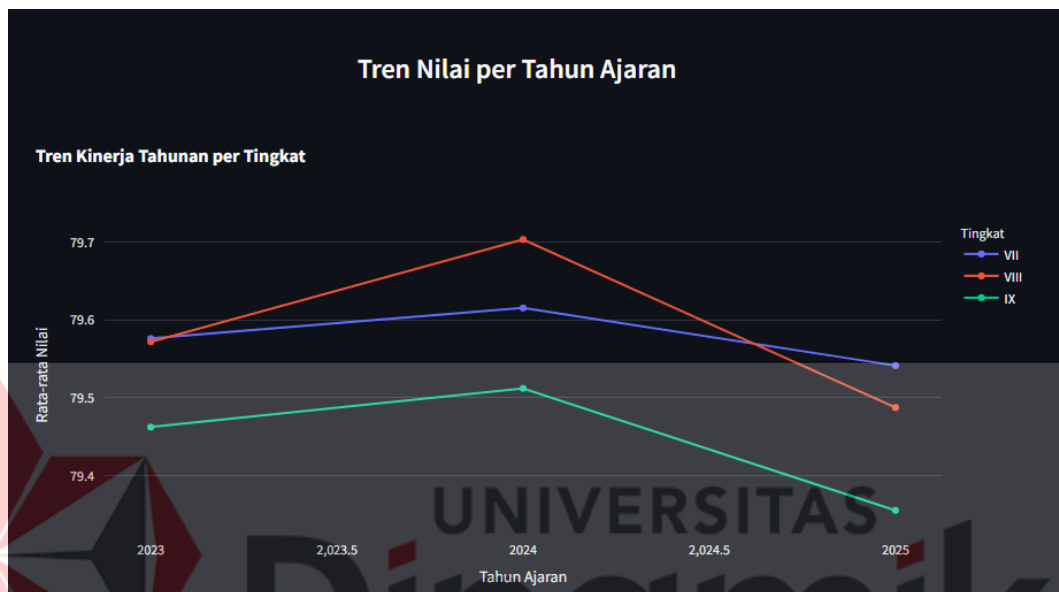


Gambar 4.19 View Analisis Tambahan

##### A. Visualisasi Tren Nilai

Aplikasi berhasil mengimplementasikan sebuah diagram garis yang berfungsi untuk menampilkan tren rata-rata nilai di setiap tingkatan dari tahun ke tahun, seperti yang ditunjukkan pada Gambar 4.16. Pembahasan implementasinya, data dikelompokkan terlebih dahulu berdasarkan tahun ajaran dan tingkat, kemudian rata-rata nilai dihitung dan di-plot secara berurutan untuk menunjukkan

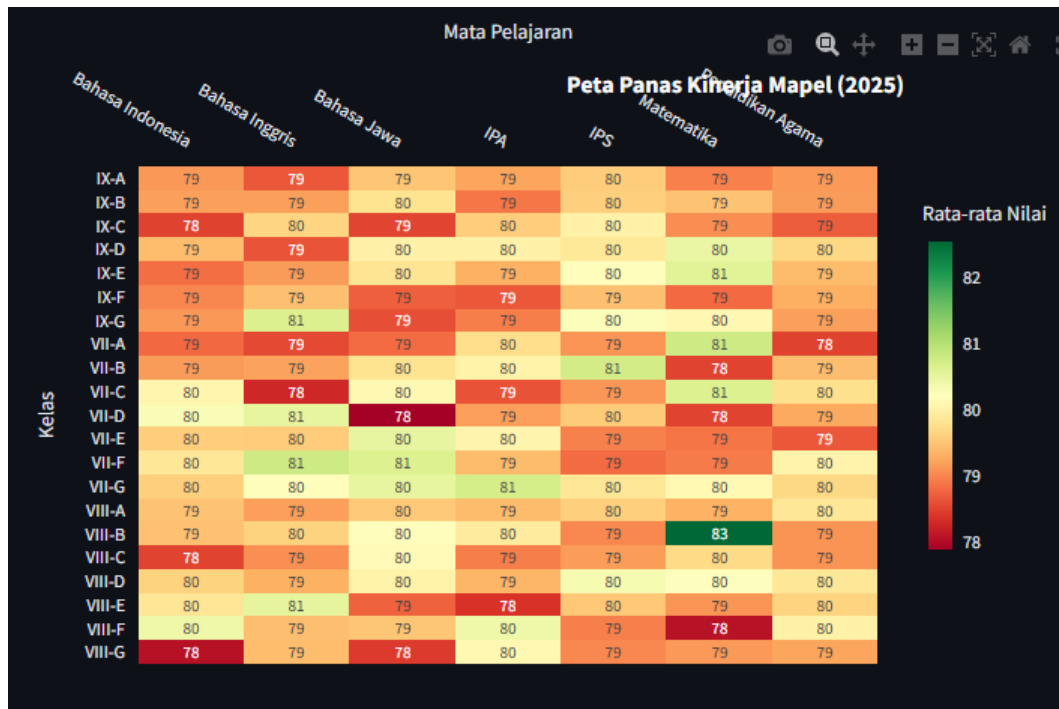
adanya tren kenaikan atau penurunan performa. Visualisasi ini menjawab kebutuhan fungsional F-05 dengan memberikan pandangan longitudinal terhadap data akademik, yang sangat berguna untuk mengevaluasi dampak dari perubahan kurikulum atau metode pengajaran dalam jangka panjang.



Gambar 4. 20 Grafik Tren Nilai Per Tahun Ajaran

#### B. Visualisasi *Heatmap* Nilai Mata Pelajaran

Hasil implementasi fungsionalitas ini adalah sebuah diagram panas (*heatmap*) yang kompleks, berfungsi untuk menampilkan rata-rata nilai per kelas untuk setiap mata pelajaran secara bersamaan (Gambar 4.18). Pembahasan teknisnya, implementasi ini memerlukan proses transformasi data (*pivot*) untuk mengatur data menjadi format matriks, di mana warna pada setiap sel merepresentasikan tingkat nilai. Fitur ini menjawab kebutuhan fungsional F-07 dan menawarkan pandangan holistik untuk mengidentifikasi pola kinerja kurikuler. Kemampuan membangun visualisasi sekompleks ini dengan efisien menunjukkan keunggulan *framework* Streamlit untuk aplikasi data (Hermawan et al., 2024).



Gambar 4. 21 *Heatmap* Nilai Mapel per Kelas

#### 4.3.5 View Tabel Siswa

Tampilan “Tabel Siswa” merupakan fitur penutup dari keseluruhan rangkaian visualisasi dalam dashboard, yang menyajikan data siswa dalam bentuk tabel interaktif. Tampilan ini dirancang sebagai lapisan analisis paling detail, yang memungkinkan pengguna untuk melihat dan menelusuri informasi nilai siswa secara individual dan menyeluruh. Halaman ini dapat diakses melalui tab ketiga pada tampilan utama aplikasi dan divisualisasikan seperti pada Gambar 4.22.

	id_siswa	nama_siswa	kelas	materi	nilai
35280	1261	Siswa VII-A-2025-1	VII-A	Matematika	75
35281	1261	Siswa VII-A-2025-1	VII-A	Matematika	84
35282	1261	Siswa VII-A-2025-1	VII-A	Matematika	80
35283	1261	Siswa VII-A-2025-1	VII-A	Matematika	79
35284	1261	Siswa VII-A-2025-1	VII-A	IPA	81
35285	1261	Siswa VII-A-2025-1	VII-A	IPA	79
35286	1261	Siswa VII-A-2025-1	VII-A	IPA	80
35287	1261	Siswa VII-A-2025-1	VII-A	IPA	80
35288	1261	Siswa VII-A-2025-1	VII-A	IPS	89

Gambar 4.22 View Tabel Siswa

#### A. Tabel nilai siswa

Sebagai lapisan analisis paling detail, sistem berhasil menyediakan fungsionalitas untuk menampilkan data siswa dan nilainya dalam bentuk tabel yang terstruktur. Pembahasan untuk fitur ini, meskipun sederhana, sangat krusial. Implementasinya menggunakan fungsi `st.dataframe` dari Streamlit untuk merender isi dari Pandas DataFrame yang telah difilter ke antarmuka pengguna. Fungsionalitas yang memenuhi kebutuhan F-08 ini berfungsi sebagai penjamin transparansi dan akurasi. Ini memungkinkan pengguna, seperti guru mata pelajaran, untuk melakukan verifikasi data secara mendalam dengan menelusuri dari visualisasi tingkat tinggi hingga ke data mentah per individu, sehingga meningkatkan kepercayaan terhadap informasi yang disajikan oleh *dashboard*.

	id_siswa	nama_siswa	kelas	materi	↑ nilai
35280	1261	Siswa VII-A-2025-1	VII-A	Matematika	75
35290	1261	Siswa VII-A-2025-1	VII-A	IPS	75
35297	1261	Siswa VII-A-2025-1	VII-A	Bahasa Inggris	75
35301	1261	Siswa VII-A-2025-1	VII-A	Bahasa Jawa	75
35305	1261	Siswa VII-A-2025-1	VII-A	Pendidikan Agama	75
35314	1262	Siswa VII-A-2025-2	VII-A	IPA	75
35336	1263	Siswa VII-A-2025-3	VII-A	Matematika	75
35391	1264	Siswa VII-A-2025-4	VII-A	Pendidikan Agama	75
35418	1265	Siswa VII-A-2025-5	VII-A	Pendidikan Agama	75
35474	1267	Siswa VII-A-2025-7	VII-A	Pendidikan Agama	75

Gambar 4. 23 Tabel nilai siswa

#### 4.4 Testing

Proses pengujian fungsionalitas aplikasi akan menggunakan metode *Blackbox*, yang menilai fungsi aplikasi tanpa memeriksa struktur *internalnya*. Metode ini memungkinkan penguji untuk memastikan setiap fitur berfungsi sesuai spesifikasi.

1. Tabel pengujian akan mencakup kolom-kolom berikut:
2. Test Scenario ID: Identifikasi unik untuk setiap skenario pengujian.
3. Test Scenario Description: Deskripsi rinci skenario pengujian.
4. Test Case ID: Identifikasi unik untuk setiap kasus pengujian.
5. Test Case Description: Deskripsi rinci setiap scenario.
6. Step: Langkah-langkah pengujian terperinci.
7. Expected Result: Hasil yang diharapkan berdasarkan spesifikasi.
8. Actual Result: Hasil aktual dari pengujian.
9. Status: Status akhir pengujian, seperti “Lulus” atau “Gagal”

1. Pengujian dianggap berhasil jika hasil aktual sesuai dengan yang diharapkan.  
Metode *Blackbox* ini fokus pada pengalaan pengguna akhir dan memastikan aplikasi memenuhi kebutuhan tanpa mempengaruhi implementasi internal.



UNIVERSITAS  
**Dinamika**



Tabel 4. 4 Tabel Skenario Testing

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Step	Expected Result	Actual Result	Status
TS-01	Pemuatan Awal dan Tampilan Default	TC-01.1	Verifikasi pemuatan aplikasi yang sukses.	1. Buka aplikasi pada web browser.	<ul style="list-style-type: none"> <li>- Halaman dimuat tanpa eror.</li> <li>-Judul utama "Dasbor Analisis Nilai Siswa SMP" ditampilkan.</li> <li>-Filter "Pilih Tampilan Data" secara default menunjukkan 'Semua Kelas'.</li> <li>-Nilai KKM secara default adalah 75.</li> </ul>	<ul style="list-style-type: none"> <li>-Halaman dimuat tanpa eror.</li> <li>-Judul utama "Dasbor Analisis Nilai Siswa SMP" ditampilkan.</li> <li>-Filter "Pilih Tampilan Data" secara default menunjukkan 'Semua Kelas'.</li> <li>-Nilai KKM secara default adalah 75.</li> </ul>	Pass

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Step	Expected Result	Actual Result	Status
		TC-01.2	Verifikasi penanganan nama kelas yang tidak valid.	1. (Dengan data uji yang mengandung nama kelas tidak standar, misal "Kelas 7"). 2. Muat ulang aplikasi.	Sebuah expander dengan judul " ⚠ Peringatan: Beberapa Nama Kelas Tidak Dapat Diproses" muncul dan menampilkan daftar nama kelas yang bermasalah.	Sebuah expander dengan judul " ⚠ Peringatan: Beberapa Nama Kelas Tidak Dapat Diproses" muncul dan menampilkan daftar nama kelas yang bermasalah.	Pass
TS-02	Fungsionalitas Filter Utama	TC-02.1	Mengubah filter ke 'Per Tingkat'.	1. Klik dropdown "Pilih Tampilan Data". 2. Pilih opsi 'Per Tingkat'.	- Judul ringkasan berubah menjadi "Ringkasan Nilai: Per Tingkat". - Grafik bar di Tab 1 menampilkan perbandingan rata-rata nilai	- Judul ringkasan berubah menjadi "Ringkasan Nilai: Per Tingkat". - Grafik bar di Tab 1 menampilkan perbandingan	Pass

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Step	Expected Result	Actual Result	Status
					untuk tingkat VII, VIII, dan IX  - <i>Heatmap</i> di Tab 2 ditampilkan.	rata-rata nilai untuk tingkat VII, VIII, dan IX  - <i>Heatmap</i> di Tab 2 ditampilkan.	
		TC-02.2	Mengubah filter ke kelas spesifik.	1.Klik dropdown "Pilih Tampilan Data".  2. Pilih salah satu nama kelas (misal: 'VII A').	- Judul ringkasan berubah menjadi "Ringkasan Nilai: VII A".  - Semua KPI (Rata-rata Nilai, Jumlah Siswa, dll.) diperbarui sesuai data kelas 'VII A'.  - <i>Heatmap</i> di Tab 2 hilang dan digantikan pesan info "Pilih	- Judul ringkasan berubah menjadi "Ringkasan Nilai: VII A".  - Semua KPI (Rata-rata Nilai, Jumlah Siswa, dll.) diperbarui sesuai data kelas 'VII A'.  - <i>Heatmap</i> di Tab 2 hilang dan digantikan pesan	Pass

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Step	Expected Result	Actual Result	Status
					'Semua Kelas' atau 'Per Tingkat' untuk menampilkan Peta Panas."	info "Pilih 'Semua Kelas' atau 'Per Tingkat' untuk menampilkan Peta Panas."	
TS-03	Interaksi dengan Input KKM	TC-03.1	Mengubah nilai KKM	1. Arahkan ke input "Atur Nilai KKM". 2. Ubah nilai dari 75 menjadi 80. 3. Tekan Enter.	- KPI "Tingkat Kelulusan" di Tab 1 nilainya berubah (diperbarui). - Di Tab 2, teks di atas diagram berubah menjadi "...menggunakan nilai KKM 80...". - Diagram Donut "Distribusi Status Kelulusan" diperbarui untuk merefleksikan persentase	- KPI "Tingkat Kelulusan" di Tab 1 nilainya berubah (diperbarui). - Di Tab 2, teks di atas diagram berubah menjadi "...menggunakan nilai KKM 80...". - Diagram Donut "Distribusi Status Kelulusan"	Pass

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Step	Expected Result	Actual Result	Status
					Tuntas/Perlu Perbaikan berdasarkan KKM 80.	diperbarui untuk merefleksikan persentase Tuntas/Perlu Perbaikan berdasarkan KKM 80.	
TS-04	Verifikasi Konten Tab 1 (Ringkasan & Perbandingan)	TC-04.1	Verifikasi akurasi KPI	1. Pilih filter untuk satu kelas spesifik. 2. Bandingkan nilai KPI "Rata-rata Nilai" dan "Jumlah Siswa" dengan perhitungan manual dari data sumber.	Nilai yang ditampilkan pada kedua KPI tersebut harus sama persis dengan hasil perhitungan manual.	Nilai yang ditampilkan pada kedua KPI tersebut harus sama persis dengan hasil perhitungan manual.	Pass
		TC-04.2	Verifikasi grafik tren	1. Pilih filter untuk satu kelas spesifik.	Grafik tersebut harus menampilkan satu garis yang	Grafik tersebut harus menampilkan satu garis yang	Pass

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Step	Expected Result	Actual Result	Status
			tahunan per kelas	2. Amati grafik "Tren Nilai per Tahun Ajaran".	menunjukkan tren rata-rata nilai untuk kelas yang dipilih di setiap tahun ajaran yang tersedia.	menunjukkan tren rata-rata nilai untuk kelas yang dipilih di setiap tahun ajaran yang tersedia.	
TS-05	Verifikasi Konten Tab 2 (Analisis Tambahan)	TC-05.1	Verifikasi akurasi Donut Chart	1. Pilih filter dan atur KKM ke nilai tertentu. 2. Hitung manual persentase siswa yang nilainya $\geq$ KKM (Tuntas) vs $<$ KKM (Perlu Perbaikan). 3. Bandingkan dengan porsi diagram Donut.	Proporsi irisan pada diagram Donut harus sesuai dengan hasil perhitungan manual persentase status kelulusan.	Proporsi irisan pada diagram Donut harus sesuai dengan hasil perhitungan manual persentase status kelulusan.	Pass

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Step	Expected Result	Actual Result	Status
		TC-05.2	Verifikasi visibilitas <i>Heatmap</i>	1. Pilih filter 'Semua Kelas' dan buka Tab 2.  2. Pilih filter kelas spesifik (misal 'VIII B') dan buka Tab 2.	- Saat filter 'Semua Kelas', <i>Heatmap</i> harus ditampilkan.  - Saat filter 'VIII B', <i>Heatmap</i> harus disembunyikan dan pesan info ditampilkan.	- Saat filter 'Semua Kelas', <i>Heatmap</i> harus ditampilkan.  - Saat filter 'VIII B', <i>Heatmap</i> harus disembunyikan dan pesan info ditampilkan.	Pass
TS-06	Verifikasi Konten Tab 3 (Tabel nilai siswa)	TC-06.1	Visualisasi Tabel Nilai	1. Pilih filter 'Semua Kelas' dan buka Tab 3.  2. Pilih filter kelas spesifik (misal 'VIII B') dan buka Tab 3.  3. Pilih lulus atau tidak	-Saat filter 'Semua Kelas', tabel nilai siswa harus sesuai.  -Saat filter 'Kelas', tabel nilai siswa harus sesuai.  --Saat filter 'Lulus', tabel	-Saat filter 'Semua Kelas', tabel nilai siswa harus sesuai.  -Saat filter 'Kelas', tabel nilai siswa harus sesuai.  --Saat filter 'Lulus', tabel	Pass

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Step	Expected Result	Actual Result	Status
					siswa harus sesuai.	nilai siswa harus sesuai.	



UNIVERSITAS  
**Dinamika**



#### 4.5 Maintenance

*maintenance* (pemeliharaan) adalah fase terakhir dalam siklus hidup pengembangan perangkat lunak (SDLC), yang dimulai setelah aplikasi selesai diuji dan digunakan oleh pengguna. Tujuan dari tahap ini adalah untuk memastikan aplikasi *dashboard* nilai siswa dapat terus berjalan dengan baik, andal, dan relevan dalam jangka panjang. Rencana pemeliharaan untuk aplikasi ini mencakup beberapa aktivitas utama:

2. Perbaikan Error (*Bug Fixing*): Melakukan pemantauan berkelanjutan untuk mengidentifikasi dan memperbaiki setiap *bug* atau kesalahan fungsi yang mungkin tidak ditemukan selama tahap pengujian, namun muncul saat digunakan oleh guru dan kepala sekolah dalam kondisi nyata.
3. Pembaruan Sistem: Melakukan pembaruan secara berkala untuk meningkatkan fungsionalitas yang ada atau menambahkan fitur baru berdasarkan masukan dan kebutuhan dari pihak sekolah di masa mendatang.
4. Pemantauan Nilai: Secara rutin memantau Nilai aplikasi, terutama kecepatan dalam memuat data dan menampilkan visualisasi, untuk mengidentifikasi potensi masalah dan melakukan optimasi jika diperlukan.
5. Dukungan Pengguna: Memberikan dukungan teknis kepada pengguna (guru dan kepala sekolah) jika mereka menghadapi kesulitan atau memiliki pertanyaan saat menggunakan aplikasi *dashboard*.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil kerja praktik yang telah dilaksanakan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Telah berhasil dirancang dan dibangun sebuah aplikasi "Dashboard Nilai Siswa Berbasis Web Menggunakan Streamlit" yang menjawab kebutuhan UPT SMP Negeri 16 Gresik dalam mengatasi kendala sistem visualisasi nilai manual. Bukti dari keberhasilan ini dijelaskan secara rinci pada Bab IV, yang mendokumentasikan seluruh proses pengerjaan mulai dari tahap Analisis, Desain, hingga Implementasi.
2. Aplikasi yang dikembangkan telah memenuhi seluruh kebutuhan fungsional yang telah diidentifikasi. Buktinya dijelaskan pada sub-bab 4.3 Implementasi, di mana setiap kebutuhan fungsional (dari F-01 hingga F-07) berhasil dibangun dan ditunjukkan hasil jadinya melalui tangkapan layar aplikasi. Fungsionalitas tersebut mencakup penampilan dashboard utama, fitur filter data, penyajian metrik kinerja siswa, serta berbagai visualisasi data seperti diagram batang, diagram garis, diagram lingkaran, dan *heatmap*.
3. Pembangunan sistem ini telah melalui tahapan SDLC dengan model Waterfall secara sistematis. Penerapan model ini terlihat dari alur pengerjaan di Bab IV yang berurutan, mulai dari analisis kebutuhan, perancangan alur sistem dan data, implementasi kode, hingga pengujian.

4. Berdasarkan pengujian fungsionalitas menggunakan metode Blackbox testing, seluruh fitur yang diimplementasikan telah berjalan sesuai dengan hasil yang diharapkan dan dinyatakan Lulus (*Pass*). Bukti konkret dari hasil ini adalah Tabel 4.4 Tabel Skenario Testing, yang menunjukkan bahwa semua skenario uji coba yang dijalankan berhasil tanpa ada kesalahan. Dengan demikian, aplikasi ini siap membantu pihak sekolah dalam memantau perkembangan akademik siswa dan mendukung pengambilan keputusan yang lebih objektif.

## 5.2 Saran

Meskipun aplikasi telah memenuhi tujuan awal, terdapat beberapa saran untuk pengembangan lebih lanjut agar aplikasi menjadi lebih komprehensif dan berdaya guna di masa mendatang:

1. Pengembangan Modul Input Data: Saat ini aplikasi berfokus pada visualisasi data yang sudah ada di database. Disarankan untuk menambahkan modul yang memungkinkan guru atau admin untuk melakukan input, edit, dan hapus (CRUD) data nilai langsung melalui antarmuka web, sehingga dapat sepenuhnya menggantikan proses input manual di spreadsheet.
2. Implementasi Sistem Autentikasi dan Hak Akses: Untuk meningkatkan keamanan data, disarankan untuk membangun sistem login bagi pengguna dengan level akses yang berbeda (misalnya: Kepala Sekolah, Wali Kelas, Guru Mata Pelajaran). Setiap peran akan memiliki hak akses yang disesuaikan dengan kebutuhannya.
3. Fitur Ekspor Laporan: Menambahkan fungsionalitas untuk mengekspor data atau visualisasi ke dalam format file seperti PDF atau Excel akan sangat

bermanfaat bagi guru dan wali kelas untuk membuat laporan fisik yang akan diserahkan kepada orang tua siswa atau pihak manajemen sekolah.

4. Optimasi Tampilan Perangkat Mobile: Mengingat batasan masalah yang menyebutkan dashboard dirancang untuk komputer, pengembangan selanjutnya dapat berfokus pada desain antarmuka yang lebih responsif agar dapat diakses dengan nyaman melalui berbagai perangkat, termasuk tablet dan *smartphone*.



UNIVERSITAS  
**Dinamika**

## DAFTAR PUSTAKA

- A. A. Wahid. (2020). “Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi,” . *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK*, 1(October).
- Dewa, S. B. A., & Azizah, N. L. (2024). Perancangan Sistem Informasi Sekolah Berbasis Web Menggunakan Metode SDLC. *Indonesian Journal of Applied Technology*, 1(2), 15. <https://doi.org/10.47134/ijat.v1i2.3053>
- Fahrezy, H. M., Silaen, T. P., & Henderi, H. (2020). Model Information Dashboard Nilai Siswa: Tools Monitoring Hasil Belajar. *Jurnal Ilmiah Matrik*, 22(1), 36–42. <https://doi.org/10.33557/jurnalmatrik.v22i1.837>
- Hermawan, E., Darmawan Panjaitan, S., & Faja Ripanti, E. (2024). Sistem Prediksi Banjir Rob Kota Pontianak Berbasis Machine Learning Menggunakan Framework Streamlit. *Jurnal Edukasi Dan Penelitian Informatika*, 10(3), 351–361.
- Pratama, S. D., Lasimin, L., & Dadaprawira, M. N. (2023). Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence Dan Boundary Value. *J-SISKO TECH (Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD)*, 6(2), 560. <https://doi.org/10.53513/jsk.v6i2.8166>
- Sholeh Kurniandini, Zaidatul Arifah, & Ahmad Zakariya. (2022). Teknologi Informasi dan Komunikasi (TIK) dalam Peningkatan Mutu Administrasi Pendidikan di Temanggung. *Al-Fahim : Jurnal Manajemen Pendidikan Islam*, 4(1), 73–85. <https://doi.org/10.54396/alfahim.v4i1.131>