

Laporan UTS Kelompok Sukuna Survivors

Automated Bubble Sheet Exam Grading



DISUSUN OLEH:

211110121 - Grace Putri Wijaya

211112075- Rio Frederich

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS MIKROSKIL
MEDAN
2025**

BAB I: PENDAHULUAN

1. Latar Belakang

Proses koreksi ujian secara manual, terutama untuk kelas dengan jumlah siswa yang besar, merupakan tugas yang memakan waktu dan rentan terhadap kesalahan manusia (human error). Guru harus memeriksa setiap jawaban satu per satu, menghitung skor, dan merekapitulasi nilai. Aplikasi EZ Exam Grader dibuat untuk mengatasi masalah ini dengan menyediakan solusi otomatis yang cepat, akurat, dan fleksibel.

2. Tujuan Proyek

- **Mengotomatisasi Penilaian:** Membuat sistem yang dapat menilai berbagai tipe soal (Pilihan Ganda, Isian Singkat, dan Esai) secara efisien.
- **Menerapkan Desain SOLID:** Membangun aplikasi dengan arsitektur yang kuat, di mana setiap komponen memiliki tanggung jawab tunggal, dan sistem secara keseluruhan mudah untuk diperluas (misalnya, menambah tipe soal baru) tanpa mengubah kode yang sudah ada.
- **Menyediakan Laporan Otomatis:** Menghasilkan rekapitulasi nilai dalam format Excel yang mudah dibaca dan diarsipkan.
- **Menyediakan Antarmuka Pengguna yang Ramah:** Membuat aplikasi dengan menu yang jelas dan interaktif untuk kemudahan penggunaan oleh guru.

BAB II: ARSITEKTUR DAN DESAIN SISTEM

1. Struktur Proyek

```
exam_grader/      # Kode sumber utama aplikasi
    app.py        # Logika utama dan alur aplikasi
    grading/      # Mesin penilai (Grader, IGradingType, dll.)
    models.py     # Kelas data (AnswerKey, StudentResult)
    reporting/   # Logika pembuatan laporan (Excel)
    utils/        # Kelas utilitas (UserInterface, FileManager)

input_data/        # Berisi semua data ujian
    answer_keys/  # File-file kunci jawaban .json
    student_answers/ # Folder-folder jawaban siswa .json

output/           # Folder tempat laporan Excel disimpan
generate_sample_data.py # Skrip untuk membuat data siswa
generate_answer_keys.py # Skrip untuk membuat kunci jawaban
main.py          # Titik masuk untuk menjalankan aplikasi
```

2. Penerapan Prinsip SOLID

- **(S) Single Responsibility Principle:** Setiap kelas memiliki satu tugas utama. FileManager hanya mengelola file, Grader hanya menghitung nilai, dan ExcelReportGenerator hanya membuat file Excel.
- **(O) Open/Closed Principle:** Sistem ini **terbuka untuk ekstensi**. Untuk menambah tipe soal baru (misalnya "Menjodohkan"), kita hanya perlu membuat kelas MatchingGradingType baru tanpa harus **mengubah** kode di dalam kelas Grader yang sudah ada.
- **(L) Liskov Substitution Principle:** Kelas Grader dapat bekerja dengan objek MCQGradingType atau EssayGradingType secara bergantian melalui interface IGradingType tanpa mengalami masalah.
- **(I) Interface Segregation Principle:** Interface yang dibuat spesifik dan ramping, seperti IGradingType, yang hanya mendefinisikan kontrak untuk fungsi grade.
- **(D) Dependency Inversion Principle:** Modul tingkat tinggi tidak bergantung pada modul tingkat rendah. Contoh terbaik adalah EssayGradingType yang bergantung pada abstraksi UserInterface, bukan pada implementasi terminal secara langsung. UserInterface "disuntikkan" (injected) ke dalamnya, membuatnya sangat fleksibel dan mudah diuji.

BAB III: DEMONSTRASI PENGGUNAAN APLIKASI

Berikut adalah alur penggunaan aplikasi dari awal hingga akhir, yang akan diilustrasikan dengan screenshot.

1. Persiapan Data

Sebelum memulai, data sampel dibuat menggunakan skrip yang telah disediakan.

1. python generate_answer_keys.py: Membuat file kunci jawaban.
2. python generate_sample_data.py: Membuat file jawaban siswa.

The image shows a file explorer window on the left and a code editor window on the right. The file explorer contains a folder named 'Input_Data' which contains subfolders for 'answer_keys' (with 'Bahasa_Inggris_X.json' and 'XL_key.json'), 'Biology' (with 'XII_key.json'), and 'Matematika' (with 'X_key.json'). It also contains 'student_answers' and 'Barisan_Inggris' folders, both containing 'X_A' and 'X_B' subfolders with various student JSON files like 'Alya_Salsabila.json', 'Budi_Hartono.json', etc. A large bracket on the left side groups all these files under the heading 'Generated by program'. The code editor shows the Python script 'generate_answer_keys.py' with code that generates answers for students based on their names and class (X-A or X-B).

```
1 K
2   *student_info: {
3     "name": "Alya_Salsabila",
4     "class": "X_A"
5   },
6   *answers: {
7     "mcq_section": {
8       "1": "C",
9       "2": "A",
10      "3": "A",
11      "4": "C",
12      "5": "D",
13      "6": "A",
14      "7": "D",
15      "8": "C",
16      "9": "C",
17      "10": "A"
18    },
19    "short_answer_section": {
20      "1": "seusai dengan buku teks",
21      "2": "Jawaban ini benar",
22      "3": "Jawaban ini benar",
23      "4": "Sayang tidak tahu",
24      "5": "Jawaban ini salah"
25    },
26    "essay_section": {
27      "1": "Setuju dengan pernyataan tersebut karena data mendukung.",
28      "2": "Menurut pendapat saya, karakter utama menunjukkan keberanian."
29    }
30  }
31 }
```

2. Menjalankan Aplikasi dan Menu Utama

Aplikasi dijalankan dengan perintah python main.py. Pengguna akan disambut dengan menu utama.

```
=====
WELCOME TO EZ EXAM GRADER!
📚 Malas Cek Nilai Sendiri? We Gotchu! ✅
=====

Menu:
1. Cara Pemakaian
2. Mulai Penilaian
3. Keluar
Pilih menu (1-3):
Pilihan tidak valid. Silakan coba lagi.

=====
WELCOME TO EZ EXAM GRADER!
📚 Malas Cek Nilai Sendiri? We Gotchu! ✅
=====

Menu:
1. Cara Pemakaian
2. Mulai Penilaian
3. Keluar
Pilih menu (1-3):
```

Pengguna memilih opsi "2. Mulai Penilaian" untuk melanjutkan.

3. Memasukkan Path Data

Aplikasi kemudian meminta dua path penting:

1. **Path folder jawaban siswa:** input_data/student_answers/Biologi/XII_B_MIA
2. **Path file kunci jawaban:** input_data/answer_keys/Matematika/XII_key.json

```
--- Memulai Sesi Penilaian ---
Masukkan path ke folder jawaban siswa: input_data/student_answers/Biologi/XII_B_MIA
Masukkan path ke file kunci jawaban (.json): input_data/answer_keys/Biologi/XII_key.json

Path diterima. Memulai proses penilaian...
```

3. Proses Penilaian

Setelah path dikonfirmasi, proses penilaian dimulai.

- **Penilaian Otomatis:** Soal Pilihan Ganda dan Isian Singkat dinilai secara otomatis oleh program. Output di terminal (dalam mode debug) menunjukkan analisis per jawaban.
- **Penilaian Manual (Esai):** Ketika program menemukan soal esai, ia akan berhenti sejenak, menampilkan jawaban siswa, dan meminta guru untuk memasukkan nilai secara manual. Ini menunjukkan interaktivitas program.

```

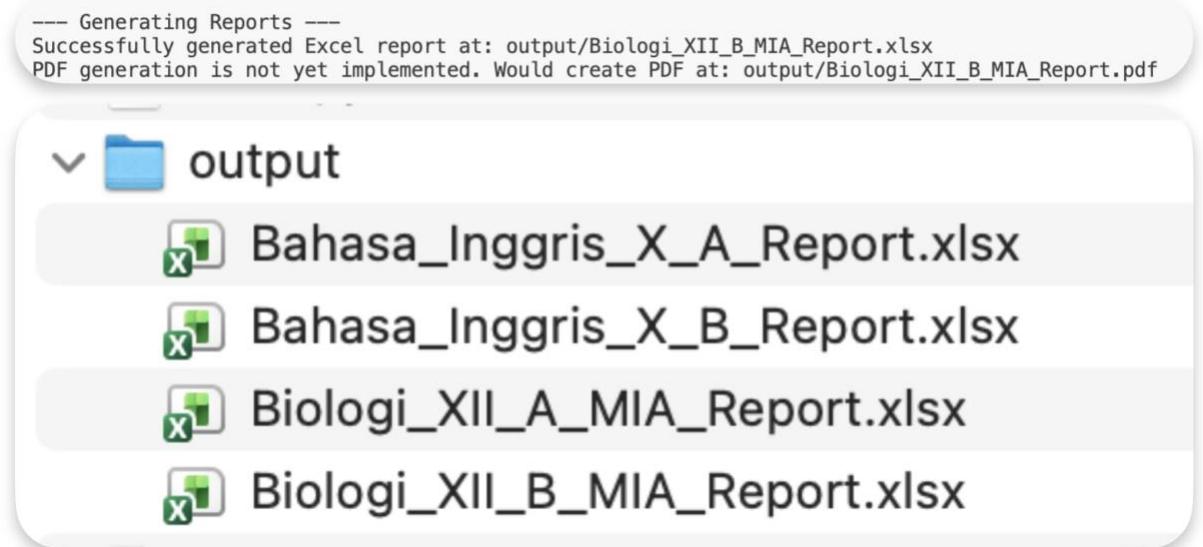
=====
--- DEBUG MODE: Analyzing Sheet for Ghani Ramadhan ---
=====
----- Penilaian Manual untuk: Ghani Ramadhan -----
Pertanyaan: Jelaskan apa itu mitokondria.
Jawaban Siswa: "Saya setuju dengan pernyataan tersebut karena data mendukung."
Masukkan nilai (0-10): 0
----- Penilaian Manual untuk: Ghani Ramadhan -----
Pertanyaan: Jelaskan proses fotosintesis.
Jawaban Siswa: "Menurut pendapat saya, karakter utama menunjukkan keberanian."
Masukkan nilai (0-10): 2
=====
Final Calculated Score (from logic): 12.00%
=====

=====
--- DEBUG MODE: Analyzing Sheet for Farel Akbar ---
=====
----- Penilaian Manual untuk: Farel Akbar -----
Pertanyaan: Jelaskan apa itu mitokondria.
Jawaban Siswa: "Pesan moral dari cerita ini adalah pentingnya persahabatan."
Masukkan nilai (0-10): 10
----- Penilaian Manual untuk: Farel Akbar -----
Pertanyaan: Jelaskan proses fotosintesis.
Jawaban Siswa: "Pesan moral dari cerita ini adalah pentingnya persahabatan."
Masukkan nilai (0-10): 0
=====
Final Calculated Score (from logic): 15.00%
=====
```

Proses ini berlanjut hingga semua siswa di dalam folder tersebut selesai dinilai.

4. Proses Hasil Akhir: Laporan Excel

Setelah semua siswa dinilai, program akan mengonfirmasi bahwa sesi telah selesai dan sebuah laporan Excel telah dibuat di dalam folder output/.



Laporan ini berisi rekapitulasi nilai yang rapi untuk seluruh kelas, yang siap diarsipkan atau dibagikan.

E5

	A	B	C	D
1	number	name	class_name	score
2		1 Grace Putri Wijaya	XII A MIA	95
3		2 Bella Ananda	XII A MIA	45
4		3 Rio Frederich	XII A MIA	94

Setelah sesi selesai, pengguna dapat kembali ke menu utama untuk menilai kelas lain atau keluar dari aplikasi.

BAB IV: KESIMPULAN

Aplikasi EZ Exam Grader berhasil didesain dan diimplementasikan sebagai sebuah sistem yang fungsional dan kokoh. Dengan memanfaatkan prinsip-prinsip desain SOLID, aplikasi ini tidak hanya memenuhi kebutuhan saat ini tetapi juga memiliki fondasi yang kuat untuk pengembangan di masa depan.

Keunggulan utama dari desain ini adalah fleksibilitasnya. Transisi dari data simulasi (JSON) ke data nyata dari OMR/OCR di masa depan (UAS) dapat dilakukan dengan hanya membuat kelas ImageProcessor baru, tanpa perlu merombak alur kerja dan logika bisnis yang sudah ada. Ini membuktikan kekuatan dari arsitektur yang modular dan terabstraksi dengan baik.