

LAPORAN AKHIR PROYEK

Aplikasi Latihan Soal UTBK



Nama Mahasiswa: Haid Syahira

NIM: 25031554057

Kelas: 2025 G

Mata Kuliah: Pemograman 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	7
A. Kesimpulan.....	7
B. Saran.....	7
LAMPIRAN.....	8
DAFTAR PUSTAKA	11

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 proyek 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 melalui perangkat 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. Screenshoot 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 pengerjaan 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 persona

LAMPIRAN

```

1 # -*- coding: utf-8 -*-
2
3 import tkinter as tk
4 from tkinter import ttk, messagebox
5 import random
6 import uuid
7 import datetime
8 import sys
9
10 from utils import (
11     ensure_dir, load_questions_for_package, normalize_correct_answer,
12     pick_questions_with_fresh_priority, pick_daily_challenge_by_level,
13     load_json, save_json, HISTORY_FILE, list_packages_from_json
14 )
15
16 # ===== CONFIG =====
17 ensure_dir()
18
19 QUESTIONS_PER_LEVEL = 8
20 SESSION_SECONDS = 75 * 60
21 DAILY_NUM = 5
22 DAILY_POINTS = {"easy": 1, "medium": 2, "hard": 3}
23
24 # ===== APP =====
25 class QuizApp(tk.Tk):
26     def __init__(self):
27         super().__init__()
28         self.title("Latihan Soal UTS")
29         try:
30             self.state("normal")
31         except Exception:
32             pass
33
34         self.style()

```

```

35
36     # ===== INIT =====
37     def _init_(self):
38         self.container = tk.Frame(self)
39         self.container.pack(fill="both", expand=True)
40
41         self.history = load_json(HISTORY_FILE, [])
42         self.packages = list_packages_from_json()
43
44         # state
45         self.questions = []
46         self.idx = 0
47         self.score = 0
48         self.daily_score = 0
49         self.daily_breakdown = {"easy": 0, "medium": 0, "hard": 0}
50         self.is_daily = False
51         self.showing_explanation = False
52
53         self.timer_level = None
54         self.remaining = SESSION_SECONDS
55         self.timer_job = None
56
57         self.current_package = None
58         self.current_level = None
59
60         self._home()
61
62     # ===== STYLE =====
63     def _style(self):
64         s = ttk.Style()
65         try:
66             s.theme_use("clam")
67         except Exception:
68             pass

```

```

69
70     # ===== TTK =====
71     def _style(self):
72         bg = "#f0f0f0"
73         text = "#333333"
74         btn = "#ffffff"
75         hover = "#cccccc"
76
77         s.configure("Frame", background=bg)
78         s.configure("Label", background=bg, foreground=text)
79         s.configure("Title.Label", font=("Segoe UI", 16, "bold"))
80         s.configure("Subtitle.Label", font=("Segoe UI", 14))
81
82         s.configure(
83             "Button",
84             font=("Segoe UI", 16),
85             padding=11,
86             background=bg,
87             foreground=text,
88         )
89         s.map("Button", background=[("active", hover)])
90
91         s.configure(
92             "Choice.Radiobutton",
93             background=bg,
94             foreground=text,
95             font=("Segoe UI", 16),
96             padding=6
97         )
98         s.map("Choice.Radiobutton", background=[("active", hover)])
99
100     # ===== TTK =====
101     def _clear(self):
102         # Clear the self.container (only widget)

```

```

103
104     # ===== TTK =====
105     def _start_timer(self):
106         if self.timer_job:
107             try:
108                 self.after_cancel(self.timer_job)
109             except Exception:
110                 pass
111             self.timer_job = None
112
113         self.remaining = SESSION_SECONDS
114         self.timer_job = self.after(1000, self._tick)
115
116     def _tick(self):
117         try:
118             s = divmod(self.remaining, 60)
119             if self.timer_label and getattr(self.timer_label, "info_exists", lambda: False)():
120                 self.timer_label.config(text="%02d:%02d" % (s[0], s[1]))
121             except tk.TclError:
122                 pass
123             except Exception:
124                 pass
125
126         if self.remaining <= 0:
127             if self.timer_job:
128                 try:
129                     self.after_cancel(self.timer_job)
130                 except Exception:
131                     pass

```

```

132
133     # ===== TTK =====
134     def _home(self):
135         # Messagebox about "both, easy, hard, none"
136         except Exception:
137             pass
138         self._init()
139
140         self.remaining = s
141         try:
142             self.timer_job = self.after(1000, self._tick)
143         except Exception:
144             pass
145         self.timer_job = None
146
147     # ===== TTK =====
148     def _home(self):
149         if self.timer_job:
150             try:
151                 self.after_cancel(self.timer_job)
152             except Exception:
153                 pass
154             self.timer_job = None
155
156         self._clear()
157
158         btn = tk.Button(self.container)
159         btn.grid(column=0, row=0, sticky="nsew")
160
161         btn_label(btn, text="Start", style="Title.Label", pack(pady=10))
162         btn_label(btn, text="Easy", style="Title.Label", pack(pady=10))
163         btn_label(btn, text="Medium", style="Title.Label", pack(pady=10))
164         btn_label(btn, text="Hard", style="Title.Label", pack(pady=10))

```

```

165
166     # ===== TTK =====
167     def _package_menu(self):
168         self._clear()
169         btn = tk.Button(self.container)
170         btn.grid(column=0, row=0, sticky="nsew")
171
172         btn_label(btn, text="V1.0: Easy", style="Title.Label", pack(pady=10))
173         if not self.packages:
174             btn_label(btn, text="No package available", style="Title.Label", pack(pady=10))
175         for p in self.packages:
176             btn_label(btn, text=p["name"],
177                 command=lambda: self._load_menu(p["package"]),
178                 pack(pady=10))
179
180     def _load_menu(self, package):
181         self._clear()
182         btn = tk.Button(self.container)
183         btn.grid(column=0, row=0, sticky="nsew")
184
185         btn_label(btn, text="Start", style="Title.Label", pack(pady=10))
186         for i in range(1, len(package)):
187             btn_label(btn, text="Question %d" % i, style="Title.Label", pack(pady=10))
188             btn_label(btn, text="Answer", style="Title.Label", pack(pady=10))
189             btn_label(btn, text="Explanation", style="Title.Label", pack(pady=10))

```

[illegible]

```

1 # Import the necessary libraries
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 from sklearn.preprocessing import StandardScaler
6 from sklearn.metrics import mean_squared_error
7 from sklearn.linear_model import LinearRegression
8 from sklearn.model_selection import train_test_split
9
10 # Load the data
11 data = pd.read_csv('data.csv')
12
13 # Split the data into training and testing sets
14 X_train, X_test, y_train, y_test = train_test_split(
15     data[['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8', 'x9', 'x10', 'x11', 'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x19', 'x20', 'x21', 'x22', 'x23', 'x24', 'x25', 'x26', 'x27', 'x28', 'x29', 'x30', 'x31', 'x32', 'x33', 'x34', 'x35', 'x36', 'x37', 'x38', 'x39', 'x40', 'x41', 'x42', 'x43', 'x44', 'x45', 'x46', 'x47', 'x48', 'x49', 'x50', 'x51', 'x52', 'x53', 'x54', 'x55', 'x56', 'x57', 'x58', 'x59', 'x60', 'x61', 'x62', 'x63', 'x64', 'x65', 'x66', 'x67', 'x68', 'x69', 'x70', 'x71', 'x72', 'x73', 'x74', 'x75', 'x76', 'x77', 'x78', 'x79', 'x80', 'x81', 'x82', 'x83', 'x84', 'x85', 'x86', 'x87', 'x88', 'x89', 'x90', 'x91', 'x92', 'x93', 'x94', 'x95', 'x96', 'x97', 'x98', 'x99', 'x100', 'x101', 'x102', 'x103', 'x104', 'x105', 'x106', 'x107', 'x108', 'x109', 'x110', 'x111', 'x112', 'x113', 'x114', 'x115', 'x116', 'x117', 'x118', 'x119', 'x120', 'x121', 'x122', 'x123', 'x124', 'x125', 'x126', 'x127', 'x128', 'x129', 'x130', 'x131', 'x132', 'x133', 'x134', 'x135', 'x136', 'x137', 'x138', 'x139', 'x140', 'x141', 'x142', 'x143', 'x144', 'x145', 'x146', 'x147', 'x148', 'x149', 'x150', 'x151', 'x152', 'x153', 'x154', 'x155', 'x156', 'x157', 'x158', 'x159', 'x160', 'x161', 'x162', 'x163', 'x164', 'x165', 'x166', 'x167', 'x168', 'x169', 'x170', 'x171', 'x172', 'x173', 'x174', 'x175', 'x176', 'x177', 'x178', 'x179', 'x180', 'x181', 'x182', 'x183', 'x184', 'x185', 'x186', 'x187', 'x188', 'x189', 'x190', 'x191', 'x192', 'x193', 'x194', 'x195', 'x196', 'x197', 'x198', 'x199', 'x200', 'x201', 'x202', 'x203', 'x204', 'x205', 'x206', 'x207', 'x208', 'x209', 'x210', 'x211', 'x212', 'x213', 'x214', 'x215', 'x216', 'x217', 'x218', 'x219', 'x220', 'x221', 'x222', 'x223', 'x224', 'x225', 'x226', 'x227', 'x228', 'x229', 'x230', 'x231', 'x232', 'x233', 'x234', 'x235', 'x236', 'x237', 'x238', 'x239', 'x240', 'x241', 'x242', 'x243', 'x244', 'x245', 'x246', 'x247', 'x248', 'x249', 'x250', 'x251', 'x252', 'x253', 'x254', 'x255', 'x256', 'x257', 'x258', 'x259', 'x260', 'x261', 'x262', 'x263', 'x264', 'x265', 'x266', 'x267', 'x268', 'x269', 'x270', 'x271', 'x272', 'x273', 'x274', 'x275', 'x276', 'x277', 'x278', 'x279', 'x280', 'x281', 'x282', 'x283', 'x284', 'x285', 'x286', 'x287', 'x288', 'x289', 'x290', 'x291', 'x292', 'x293', 'x294', 'x295', 'x296', 'x297', 'x298', 'x299', 'x300', 'x301', 'x302', 'x303', 'x304', 'x305', 'x306', 'x307', 'x308', 'x309', 'x310', 'x311', 'x312', 'x313', 'x314', 'x315', 'x316', 'x317', 'x318', 'x319', 'x320', 'x321', 'x322', 'x323', 'x324', 'x325', 'x326', 'x327', 'x328', 'x329', 'x330', 'x331', 'x332', 'x333', 'x334', 'x335', 'x336', 'x337', 'x338', 'x339', 'x340', 'x341', 'x342', 'x343', 'x344', 'x345', 'x346', 'x347', 'x348', 'x349', 'x350', 'x351', 'x352', 'x353', 'x354', 'x355', 'x356', 'x357', 'x358', 'x359', 'x360', 'x361', 'x362', 'x363', 'x364', 'x365', 'x366', 'x367', 'x368', 'x369', 'x370', 'x371', 'x372', 'x373', 'x374', 'x375', 'x376', 'x377', 'x378', 'x379', 'x380', 'x381', 'x382', 'x383', 'x384', 'x385', 'x386', 'x387', 'x388', 'x389', 'x390', 'x391', 'x392', 'x393', 'x394', 'x395', 'x396', 'x397', 'x398', 'x399', 'x400', 'x401', 'x402', 'x403', 'x404', 'x405', 'x406', 'x407', 'x408', 'x409', 'x410', 'x411', 'x412', 'x413', 'x414', 'x415', 'x416', 'x417', 'x418', 'x419', 'x420', 'x421', 'x422', 'x423', 'x424', 'x425', 'x426', 'x427', 'x428', 'x429', 'x430', 'x431', 'x432', 'x433', 'x434', 'x435', 'x436', 'x437', 'x438', 'x439', 'x440', 'x441', 'x442', 'x443', 'x444', 'x445', 'x446', 'x447', 'x448', 'x449', 'x450', 'x451', 'x452', 'x453', 'x454', 'x455', 'x456', 'x457', 'x458', 'x459', 'x460', 'x461', 'x462', 'x463', 'x464', 'x465', 'x466', 'x467', 'x468', 'x469', 'x470', 'x471', 'x472', 'x473', 'x474', 'x475', 'x476', 'x477', 'x478', 'x479', 'x480', 'x481', 'x482', 'x483', 'x484', 'x485', 'x486', 'x487', 'x488', 'x489', 'x490', 'x491', 'x492', 'x493', 'x494', 'x495', 'x496', 'x497', 'x498', 'x499', 'x500', 'x501', 'x502', 'x503', 'x504', 'x505', 'x506', 'x507', 'x508', 'x509', 'x510', 'x511', 'x512', 'x513', 'x514', 'x515', 'x516', 'x517', 'x518', 'x519', 'x520', 'x521', 'x522', 'x523', 'x524', 'x525', 'x526', 'x527', 'x528', 'x529', 'x530', 'x531', 'x532', 'x533', 'x534', 'x535', 'x536', 'x537', 'x538', 'x539', 'x540', 'x541', 'x542', 'x543', 'x544', 'x545', 'x546', 'x547', 'x548', 'x549', 'x550', 'x551', 'x552', 'x553', 'x554', 'x555', 'x556', 'x557', 'x558', 'x559', 'x560', 'x561', 'x562', 'x563', 'x564', 'x565', 'x566', 'x567', 'x568', 'x569', 'x570', 'x571', 'x572', 'x573', 'x574', 'x575', 'x576', 'x577', 'x578', 'x579', 'x580', 'x581', 'x582', 'x583', 'x584', 'x585', 'x586', 'x587', 'x588', 'x589', 'x590', 'x591', 'x592', 'x593', 'x594', 'x595', 'x596', 'x597', 'x598', 'x599', 'x600', 'x601', 'x602', 'x603', 'x604', 'x605', 'x606', 'x607', 'x608', 'x609', 'x610', 'x611', 'x612', 'x613', 'x614', 'x615', 'x616', 'x617', 'x618', 'x619', 'x620', 'x621', 'x622', 'x623', 'x624', 'x625', 'x626', 'x627', 'x628', 'x629', 'x630', 'x631', 'x632', 'x633', 'x634', 'x635', 'x636', 'x637', 'x638', 'x639', 'x640', 'x641', 'x642', 'x643', 'x644', 'x645', 'x646', 'x647', 'x648', 'x649', 'x650', 'x651', 'x652', 'x653', 'x654', 'x655', 'x656', 'x657', 'x658', 'x659', 'x660', 'x661', 'x662', 'x663', 'x664', 'x665', 'x666', 'x667', 'x668', 'x669', 'x670', 'x671', 'x672', 'x673', 'x674', 'x675', 'x676', 'x677', 'x678', 'x679', 'x680',
```

```

def __init__(self, name, age, sex, height):
    self.name = name
    self.age = age
    self.sex = sex
    self.height = height

    # Create a list of attributes
    self.attributes = [
        'name', 'age', 'sex', 'height'
    ]

    # Create a dictionary of attributes
    self.attributes_dict = {
        'name': self.name,
        'age': self.age,
        'sex': self.sex,
        'height': self.height
    }

    # Create a list of methods
    self.methods = [
        'get_name', 'get_age', 'get_sex', 'get_height'
    ]

    # Create a dictionary of methods
    self.methods_dict = {
        'get_name': self.get_name,
        'get_age': self.get_age,
        'get_sex': self.get_sex,
        'get_height': self.get_height
    }

    # Create a list of properties
    self.properties = [
        'name', 'age', 'sex', 'height'
    ]

    # Create a dictionary of properties
    self.properties_dict = {
        'name': self.name,
        'age': self.age,
        'sex': self.sex,
        'height': self.height
    }

    # Create a list of methods
    self.methods = [
        'get_name', 'get_age', 'get_sex', 'get_height'
    ]

    # Create a dictionary of methods
    self.methods_dict = {
        'get_name': self.get_name,
        'get_age': self.get_age,
        'get_sex': self.get_sex,
        'get_height': self.get_height
    }

    # Create a list of properties
    self.properties = [
        'name', 'age', 'sex', 'height'
    ]

    # Create a dictionary of properties
    self.properties_dict = {
        'name': self.name,
        'age': self.age,
        'sex': self.sex,
        'height': self.height
    }

```

```

10 // * imports *
11 #include <iostream>
12 #include <vector>
13 #include <string>
14 #include <algorithm>
15 #include <map>
16 #include <set>
17 #include <stack>
18 #include <queue>
19 #include <deque>
20 #include <list>
21 #include <bitset>
22 #include <unordered_map>
23 #include <unordered_set>
24 #include <memory>
25 #include <functional>
26 #include <numeric>
27 #include <random>
28 #include <chrono>
29 #include <regex>
30 #include <thread>
31 #include <atomic>
32 #include <mutex>
33 #include <condition_variable>
34 #include <future>
35 #include <promise>
36 #include <shared_ptr>
37 #include <weak_ptr>
38 #include <any>
39 #include <variant>
40 #include <optional>
41 #include <variant>
42 #include <string_view>
43 #include <filesystem>
44 #include <memory>
45 #include <memory>
46 #include <memory>
47 #include <memory>
48 #include <memory>
49 #include <memory>
50 #include <memory>
51 #include <memory>
52 #include <memory>
53 #include <memory>
54 #include <memory>
55 #include <memory>
56 #include <memory>
57 #include <memory>
58 #include <memory>
59 #include <memory>
60 #include <memory>
61 #include <memory>
62 #include <memory>
63 #include <memory>
64 #include <memory>
65 #include <memory>
66 #include <memory>
67 #include <memory>
68 #include <memory>
69 #include <memory>
70 #include <memory>
71 #include <memory>
72 #include <memory>
73 #include <memory>
74 #include <memory>
75 #include <memory>
76 #include <memory>
77 #include <memory>
78 #include <memory>
79 #include <memory>
80 #include <memory>
81 #include <memory>
82 #include <memory>
83 #include <memory>
84 #include <memory>
85 #include <memory>
86 #include <memory>
87 #include <memory>
88 #include <memory>
89 #include <memory>
90 #include <memory>
91 #include <memory>
92 #include <memory>
93 #include <memory>
94 #include <memory>
95 #include <memory>
96 #include <memory>
97 #include <memory>
98 #include <memory>
99 #include <memory>
100 #include <memory>

```

[illegible][illegible]

```

def load_file(path, default=None):
    """Load file"""
    return default

def save_file(path, data):
    """Save file"""
    # Create a temporary file then replace the target file.
    # This is to avoid race conditions.
    tmp = path + ".tmp"
    try:
        with open(tmp, "w", encoding="utf-8") as f:
            f.write(data)
            f.flush()
            os.rename(tmp, path)
    except Exception:
        try:
            os.remove(tmp)
        except:
            pass
        raise

def load_file(path):
    """Load file"""
    return load_file(path, default=None)

```

```

# empty list to store
def load_questions_for_package(pkg_name):
    return questions_in []

# ===== normalize =====
def normalize_correct_answer(ans):
    # normalize 'correct_answer' in each question to be an string like this string.
    # accepts string like, single letter answer like 'A', or multi-lettered string
    # like 'ABC'
    if not ans:
        return []
    for i in ans:
        # if i is get('correct_answer')
        # unless i is get('choices') is []
        if isinstance(i, int):
            continue
        if isinstance(i, str):
            # if i is string()
            # single letter word or zero
            if len(i) == 1 and i.isalpha():
                # like 'a' and 'upper(1)' and 'A'
                if i == '0' or i.isupper():
                    # 'correct_answer' + len
                    continue
            # if i is not string is string
    # yes
    # 'correct_answer' + choices.index(x)
    continue
    except Exception:
        # correct = [str(i).strip().lower() for i in i.choices]
        try:
            # 'correct_answer' + choices.index(x).lower()

```

```
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 == 'all' or not q.get('level') or '_form_pkg_name' in q.get('level'))]
    if not pool:
        return []

    today = datetime.date.today().isoformat()
    seed_str = f'{today}:{package or ""}:{level}'
    seed = sha1(hash(seed_str)).hexdigest()
    rng = random.Random(seed)
    selected = rng.sample(pool, k=min(count, len(pool)))
    rng.shuffle(selected)
    return selected
```

[illegible][illegible]

Bloom Study

◀ Soal 1 / 8 ▶ 🕒 74:59

Jumlah 20 suku pertama deret aritmetika dengan $a_1=5$ dan $d=3$ adalah...

590

600

610

620

630

● Next Step

🔄 Kembali

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/>