

LAPORAN AKHIR PROYEK

Aplikasi Latihan Soal UTBK



Nama Mahasiswa: Sulisa Dermawati Br Sinulingga

NIM: 25031554012

Kelas: 2025 G

Mata Kuliah: Pemogarman Dasar

Dosen Pengampu:

1. Hasanuddin Al-Habib, S.Si., M.Si.
2. 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
A. Latar Belakang	1
B. Rumusan Masalah	1
C. Tujuan	1
BAB II ANALISIS DAN PERANCANGAN	3
Analisis Kebutuhan Aplikasi.....	3
Diagram Alur	3
Sketsa Desain Antarmuka	4
BAB III IMPLEMENTASI.....	5
A. Penjelasan Kode Program	5
B. Manual Penggunaan Aplikasi.....	5
C. Screenshoot Aplikasi.....	6
BAB IV PENUTUP	8
A. Kesimpulan.....	7
B. Saran.....	8
LAMPIRAN	10
DAFTAR PUSTAKA	13

BAB I

PENDAHULUAN

A. Latar Belakang

Ujian Tulis Berbasis Komputer (UTBK) adalah salah satu tahapan penting bagi siswa yang ingin melanjutkan pendidikan ke perguruan tinggi negeri. Untuk menghadapi UTBK, siswa membutuhkan latihan soal yang cukup serta pembahasan yang jelas agar dapat memahami materi dengan baik.

Seiring dengan perkembangan teknologi, proses belajar tidak lagi terbatas pada buku cetak. Aplikasi digital dapat dijadikan solusi untuk membantu siswa belajar secara lebih fleksibel dan interaktif. Namun, masih banyak siswa yang kesulitan menemukan aplikasi latihan soal UTBK yang sederhana, mudah digunakan, dan memiliki pembahasan yang jelas.

Oleh karena itu, pada proyek ini dibuat sebuah aplikasi latihan soal UTBK berbasis Python dengan tampilan GUI (Graphical User Interface). Aplikasi ini diharapkan dapat membantu siswa dalam berlatih soal UTBK secara mandiri, efektif, dan mudah digunakan.

B. Rumusan Masalah

1. Bagaimana cara menyediakan soal-soal UTBK yang sesuai dengan kurikulum dan standar terbaru dalam bentuk aplikasi digital?
2. Bagaimana meningkatkan aksesibilitas dan efektivitas belajar siswa melalui aplikasi latihan soal UTBK?
3. Bagaimana menyajikan fitur pembahasan yang interaktif agar pengguna dapat memahami materi dengan lebih baik?
4. Apa saja tantangan teknis dan pedagogis dalam merancang aplikasi edukatif yang menarik dan mudah digunakan?

C. Tujuan

Sesuai dengan rumusan masalah yang telah dijabarkan di atas, maka dapat diketahui bahwa tujuan dari projek ini antara lain:

1. Menyediakan platform digital yang berisi kumpulan soal UTBK dari berbagai mata pelajaran secara terstruktur dan terstandarisasi.
2. Mempermudah siswa dalam mengakses latihan soal UTBK kapan saja dan di mana saja melalui perangkat mobile atau komputer.
3. Menyediakan fitur pembahasan dan penilaian otomatis untuk membantu siswa memahami kesalahan dan meningkatkan kemampuan.
4. Mendorong semangat belajar mandiri dan meningkatkan kesiapan siswa menghadapi UTBK secara lebih efektif.

BAB II

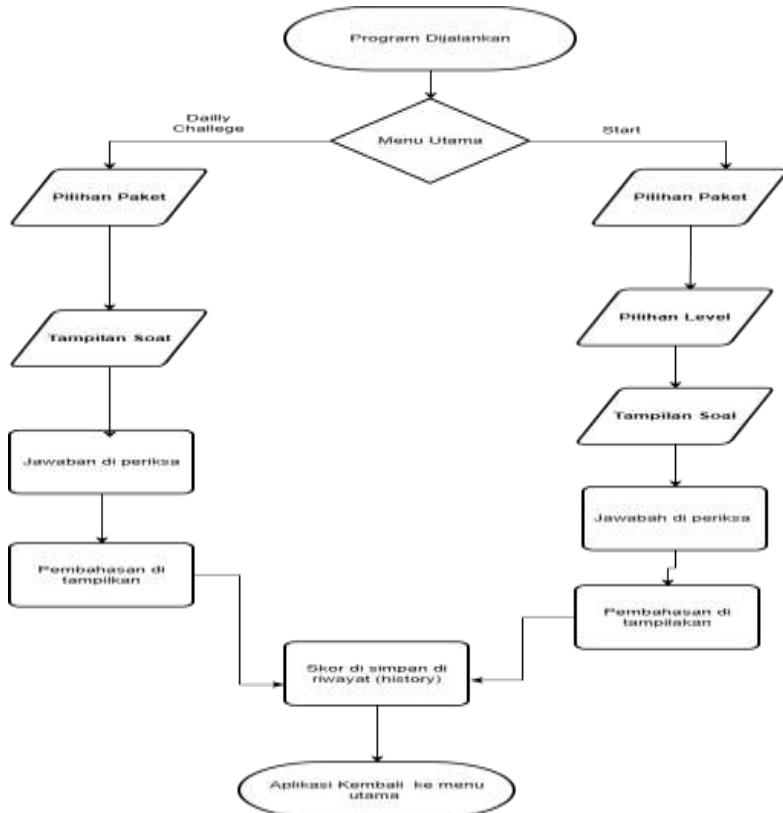
ANALISIS DAN PERANCANGAN

Analisis Kebutuhan Aplikasi

Aplikasi latihan soal UTBK ini memiliki kebutuhan sebagai berikut:

1. Menampilkan menu utama (Start, Daily Challenge, History).
2. Menampilkan pilihan paket soal (PK, PM, PPU, PBM, dll).
3. Menampilkan soal, pilihan jawaban, dan pembahasan.
4. Menghitung skor secara otomatis.
5. Menyimpan riwayat pengerjaan soal (history).
6. Menyediakan fitur timer.

Diagram Alur



Sketsa Desain Antarmuka

Desain aplikasi dibuat sederhana agar mudah digunakan oleh pengguna.

Tampilan aplikasi terdiri dari:

- Menu utama
- Menu paket soal
- Halaman soal
- Halaman history

BAB III

IMPLEMENTASI

A. Penjelasan Kode Program

Aplikasi ini dibuat menggunakan bahasa pemrograman **Python** dengan library **Tkinter** untuk membangun antarmuka grafis.

1. Struktur Program

Folder/File	Fungsi
src/	Tempat kode utama program
main.py	File utama yang dijalankan
utils.py	Berisi fungsi bantu (baca file, simpan skor, dll)
data/	Menyimpan data soal & history
soal.json	Database soal UTBK
history.json	Riwayat skor pengguna

2. Contoh Kode Penting

Contoh penggunaan class pada program :

```
24
25  # ===== APP =====
26  class QuizApp(tk.Tk):
27      def __init__(self):
28          super().__init__()
29          self.title("Latihan Soal UTBK")
30          self.resizable(False, False)
```

Kode tersebut digunakan untuk membuat window utama aplikasi menggunakan konsep **Object Oriented Programming (OOP)**.

B. Manual Penggunaan Aplikasi

- Jalankan file main.py
- Pilih menu **Start**
- Pilih paket dan level soal
- Jawab soal yang ditampilkan
- Klik tombol **Next** untuk melanjutkan
- Lihat pembahasan
- Riwayat dapat dilihat pada menu **History**

C. Screenshot Aplikasi

Gambar berikut merupakan tampilan **halaman utama (Home Screen)** dari aplikasi *Latihan Soal UTBK*.

Pada halaman ini, pengguna disediakan beberapa menu utama, yaitu:

- **Start** untuk memulai latihan soal UTBK berdasarkan paket dan tingkat kesulitan.
- **Daily Challenge** untuk mengerjakan tantangan harian dengan jumlah soal terbatas.
- **History** untuk melihat riwayat skor hasil pengerjaan sebelumnya.

Desain antarmuka aplikasi menggunakan warna pastel agar terlihat nyaman di mata dan mudah digunakan oleh pengguna.



BAB IV

PENUTUP

A. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian aplikasi latihan soal UTBK yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi latihan soal UTBK dibuat menggunakan bahasa pemrograman Python dengan bantuan library Tkinter sebagai antarmuka pengguna.
2. Aplikasi ini mampu menyediakan kumpulan soal UTBK dari beberapa paket mata pelajaran yang disusun secara terstruktur dan mudah diakses oleh pengguna.
3. Fitur utama seperti pemilihan paket soal, level kesulitan, timer, penilaian otomatis, pembahasan soal, serta riwayat penggerjaan telah berjalan dengan baik sesuai dengan tujuan awal proyek.
4. Penggunaan format data JSON dalam penyimpanan soal dan history memudahkan proses pengelolaan data serta mendukung konsep file access.
5. Secara keseluruhan, aplikasi ini dapat membantu siswa dalam melakukan latihan soal UTBK secara mandiri dan meningkatkan kesiapan dalam menghadapi ujian.

B. Saran

Untuk pengembangan aplikasi di masa mendatang, terdapat beberapa saran yang dapat dipertimbangkan, antara lain:

1. Menambahkan lebih banyak variasi soal dan paket mata pelajaran agar latihan menjadi lebih lengkap.
2. Mengembangkan aplikasi ke platform mobile (Android atau iOS) sehingga dapat diakses melalui smartphone.
3. Menambahkan fitur grafik atau statistik perkembangan nilai pengguna agar pengguna dapat memantau kemajuan belajar.
4. Meningkatkan tampilan antarmuka agar lebih interaktif dan menarik.
5. Menambahkan sistem akun pengguna agar data hasil latihan dapat tersimpan secara personal.

LAMPIRAN

```
  memory) % QUnitTest > CI\_js_
```

```
import QTimer as qt
from twisted import tests, messagebox
import random
import weakref
import database
import sys

from util import:
    ensure_dirs, load_questions_for_package, normalise_correct_answer,
    pick_questions_with_fresh_priority, pick_daily_challenge_by_level,
    load_json, write_json, HISTORY_FILE, list_packages_from_asml
)
```

```
# configuration constants
ensure_dirs()
```

```
QUESTIONS_PER_LEVEL = 4
SESSION_SECONDS = 75 * 60
DAILY_REFRESH = 7
DAILY_POINTS = {"easy": 1, "medium": 2, "hard": 4}
```

```
# application logic
class QuizApp(tw.TK):
    def __init__(self):
        super().__init__()
        self.title("Learn Haskell")
        self.style("darkblue", "darkgreen")
        try:
            self.state("zoomed")
        except Exception:
            pass
        self.style()
```

```
    def __init__(self, master):
        self.master = master
        self.container = tk.Frame(self.master)
        self.container.pack(fill='both', expand=True)

        self.history = load_json(HISTORY_FILE, {})

        self.packages = list_packages_from_repo()

        # stats
        self.questions = []
        self.id = 0
        self.score = 0
        self.daily_score = 0
        self.daily_breakdown = {"easy": 0, "medium": 0, "hard": 0}
        self.is_daily = False
        self.showing_explanation = False

        self.timer_label = None
        self.remaining = SESSION_SECONDS
        self.timer_id = None

        self.current_package = None
        self.current_level = None

        self._name()

    # ***** GUI *****

    def _style(self):
        ttk.Style().theme_use("clam")
        except Exception:
```

```

    <!-- package_main.xml -->
<apk>
    <file name="AndroidManifest.xml" type="text/utf-8"/>
    <file name="res/drawable/icon.png" type="image/png"/>
    <file name="res/layout/main.xml" type="text/utf-8"/>
    <file name="res/values/colors.xml" type="text/utf-8"/>
    <file name="src/com/example/test/TestActivity.java" type="text/utf-8"/>
</apk>

    <!-- package_main2.xml -->
<apk>
    <file name="AndroidManifest.xml" type="text/utf-8"/>
    <file name="res/drawable/icon.png" type="image/png"/>
    <file name="res/layout/main.xml" type="text/utf-8"/>
    <file name="res/values/colors.xml" type="text/utf-8"/>
    <file name="src/com/example/test/TestActivity.java" type="text/utf-8"/>
</apk>

    <!-- package_main3.xml -->
<apk>
    <file name="AndroidManifest.xml" type="text/utf-8"/>
    <file name="res/drawable/icon.png" type="image/png"/>
    <file name="res/layout/main.xml" type="text/utf-8"/>
    <file name="res/values/colors.xml" type="text/utf-8"/>
    <file name="src/com/example/test/TestActivity.java" type="text/utf-8"/>
</apk>

    <!-- package_main4.xml -->
<apk>
    <file name="AndroidManifest.xml" type="text/utf-8"/>
    <file name="res/drawable/icon.png" type="image/png"/>
    <file name="res/layout/main.xml" type="text/utf-8"/>
    <file name="res/values/colors.xml" type="text/utf-8"/>
    <file name="src/com/example/test/TestActivity.java" type="text/utf-8"/>
</apk>

```

```
public void testMethod() {
    // Test logic here
}

@Before
public void setup() {
    // Setup logic here
}

@After
public void teardown() {
    // Teardown logic here
}
```

```
public void testMethod() {
    // Test logic here
}

@Before
public void setup() {
    // Setup logic here
}

@After
public void teardown() {
    // Teardown logic here
}
```

```
public void testMethod() {
    // Test logic here
}

@Before
public void setup() {
    // Setup logic here
}

@After
public void teardown() {
    // Teardown logic here
}
```

```
public void testMethod() {
    // Test logic here
}

@Before
public void setup() {
    // Setup logic here
}

@After
public void teardown() {
    // Teardown logic here
}
```

```
public void testMethod() {
    // Test logic here
}

@Before
public void setup() {
    // Setup logic here
}

@After
public void teardown() {
    // Teardown logic here
}
```

```
public void testMethod() {
    // Test logic here
}

@Before
public void setup() {
    // Setup logic here
}

@After
public void teardown() {
    // Teardown logic here
}
```

```
public void testMethod() {
    // Test logic here
}

@Before
public void setup() {
    // Setup logic here
}

@After
public void teardown() {
    // Teardown logic here
}
```

```
public void testMethod() {
    // Test logic here
}

@Before
public void setup() {
    // Setup logic here
}

@After
public void teardown() {
    // Teardown logic here
}
```

```
def pick_daily_challenge_by_level(db, package, level='all', count=5):
    """Deterministic daily selection based on date + package + level.
    Returns up to 'count' items from db matching level.

    level = (level or 'all').strip().lower()
    pool = [q for q in (db or []) if level in (q['level'] or not q.get('level')) or _norm_pkg_name(q.get('package')) == package]
    if not pool:
        return []
    today = datetime.date.today().isoformat()
    seed_str = f'{today}::{package or ''}::{level}'
    seed = abs(hash(seed_str)) % 10**32
    rng = np.random.RandomState(seed)
    selected = rng.sample(pool, k=min(count, len(pool)))
    rng.shuffle(selected)
    return selected
```

DAFTAR PUSTAKA

pkbm.id. (2024). *Syarat, Ketentuan & Informasi Umum UTBK SNBT 2024.*

<https://pkbm.id/blog/syarat,-ketentuan-&-informasi-umum-utbk-snbt-2024>

Python Software Foundation. (2024). *Python 3 documentation.*

<https://docs.python.org/3/>