LAPORAN PROYEK PEMROGRAMAN DASAR

SiRapor: APLIKASI PENYAJIAN DAN PENGOLAHAN DATA RAPOR BAGI SISWA SEKOLAH MENENGAH ATAS

Disusun untuk Memenuhi Tugas Mata Kuliah Pemrograman Dasar

Disusun Oleh:

Aufaa Luthfiyyah Zaahidah (25031554098)

Hafizh Akmalian Putra Pambudi (25031554104)

Miftaqul Jannah (25031554265)

2025G

Dosen Pengampu:

Hasanuddin Al-Habib, S.Si., M.Si.

Dr. Heri Purnawan, S.Si., M.Si.

PROGRAM STUDI S1 SAINS DATA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SURABAYA 2025

DAFTAR ISI

DAFTAR ISI ii

BAB I PENDAHULUAN

1.1 Latar Belakang 1

1.2 Rumusan Masalah 1

1.3 Tujuan 1

BAB II ANALISIS DAN PERENCANAAN

2.1 Analisis Kebutuhan Aplikasi 2

2.1.1 Analisis Masalah 2

2.1.2 Identifikasi Kebutuhan Aplikasi 3

2.1.2.1 Kebutuhan Fungsional 3

2.1.2.2 Kebutuhan Non-Fungsional 4

2.1.3 Analisis Kebutuhan Perangkat Lunak 4

2.1.4 Analisis Kebutuhan Perangkat Keras 5

2.2 Diagram Alur 6

2.3 Sketsa Desain Antarmuka 7

BAB III IMPLEMENTASI

3.1 Penjelasan Kode Program 11

3.1.1. Penjelasan Fitur-Fitur 12

LAMPIRAN 18

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah memberikan kemajuan dalam berbagai bidang, termasuk bidang pendidikan. Salah satu proses penting di bidang pendidikan adalah proses pengolahan dan penyajian nilai rapor siswa. Pada umumnya, perhitungan nilai dilakukan secara manual melalui lembar kerja atau aplikasi sederhana. Proses ini tidak jarang menimbulkan kendala, seperti kesalahan perhitungan *(human error)*, waktu pengolahan yang lama, dan kesulitan dalam melihat hasil akhir atau peringkat nilai siswa secara cepat.

Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah aplikasi yang dapat membantu mempermudah proses pengolahan nilai rapor secara otomatis. Berdasarkan permasalahan tersebut, penulis mengembangkan sebuah proyek aplikasi bernama SiRapor, yaitu aplikasi penyajian dan pengolahan data rapor untuk siswa Sekolah Menengah Atas. Aplikasi ini dirancang untuk menghitung nilai rata-rata secara otomatis, menampilkan hasil peringkat siswa, serta meminimalisasi kesalahan dalam proses penghitungan nilai.

Melalui penerapan aplikasi ini, diharapkan proses pengelolaan nilai dapat menjadi lebih cepat, akurat, dan efisien. Selain itu, aplikasi SiRapor juga dapat membantu guru, wali kelas, maupun pihak sekolah dalam melakukan evaluasi hasil pembelajaran siswa secara lebih sistematis dan terstruktur.

1.2 Rumusan Masalah

1. Apa saja permasalahan yang terjadi dalam proses pengolahan nilai dan penyusunan rapor siswa?
2. Bagaimana aplikasi SiRapor dapat menjawab permasalahan yang ada?
3. Bagaimana proses pengembangan aplikasi SiRapor?

1.3 Tujuan

1. Untuk mengetahui permasalahan yang terjadi dalam proses pengolahan nilai dan penyusunan rapor siswa
2. Untuk mengetahui aplikasi SiRapor dapat menjawab permasalahan yang ada
3. Untuk mengetahui proses pengembangan aplikasi SiRapor?

BAB II
ANALISIS DAN PERENCANAAN

2.1 Analisis Kebutuhan Aplikasi

Analisis kebutuhan aplikasi adalah proses identifikasi terhadap berbagai aspek yang berkaitan dengan pengembangan aplikasi. Tahap ini bertujuan untuk mengetahui kebutuhan yang harus dipenuhi aplikasi, mencakup aspek fungsional dan non-fungsional lainnya. Oleh karena itu, analisis kebutuhan aplikasi adalah langkah penting dalam pengembangan aplikasi SiRapor untuk memastikan bahwa aplikasi memenuhi kebutuhan pengguna dan tujuan pendidikan.

2.1.1 Analisis Masalah

Penghitungan secara manual sering kali tidak efisien dan menimbulkan berbagai permasalahan, permasalahan tersebut diantaranya adalah:

1. Terjadinya Kesalahan Perhitungan (Human Error)

   Penghitungan nilai rata-rata secara manual rentan terhadap kesalahan, seperti salah menjumlahkan nilai, salah membagi jumlah nilai dengan jumlah komponen penilaian, atau keliru dalam memasukkan angka. Kesalahan kecil ini dapat menyebabkan hasil nilai akhir tidak akurat dan merugikan siswa.

2. Membutuhkan Waktu yang Relatif Lama

   Proses penghitungan manual harus dilakukan satu per satu untuk setiap siswa dan setiap mata pelajaran. Semakin banyak jumlah siswa dan komponen penilaian, semakin lama waktu yang dibutuhkan, sehingga proses penyusunan rapor menjadi tidak efisien.

3. Sulit Menentukan Peringkat Siswa

   Penentuan peringkat siswa memerlukan perbandingan nilai rata-rata secara menyeluruh. Jika dilakukan secara manual, proses ini menjadi rumit dan berisiko salah urut, terutama ketika nilai siswa memiliki selisih yang kecil.

4. Kurang Efisien dalam Proses Evaluasi

   Penghitungan manual menyulitkan guru atau wali kelas untuk melakukan evaluasi hasil belajar secara cepat, seperti melihat distribusi nilai atau membandingkan hasil antar siswa, karena data tidak tersaji secara otomatis dan terstruktur.

Dengan memahami permasalahan-permasalahan tersebut aplikasi SiRapor dirancang untuk memberikan solusi yang efektif dan efisien. SiRapor menerapkan sistem pengolahan nilai berbasis komputer yang memungkinkan proses perhitungan dilakukan secara otomatis dan terstruktur sehingga mampu meminimalkan kesalahan penghitungan, mempercepat pengolahan nilai serta nilai rata rata yang konsisten dan akurat. Selain itu, aplikasi ini juga mempermudah penentuan peringkat siswa karena data nilai diolah dan diurutkan secara otomatis.

2.1.2 Identifikasi Kebutuhan Aplikasi

Untuk menyelesaikan permasalahan yang ada, kebutuhan aplikasi diidentifikasi berdasarkan aspek fungsional dan non-fungsional yang relevan dengan tujuan pengembangan aplikasi SiRapor.

2.1.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berkaitan langsung dengan fungsi utama yang harus dijalankan oleh aplikasi SiRapor.

1. Pengolahan Data Siswa

   Aplikasi SiRapor dapat digunakan untuk mengelola data siswa, meliputi penambahan, pengubahan, dan penghapusan data siswa yang diperlukan dalam proses penilaian.

2. Pengolahan Data Mata Pelajaran

   Aplikasi menyediakan fitur untuk mengelola data mata pelajaran yang digunakan dalam proses pengolahan nilai dan penyusunan rapor siswa.

3. Input Nilai Siswa

   Aplikasi Memungkinkan pengguna untuk memasukkan nilai siswa pada setiap mata pelajaran dan komponen penilaian yang telah ditentukan.

4. Perhitungan Nilai Otomatis

   Aplikasi dapat melakukan perhitungan nilai rata-rata siswa secara otomatis berdasarkan data nilai yang dimasukkan, sehingga dapat meminimalkan kesalahan perhitungan.

5. Penyusunan dan Tampilan Rapor

   Aplikasi dapat menampilkan hasil pengolahan nilai dalam bentuk rapor siswa yang tersusun secara rapi dan mudah dipahami oleh pengguna.

6. Penentuan Peringkat Siswa

   Aplikasi dapat mengurutkan nilai siswa secara otomatis untuk menentukan peringkat berdasarkan nilai akhir yang diperoleh.

7. Penyimpanan Data Nilai

   Aplikasi dapat menyimpan data nilai siswa sehingga data dapat diakses kembali tanpa perlu melakukan input ulang.

2.1.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang berkaitan dengan kualitas, kinerja, dan kenyamanan penggunaan aplikasi SiRapor.

1. Kinerja

   Aplikasi SiRapor mampu memproses data nilai dengan cepat dan responsif sehingga tidak menghambat aktivitas pengguna dalam pengolahan rapor.

2. Keandalan

   Aplikasi dapat berjalan dengan stabil dan menghasilkan perhitungan nilai yang konsisten serta akurat.

3. Kemudahan Penggunaan

   Antarmuka aplikasi dirancang sederhana dan mudah dipahami agar dapat digunakan oleh pengguna tanpa memerlukan pelatihan khusus..

4. Portabilitas

   Aplikasi dapat dijalankan pada perangkat komputer atau laptop dengan sistem operasi yang umum digunakan.

5. Keamanan Data

   Aplikasi mampu menjaga data nilai siswa agar tidak mudah hilang atau berubah tanpa sepengetahuan pengguna.

2.1.3 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam pembuatan aplikasi SiRapor antara lain:

1. Python

Python digunakan sebagai bahasa pemrograman utama dalam pengembangan aplikasi SiRapor. Python berfungsi menjalankan seluruh logika program, membuat antarmuka (tampilan aplikasi), mengelola sistem

login, mengelola data nilai siswa, membuat laporan PDF, mengatur file dan direktori.

2. **Visual Studio Code**

   Visual Studio Code digunakan sebagai aplikasi editor untuk menulis, mengedit, dan menjalankan kode program Python.

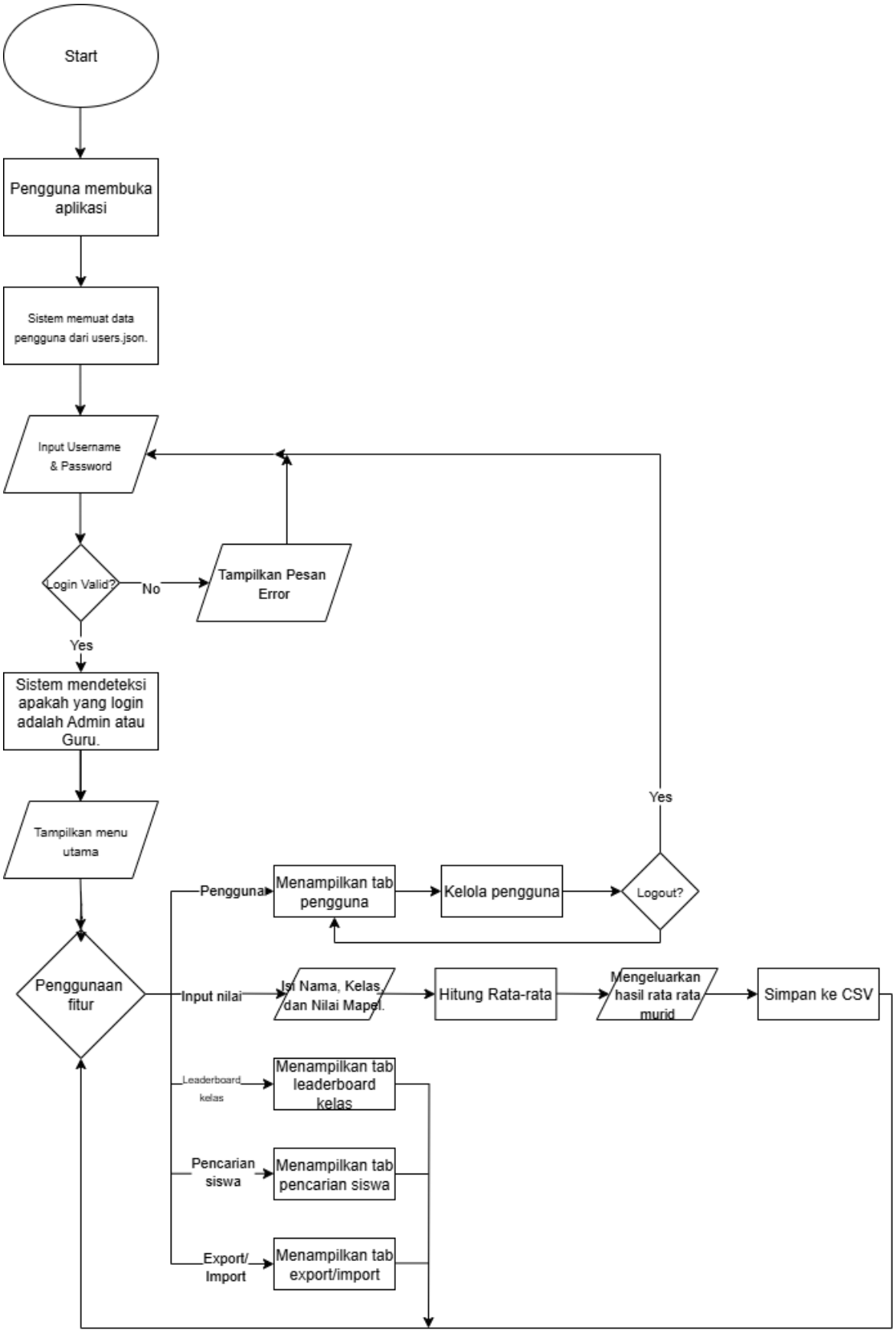
2.1.4 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam pembuatan aplikasi SiRapor ini antara lain:

1. **Laptop**

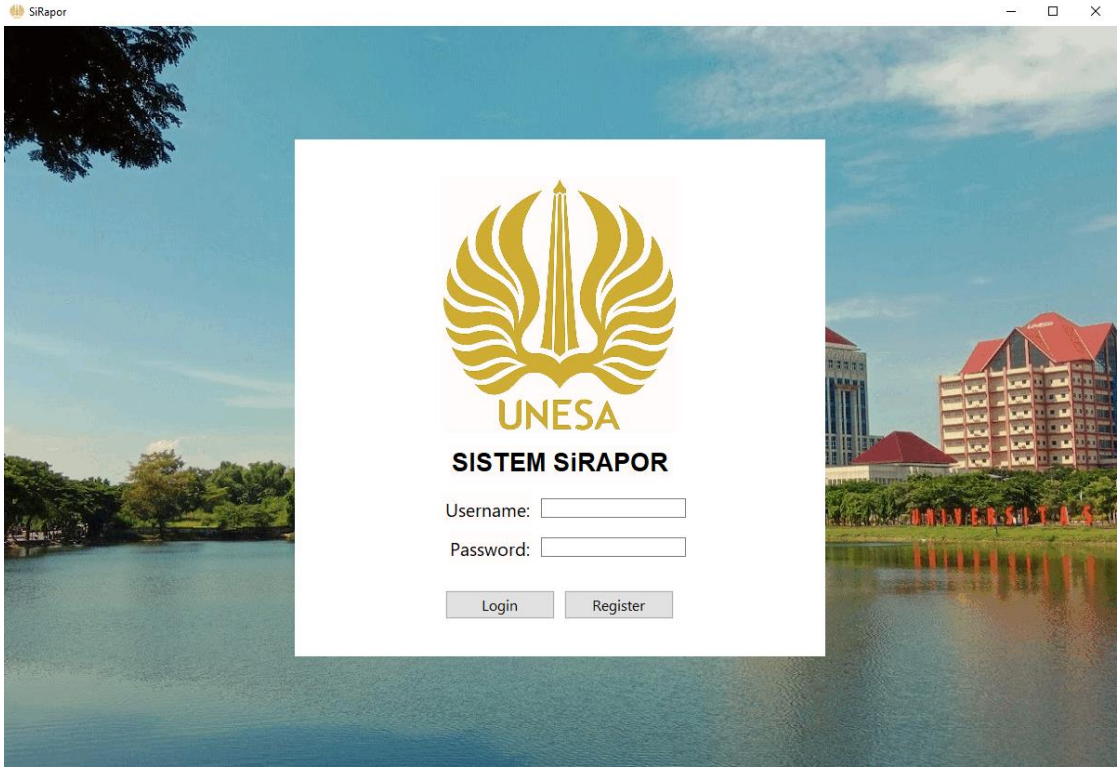
   Laptop merupakan sarana utama untuk menjalankan seluruh proses pengembangan aplikasi, mulai dari penulisan kode program hingga pengujian aplikasi.

2.2 Diagram Alur



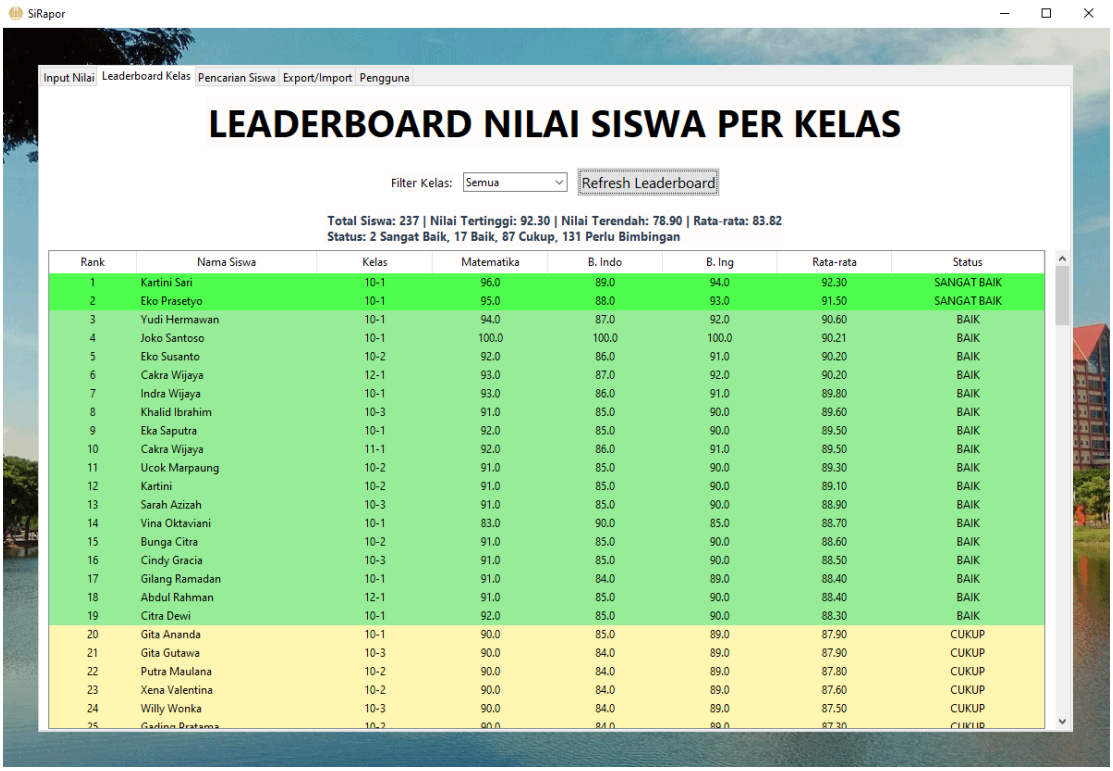
Gambar 2.2. 1 Diagram Alur Proyek SiRapor

2.3 Sketsa Desain Antarmuka

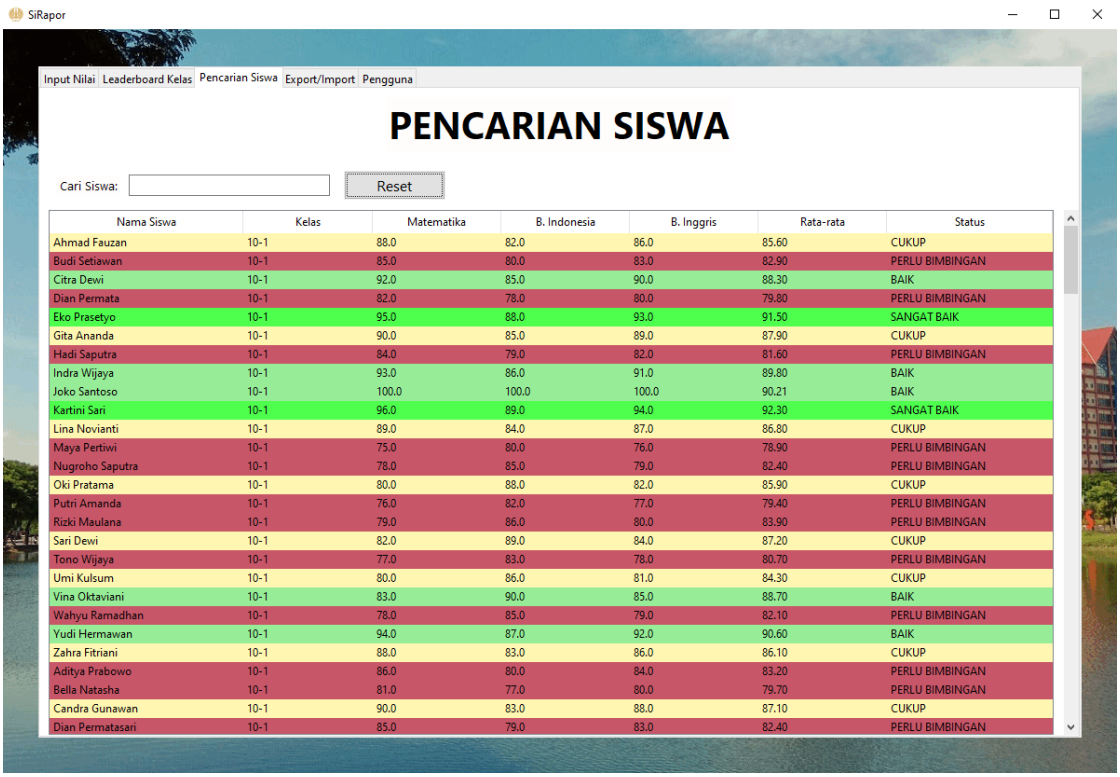


Gambar 2.2. 2 Desain Antarmuka user di bagian Halaman Login

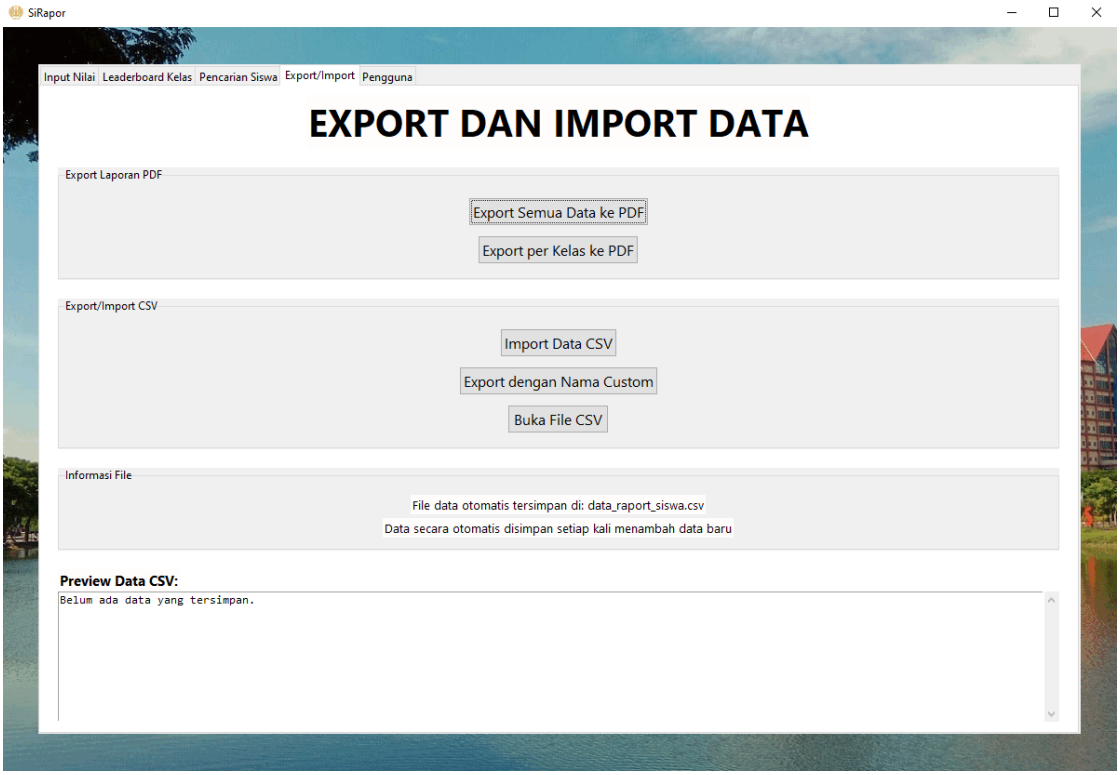
Gambar 2.2. 3 Desain Antarmuka user di bagian Input Nilai



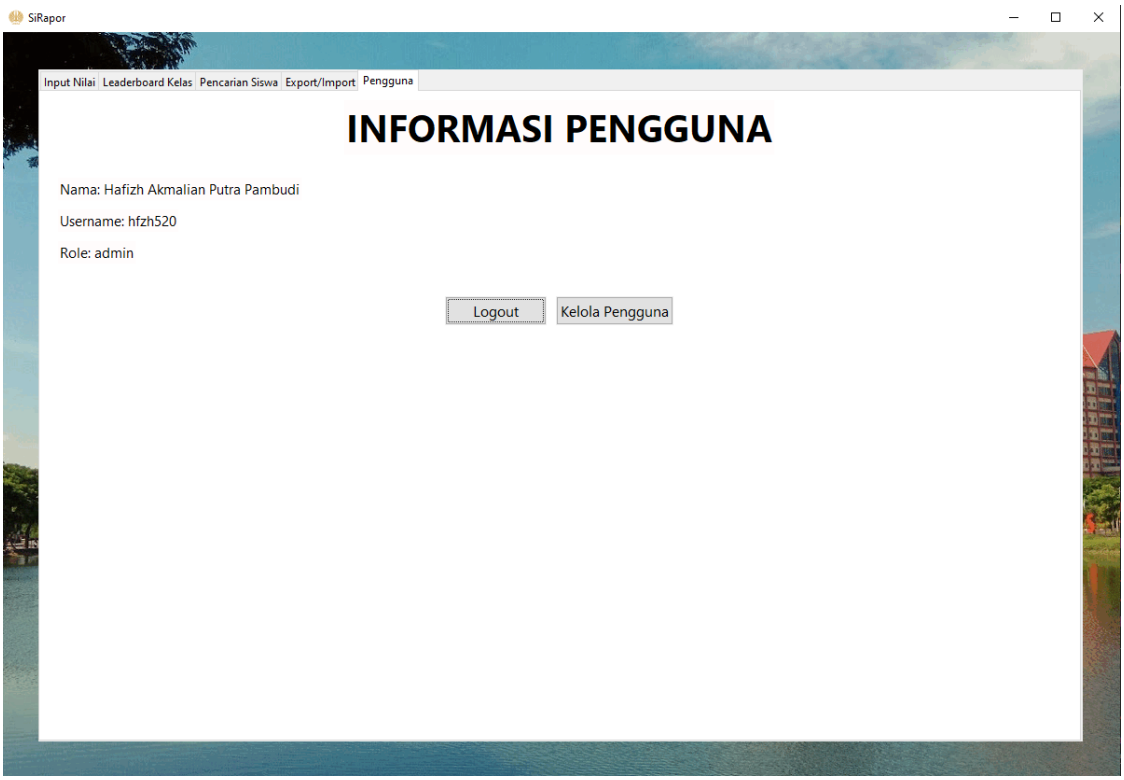
Gambar 2.2. 3 Desain Antarmuka user di bagian Leaderboard Kelas



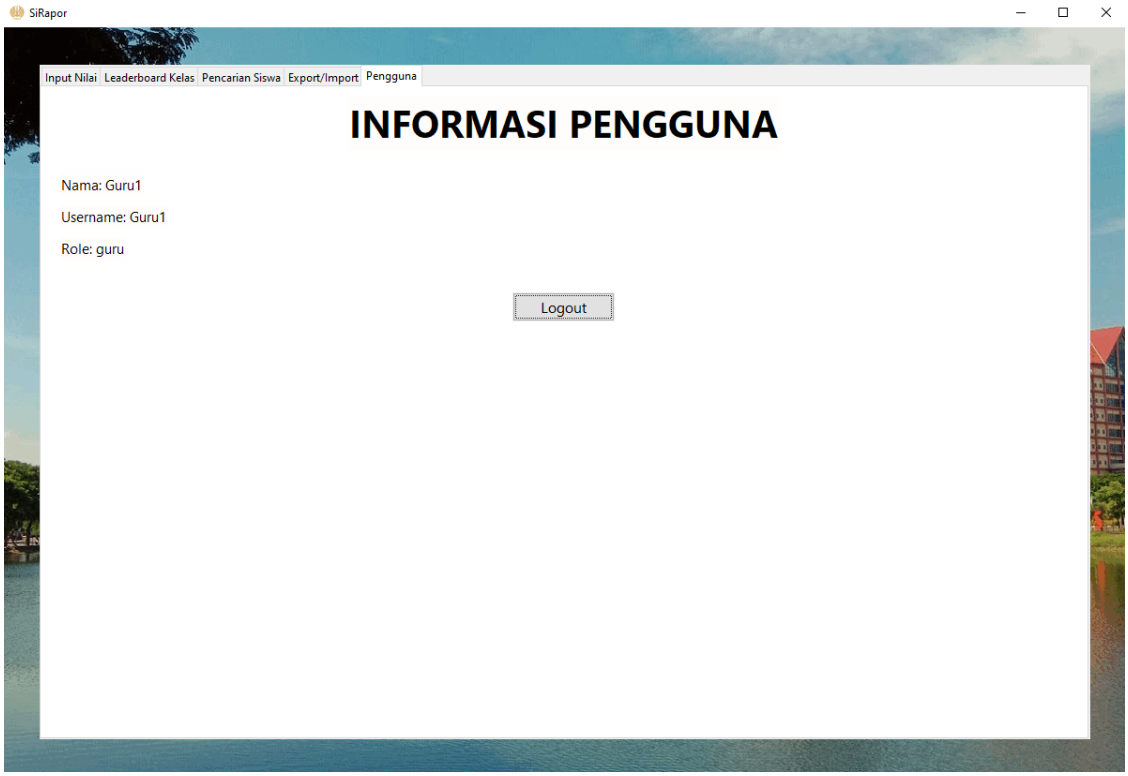
Gambar 2.2. 4 Desain Antarmuka user di bagian Pencarian Siswa



Gambar 2.2. 5 Desain Antarmuka user di bagian Export Import Data



Gambar 2.2. 6 Desain Antarmuka user di bagian Informasi Pengguna



Gambar 2.2. 7 Desain Antarmuka user di bagian Informasi Pengguna

BAB III
IMPLEMENTASI

3.1 Penjelasan Kode Program

Untuk memahami bagaimana aplikasi SiRapor bekerja, perlu dilihat bagian-bagian utama yang membangun sistemnya. Aplikasi ini disusun dalam beberapa file yang masing-masing memiliki fungsi berbeda, namun saling terhubung untuk mendukung proses pengolahan data siswa dan pembuatan rapor digital. Sebelum masuk pada penjelasan mengenai proses kerja sistem, terlebih dahulu perlu dijelaskan file-file yang terdapat dalam folder. Secara umum, folder dari pembuatan aplikasi SiRapor terdiri dari beberapa file utama, yaitu:

1. **SiRapor.py**
   File utama yang digunakan untuk menjalankan aplikasi. Berisi pemrograman, tampilan (GUI) menggunakan Tkinter, dan logika pemrosesan data.
2. **data_raport_siswa.csv**
   Berisi data nilai siswa dan tempat untuk menyimpan data nilai setelah di inputkan.
3. **users.json**
   Menyimpan data login serta password user.
4. **logo.png**
   Digunakan sebagai logo utama pada tampilan login dan register
5. **Unesa.ico**
   Digunakan sebagai ikon aplikasi yang muncul dibagian kiri atas jendela setelah login.
6. **unesaBg,gif**
   Digunakan untuk background tampilan awal login aplikasi SiRapor

Selanjutnya, aplikasi ini dibuat dengan menggunakan beberapa library python yang masing-masing mempunyai peran penting. Library-library tersebut di antaranya adalah:

```python
import tkinter as tk
from tkinter import ttk, messagebox, filedialog
import csv
import os
from datetime import datetime
import hashlib
import json
from reportlab.lib.pagesizes import letter, A4
from reportlab.platypus import SimpleDocTemplate, Table, TableStyle, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
from reportlab.lib import colors
from reportlab.lib.units import inch
from reportlab.pdfbase import pdfmetrics
from reportlab.pdfbase.ttfonts import TTFont
```

1. **Json**, digunakan untuk menyimpan data pengguna (user dan admin) ke file users.json. Fungsinya menyimpan username, password, role, dan nama user, serta membaca kembali data user tersebut saat login.
2. **Hashlib**, digunakan untuk hashing password. Fungsinya untuk menyimpan password dalam bentuk terenkripsi sehingga menjaga keamanan sistem login.
3. **Reportlab**, digunakan untuk mengekspor rapor ke PDF. Fungsinya untuk membuat file PDF laporan nilai, membuat tabel, border, layout, dan style pada PDF.
4. **Tkinter**, digunakan untuk tampilan antarmuka. Fungsinya untuk membuat tampilan GUI.
5. **OS**, library os berperan penting untuk mengelola keberadaan file data yang dibutuhkan oleh sistem
6. **Datetime**, digunakan sebagai penanda (timestamp) yang membantu dalam pelacakan dan pengelolaan data rapor

3.1.1 Penjelasan Fitur-Fitur

Di dalam aplikasi SiRapor terdapat beberapa fitur yang membantu pengguna. Fitur-fitur tersebut di antaranya adalah fitur login dan register, fitur pengolahan nilai rata-rata, status yang diberikan sesuai dengan hasil penghitungan nilai rata-rata, fitur export dalam bentuk PDF. Dengan fitur-fitur tersebut, SiRapor mampu memberikan solusi praktis bagi guru dalam pengolahan nilai siswanya.

a. Login dan Register

Fitur login pada aplikasi SiRapor merupakan sistem autentikasi pengguna yang digunakan untuk membatasi akses ke dalam aplikasi. Tujuan utama dari fitur ini adalah untuk memastikan bahwa hanya pengguna yang

memiliki akun yang valid yang dapat mengakses dan mengelola data nilai siswa.

```python
class LoginSystem:
    def __init__(self):
        self.users_file = "users.json"
        self.current_user = None
        self.load_users()

    def load_users(self):
        if os.path.exists(self.users_file):
            try:
                with open(self.users_file, 'r') as f:
                    self.users = json.load(f)
            except:
                self.users = {}
        else:
            self.users = {
                "admin": {
                    "password": self.hash_password("admin"),
                    "role": "admin",
                    "nama": "Administrator"
                }
            }
            self.save_users()
```

(method) def hash_password(password: Any) -> str

Pada Class LoginSystem terdapat fungsi __init__ untuk menentukan bahwa data akun pengguna akan disimpan dalam file bernama users.json, file ini berfungsi seperti database kecil. Selanjutnya terdapat fungsi load_users untuk memuat data pengguna. Jika file ada maka program akan membacanya, sedangkan jika tidak ada program akan membuat akun admin default.

```python
def hash_password(self, password):
    return hashlib.sha256(password.encode()).hexdigest()
```

Dalam Class LoginSystem terdapat fungsi hash_password digunakan untuk mengubah kata sandi asli menjadi bentuk has menggunakan algoritma SHA256. Sehingga password tidak pernah disimpan dalam bentuk teks biasa. Fungsi ini menambah keamanan kata sandi pengguna.

```python
def register(self, username, password, nama, role="guru"):
    if username in self.users:
        return False, "Username sudah terdaftar"

    self.users[username] = {
        "password": self.hash_password(password),
        "role": role,
        "nama": nama
    }
    self.save_users()
    return True, "Registrasi berhasil"
```

Fungsi register digunakan untuk menambahkan akun baru dengan terlebih dahulu mengecek apakah username sudah terdaftar, lalu menyimpan data pengguna serta password yang sudah di-hash agar lebih aman.

```
def login(self, username, password):
    if username not in self.users:
        return False, "Username tidak ditemukan"

    if self.users[username]["password"] != self.hash_password(password):
        return False, "Password salah"

    self.current_user = {
        "username": username,
        "role": self.users[username]["role"],
        "nama": self.users[username]["nama"]
    }
    return True, "Login berhasil"
```

Fungsi login digunakan untuk memverifikasi username dan password dengan cara mencocokan hash password yang dimasukkan dengan data yang tersimpan.

b. Penghitungan Rata-rata

Fitur pengolahan nilai rata-rata pada aplikasi SiRapor merupakan bagian penting yang digunakan untuk menghitung nilai akhir siswa berdasarkan nilai-nilai mata pelajaran yang dimasukkan. Fungsi ini menjumlahkan seluruh nilai yang valid, memeriksa apakah setiap nilai berada dalam rentang yang diperbolehkan, lalu membaginya dengan jumlah mata pelajaran yang diisi untuk menghasilkan nilai rata-rata.

```
def hitung_rata_rata(self):
    nama = self.entry_nama.get().strip()
    kelas = self.entry_kelas.get().strip()

    if not nama:
        messagebox.showerror("Error", "Nama siswa harus diisi!")
        return

    if not kelas:
        messagebox.showerror("Error", "Kelas harus diisi!")
        return

    total_nilai = 0
    jumlah_mapel = 0
    nilai_mapel = {}
```

Proses tersebut dijalankan melalui potongan code hitung_rata_rata() yang menangani pengambilan input, validasi nilai, sehingga menghasilkan rata-rata akhir yang kemudian siap disimpan ke dalam data aplikasi.

Data yang telah dihitung nilai rata-ratanya akan disimpan ke dalam sistem. Agar data tersebut tidak hilang dan dapat diakses kembali kapan saja, fungsi simpan_data() digunakan untuk mencatatnya ke dalam file CSV. Proses penyimpanan ini hanya dapat dilakukan setelah perhitungan nilai selesai, sehingga memastikan bahwa data yang tersimpan adalah data yang valid dan lengkap.

c. Status Nilai

Fitur status pada SiRapor digunakan untuk mempermudah guru dalam melihat status tiap siswa, mengevaluasi hasil akhir siswa, serta memantau siswa yang perlu perhatian khusus, sehingga membantu guru dalam pengambilan keputusan dan pelaporan hasil belajar siswa.

```
def hitung_status(self, rata_rata):
    """Menentukan status berdasarkan nilai rata-rata"""
    if rata_rata >= 91:
        return "SANGAT BAIK", 'sangat_baik'
    elif rata_rata >= 88:
        return "BAIK", 'baik'
    elif rata_rata >= 84:
        return "CUKUP", 'cukup'
    else:
        return "PERLU BIMBINGAN", 'bimbingan'
```

Dalam fungsi hitung_status ini membantu menentukan kategori status berdasarkan nilai rata-rata siswa yang diinputkan. Lalu fungsi ini ditampilkan di beberapa bagian UI (hitung nilai rata-rata, leaderboard, pencarian siswa, serta export PDF)

d. Urutan Peringkat

Fitur urutan peringkat (Leaderboard) kelas pada aplikasi SiRapor digunakan untuk menampilkan urutan peringkat siswa berdasarkan nilai rata-rata yang telah disimpan ke dalam sistem. Fitur ini membantu guru melihat pencapaian siswa secara cepat dan terstruktur.

```
class SiRapor:
    def TAB_LEADERBOARD(self):
        header_label = ttk.Label(self.frame_leaderboard, text="LEADERBOARD NILAI SISWA PER KELAS", style='Header.TLabel')
        header_label.pack(pady=10)
```

Pertama, sistem menyediakan tab Leaderboard Kelas yang berisi tabel dengan beberapa kolom, salah satunya adalah kolom Rank.

```
columns = ('Rank', 'Nama', 'Kelas', 'Matematika', 'B. Indo', 'B. Inggris', 'Rata-rata', 'Status')
```

Kolom ini berfungsi untuk menunjukkan posisi atau peringkat siswa di dalam kelas berdasarkan hasil penilaian.

```
class SiRapor:
    def refresh_leaderboard(self, event=None):
        for item in self.tree.get_children():
            self.tree.delete(item)
```

Urutan peringkat ditentukan dari fungsi tersebut.

```
sorted_data = sorted(filtered_data, key=lambda x: x['rata_rata'], reverse=True)
```

Selanjutnya, ketika halaman leaderboard ditampilkan atau tombol Refresh Leaderboard ditekan, sistem akan menjalankan proses pembaruan data. Pada tahap ini, data siswa yang tersimpan akan diurutkan berdasarkan nilai rata-rata dari yang tertinggi ke terendah. Proses pengurutan ini memastikan bahwa siswa dengan nilai terbaik akan berada di posisi teratas.

```
for rank, siswa in enumerate(sorted_data, 1):
    status, tag = self.hitung_status(siswa['rata_rata'])

    self.tree.insert('', 'end', values=(
        rank,
        siswa['nama'],
        siswa['kelas'],
        siswa['nilai_mapel'].get('Matematika', '-'),
        siswa['nilai_mapel'].get('Bahasa Indonesia', '-'),
        siswa['nilai_mapel'].get('Bahasa Inggris', '-'),
        f"{siswa['rata_rata']:.2f}",
        status
    ), tags=(tag,))
```

Setelah data berhasil diurutkan, sistem secara otomatis memberikan nomor peringkat kepada setiap siswa. Peringkat dimulai dari angka satu untuk siswa dengan nilai rata-rata tertinggi, kemudian dilanjutkan ke siswa berikutnya secara berurutan. Nomor peringkat tersebut ditampilkan pada kolom rank di tabel leaderboard.

Selain menampilkan peringkat, sistem juga menampilkan status pencapaian siswa seperti Sangat Baik, Baik, Cukup, atau Perlu Bimbingan. Untuk memudahkan pemahaman, setiap status ditandai dengan warna latar yang berbeda pada tabel, sehingga guru dapat dengan cepat mengidentifikasi kondisi akademik siswa.

Dengan adanya fitur leaderboard ini, proses evaluasi nilai menjadi lebih jelas, objektif, dan mudah dipahami, baik dalam melihat peringkat siswa secara keseluruhan maupun berdasarkan kelas tertentu.

e. Ekspor PDF

Fitur export PDF pada aplikasi SiRapor berfungsi untuk mempermudah guru dalam menghasilkan laporan nilai siswa dalam bentuk dokumen yang rapi dan siap dibagikan. Dengan adanya fitur ini, guru dapat mencetak atau menyimpan rapor dalam format PDF secara fleksibel sesuai kebutuhan. Fitur

ini menjadi fitur terakhir dalam rangkaian fungsi SiRapor, menutup seluruh proses pengolahan data dengan menyediakan keluaran akhir berupa dokumen laporan resmi yang dapat digunakan untuk administrasi maupun evaluasi. Fitur export ini mendukung tiga pilihan output, yaitu export per siswa, export per kelas, serta export seluruh kelas, sehingga memberikan keleluasaan kepada pengguna untuk memilih laporan yang ingin dihasilkan.

```
    def export_single_pdf(self, student_data):
        filename = filedialog.asksaveasfilename(
            defaultextension=".pdf",
            filetypes=[("PDF files", "*.pdf")],
            title="Simpan Laporan PDF",
            initialfile=f"Raport_{student_data['nama']}_{student_data['kelas']}.pdf"
        )
```

Export per siswa digunakan ketika guru ingin mencetak atau menyimpan laporan rapor untuk satu siswa tertentu. Proses diawali dari pemilihan siswa melalui tabel hasil pencarian, kemudian data tersebut dikirim ke fungsi export_single_pdf() untuk membentuk dokumen rapor.

```
class SiRapor:
    def export_class_pdf(self):
        if not self.data_siswa:
            messagebox.showinfo("Info", "Tidak ada data untuk diexport!")
            return
```

Export per kelas digunakan ketika guru membutuhkan laporan nilai seluruh siswa dalam satu kelas tertentu. Program menampilkan dialog pemilihan kelas, kemudian menghasilkan PDF yang berisi data lengkap seluruh siswa pada kelas tersebut. Fungsi export_class_pdf() inilah yang nantinya akan menampilkan data berdasarkan kategori kelasnya.

```
class SiRapor:
    def export_all_pdf(self):
        if not self.data_siswa:
            messagebox.showinfo("Info", "Tidak ada data untuk diexport!")
            return
```

Pada fungsi export_all_pdf() yang terdapat pada class SiRapor tersebut nantinya akan menghasilkan export seluruh kelas menghasilkan laporan lengkap berisi semua siswa, dari seluruh kelas yang ada. PDF ini juga nantinya akan menampilkan statistik umum seperti jumlah siswa, nilai tertinggi, terendah, dan rata-rata keseluruhan.

LAMPIRAN

Kode Program: https://github.com/xh420/SiRapor