

## **LAPORAN AKHIR PROYEK**

### ***Aplikasi Latihan Soal UTBK***



Nama Mahasiswa: Sulisa Dermawati Br Sinulingga

NIM: 25031554012

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 .....	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 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 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. 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 personal.



## 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, normalise_correct_answer,
12     pick_questions_with_fresh_priority, pick_daily_challenge_by_level,
13     load_json, save_json, HISTOGRAM_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_RNG = 1
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.storing_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     # ===== UI =====
71     def _ui(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", 18, "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     # ===== TILED =====
101     def _clear(self):
102         # Clear the self.container (only widget)

```

```

103
104     # ===== TILED =====
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                 self.after_cancel(self.timer_job)
129             except Exception:
130                 pass

```

```

131
132     def _home(self):
133         # Messagebox about "both, both", "both, none"
134         messagebox.showinfo("both, both", "both, none")
135         pass
136         self._clear()
137         return
138
139         self.remaining = s
140         try:
141             self.timer_job = self.after(1000, self._tick)
142         except Exception:
143             pass
144         self.timer_job = None
145
146     # ===== TILED =====
147     def _home(self):
148         if self.timer_job:
149             try:
150                 self.after_cancel(self.timer_job)
151             except Exception:
152                 pass
153             self.timer_job = None
154
155         self._clear()
156
157         bg = tk.Frame(self.container)
158         bg.grid(column=0, row=0, sticky="nsew")
159
160         title_label = tk.Label(bg, text="LATIHAN SOAL UTS", style="Title.Label", justify="center")
161         subtitle_label = tk.Label(bg, text="Soal UTS", style="Subtitle.Label", justify="center")
162         btn_label = tk.Label(bg, text="Start", style="Button", justify="center")
163         btn_label.config(command=self._start_timer)
164         btn_label.grid(column=0, row=1, sticky="nsew")
165
166         btn_label = tk.Label(bg, text="Home", style="Button", justify="center")
167         btn_label.config(command=self._home)
168         btn_label.grid(column=0, row=2, sticky="nsew")

```

```

169
170     # ===== TILED =====
171     def _package_menu(self):
172         self._clear()
173         bg = tk.Frame(self.container)
174         bg.grid(column=0, row=0, sticky="nsew")
175
176         title_label = tk.Label(bg, text="PILIH SOAL", style="Title.Label", justify="center")
177         if not self.packages:
178             title_label.config(text="Tidak ada paket tersedia", style="Subtitle.Label", justify="center")
179         for p in self.packages:
180             btn_label = tk.Label(bg, text=p["name"], style="Button", justify="center")
181             btn_label.config(command=lambda p: self._load_menu(p))
182             btn_label.grid(column=0, row=1, sticky="nsew")
183
184         btn_label = tk.Label(bg, text="Kembali", style="Button", justify="center")
185         btn_label.config(command=self._home)
186         btn_label.grid(column=0, row=2, sticky="nsew")
187
188     def _load_menu(self, package):
189         self._clear()
190         bg = tk.Frame(self.container)
191         bg.grid(column=0, row=0, sticky="nsew")
192
193         title_label = tk.Label(bg, text="PILIH SOAL", style="Title.Label", justify="center")
194         for q in package["questions"]:
195             btn_label = tk.Label(bg, text=q["text"], style="Button", justify="center")
196             btn_label.config(command=lambda q: self._start_quiz(package, q))
197             btn_label.grid(column=0, row=1, sticky="nsew")
198
199         btn_label = tk.Label(bg, text="Kembali", style="Button", justify="center")
200         btn_label.config(command=self._package_menu)
201         btn_label.grid(column=0, row=2, sticky="nsew")
202
203     # ===== TILED =====
204     def _start_quiz(self, package, level):
205         self._clear()
206         bg = tk.Frame(self.container)
207         bg.grid(column=0, row=0, sticky="nsew")
208
209         title_label = tk.Label(bg, text="LATIHAN SOAL UTS", style="Title.Label", justify="center")
210         subtitle_label = tk.Label(bg, text="Soal UTS", style="Subtitle.Label", justify="center")
211         btn_label = tk.Label(bg, text="Start", style="Button", justify="center")
212         btn_label.config(command=self._start_timer)
213         btn_label.grid(column=0, row=1, sticky="nsew")

```

[illegible][illegible][illegible]

```

104 # -# merge > 70 (maybe 50) ms delay
105 # -# (source: https://doi.org/10.1016/j.jneurosci.2014.04.034)
106 def wait_until(self):
107
108     self.wait_until_ready()
109     self.waiting_explanation = None
110     return
111
112 if self.idle < len(self.questions) + 1:
113     self.idle += 1
114     self.wait_question()
115 else:
116     self.finish()
117
118 # ======
119 def finish(self):
120     # record time, get of waiting
121     self.time_job
122     try:
123         self.time_job(self.time_job)
124     except Exception:
125         pass
126     self.time_job = None
127
128 self.idle = 1 + get("10") for a in self.questions in [] if a.get("10")
129
130 final_time = self.time_job if self.idle else self.time
131
132 results = {}
133
134 self.time_job = None
135
136 self.time_job = None
137
138 self.time_job = None
139
140 self.time_job = None
141
142 self.time_job = None
143
144 self.time_job = None
145
146 self.time_job = None
147
148 self.time_job = None
149
150 self.time_job = None
151
152 self.time_job = None
153
154 self.time_job = None
155
156 self.time_job = None
157
158 self.time_job = None
159
160 self.time_job = None
161
162 self.time_job = None
163
164 self.time_job = None
165
166 self.time_job = None
167
168 self.time_job = None
169
170 self.time_job = None
171
172 self.time_job = None
173
174 self.time_job = None
175
176 self.time_job = None
177
178 self.time_job = None
179
180 self.time_job = None
181
182 self.time_job = None
183
184 self.time_job = None
185
186 self.time_job = None
187
188 self.time_job = None
189
190 self.time_job = None
191
192 self.time_job = None
193
194 self.time_job = None
195
196 self.time_job = None
197
198 self.time_job = None
199
200 self.time_job = None
201
202 self.time_job = None
203
204 self.time_job = None
205
206 self.time_job = None
207
208 self.time_job = None
209
210 self.time_job = None
211
212 self.time_job = None
213
214 self.time_job = None
215
216 self.time_job = None
217
218 self.time_job = None
219
220 self.time_job = None
221
222 self.time_job = None
223
224 self.time_job = None
225
226 self.time_job = None
227
228 self.time_job = None
229
230 self.time_job = None
231
232 self.time_job = None
233
234 self.time_job = None
235
236 self.time_job = None
237
238 self.time_job = None
239
240 self.time_job = None
241
242 self.time_job = None
243
244 self.time_job = None
245
246 self.time_job = None
247
248 self.time_job = None
249
250 self.time_job = None
251
252 self.time_job = None
253
254 self.time_job = None
255
256 self.time_job = None
257
258 self.time_job = None
259
260 self.time_job = None
261
262 self.time_job = None
263
264 self.time_job = None
265
266 self.time_job = None
267
268 self.time_job = None
269
270 self.time_job = None
271
272 self.time_job = None
273
274 self.time_job = None
275
276 self.time_job = None
277
278 self.time_job = None
279
280 self.time_job = None
281
282 self.time_job = None
283
284 self.time_job = None
285
286 self.time_job = None
287
288 self.time_job = None
289
290 self.time_job = None
291
292 self.time_job = None
293
294 self.time_job = None
295
296 self.time_job = None
297
298 self.time_job = None
299
300 self.time_job = None
301
302 self.time_job = None
303
304 self.time_job = None
305
306 self.time_job = None
307
308 self.time_job = None
309
310 self.time_job = None
311
312 self.time_job = None
313
314 self.time_job = None
315
316 self.time_job = None
317
318 self.time_job = None
319
320 self.time_job = None
321
322 self.time_job = None
323
324 self.time_job = None
325
326 self.time_job = None
327
328 self.time_job = None
329
330 self.time_job = None
331
332 self.time_job = None
333
334 self.time_job = None
335
336 self.time_job = None
337
338 self.time_job = None
339
340 self.time_job = None
341
342 self.time_job = None
343
344 self.time_job = None
345
346 self.time_job = None
347
348 self.time_job = None
349
350 self.time_job = None
351
352 self.time_job = None
353
354 self.time_job = None
355
356 self.time_job = None
357
358 self.time_job = None
359
360 self.time_job = None
361
362 self.time_job = None
363
364 self.time_job = None
365
366 self.time_job = None
367
368 self.time_job = None
369
370 self.time_job = None
371
372 self.time_job = None
373
374 self.time_job = None
375
376 self.time_job = None
377
378 self.time_job = None
379
380 self.time_job = None
381
382 self.time_job = None
383
384 self.time_job = None
385
386 self.time_job = None
387
388 self.time_job = None
389
390 self.time_job = None
391
392 self.time_job = None
393
394 self.time_job = None
395
396 self.time_job = None
397
398 self.time_job = None
399
400 self.time_job = None
401
402 self.time_job = None
403
404 self.time_job = None
405
406 self.time_job = None
407
408 self.time_job = None
409
410 self.time_job = None
411
412 self.time_job = None
413
414 self.time_job = None
415
416 self.time_job = None
417
418 self.time_job = None
419
420 self.time_job = None
421
422 self.time_job = None
423
424 self.time_job = None
425
426 self.time_job = None
427
428 self.time_job = None
429
430 self.time_job = None
431
432 self.time_job = None
433
434 self.time_job = None
435
436 self.time_job = None
437
438 self.time_job = None
439
440 self.time_job = None
441
442 self.time_job = None
443
444 self.time_job = None
445
446 self.time_job = None
447
448 self.time_job = None
449
450 self.time_job = None
451
452 self.time_job = None
453
454 self.time_job = None
455
456 self.time_job = None
457
458 self.time_job = None
459
460 self.time_job = None
461
462 self.time_job = None
463
464 self.time_job = None
465
466 self.time_job = None
467
468 self.time_job = None
469
470 self.time_job = None
471
472 self.time_job = None
473
474 self.time_job = None
475
476 self.time_job = None
477
478 self.time_job = None
479
480 self.time_job = None
481
482 self.time_job = None
483
484 self.time_job = None
485
486 self.time_job = None
487
488 self.time_job = None
489
490 self.time_job = None
491
492 self.time_job = None
493
494 self.time_job = None
495
496 self.time_job = None
497
498 self.time_job = None
499
500 self.time_job = None
501
502 self.time_job = None
503
504 self.time_job = None
505
506 self.time_job = None
507
508 self.time_job = None
509
510 self.time_job = None
511
512 self.time_job = None
513
514 self.time_job = None
515
516 self.time_job = None
517
518 self.time_job = None
519
520 self.time_job = None
521
522 self.time_job = None
523
524 self.time_job = None
525
526 self.time_job = None
527
528 self.time_job = None
529
530 self.time_job = None
531
532 self.time_job = None
533
534 self.time_job = None
535
536 self.time_job = None
537
538 self.time_job = None
539
540 self.time_job = None
541
542 self.time_job = None
543
544 self.time_job = None
545
546 self.time_job = None
547
548 self.time_job = None
549
550 self.time_job = None
551
552 self.time_job = None
553
554 self.time_job = None
555
556 self.time_job = None
557
558 self.time_job = None
559
560 self.time_job = None
561
562 self.time_job = None
563
564 self.time_job = None
565
566 self.time_job = None
567
568 self.time_job = None
569
570 self.time_job = None
571
572 self.time_job = None
573
574 self.time_job = None
575
576 self.time_job = None
577
578 self.time_job = None
579
580 self.time_job = None
581
582 self.time_job = None
583
584 self.time_job = None
585
586 self.time_job = None
587
588 self.time_job = None
589
590 self.time_job = None
591
592 self.time_job = None
593
594 self.time_job = None
595
596 self.time_job = None
597
598 self.time_job = None
599
600 self.time_job = None
601
602 self.time_job = None
603
604 self.time_job = None
605
606 self.time_job = None
607
608 self.time_job = None
609
610 self.time_job = None
611
612 self.time_job = None
613
614 self.time_job = None
615
616 self.time_job = None
617
618 self.time_job = None
619
620 self.time_job = None
621
622 self.time_job = None
623
624 self.time_job = None
625
626 self.time_job = None
627
628 self.time_job = None
629
630 self.time_job = None
631
632 self.time_job = None
633
634 self.time_job = None
635
636 self.time_job = None
637
638 self.time_job = None
639
640 self.time_job = None
641
642 self.time_job = None
643
644 self.time_job = None
645
646 self.time_job = None
647
648 self.time_job = None
649
650 self.time_job = None
651
652 self.time_job = None
653
654 self.time_job = None
655
656 self.time_job = None
657
658 self.time_job = None
659
660 self.time_job = None
661
662 self.time_job = None
663
664 self.time_job = None
665
666 self.time_job = None
667
668 self.time_job = None
669
670 self.time_job = None
671
672 self.time_job = None
673
674 self.time_job = None
675
676 self.time_job = None
677
678 self.time_job = None
679
680 self.time_job = None
681
682 self.time_job = None
683
684 self.time_job = None
685
686 self.time_job = None
687
688 self.time_job = None
689
690 self.time_job = None
691
692 self.time_job = None
693
694 self.time_job = None
695
696 self.time_job = None
697
698 self.time_job = None
699
700 self.time_job = None
701
702 self.time_job = None
703
704 self.time_job = None
705
706 self.time_job = None
707
708 self.time_job = None
709
710 self.time_job = None
711
712 self.time_job = None
713
714 self.time_job = None
71
```

```

def test():
    """Test the function"""
    # Create a list of numbers
    numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

    # Call the function
    result = my_function(numbers)

    # Print the result
    print(result)

# Run the test
test()

```

[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.fsync(f.fileno())
        os.replace(tmp, path)
    except Exception:
        try:
            os.unlink(tmp)
        except:
            pass
    return data

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

```

[illegible]

```
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]

```

"lat": "50-00",
"longitude": "0",
"target": "pendulum arm",
"task": "swing",
"question": "Sebuah ayunan 1, 0, 1, 0, 1, 0, ayun selanjutnya adalah:",
"choices": "100", "00", "11", "100", "001",
"correct_answer": "10",
"explanation": "Jika 01, berarti 10 = 0."

"lat": "50-00",
"longitude": "0",
"target": "pendulum arm",
"task": "swing",
"question": "Jika anda melek titik tengah, dan beberapa menit kita terbang, anda:",
"choices": "Tawa dengan kita terbang", "Merasa benar kita terbang", "Tawa dengan kita terbang", "Berasa benar kita terbang",
"correct_answer": "Berasa benar kita terbang",
"explanation": "Jika: anda benar terbang, beberapa menit terbang = benar: merasa benar kita terbang."

"lat": "50-00",
"longitude": "0",
"target": "pendulum arm",
"task": "swing",
"question": "Sebuah huruf A, B, C, D, E, huruf selanjutnya adalah:",
"choices": "100", "00", "00", "00", "00",
"correct_answer": "0",
"explanation": "Jika: Jawab: E huruf selanjutnya = 0."

```

## 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/>