

Logbook Penggunaan AI untuk Lomba Festika 2025

Dibuat oleh kelompok “**Import os**” dari **SMK Telkom Malang** yang beranggotakan :

1. Wiryateja Pamungkas - X RPL - (+62) 818-0306-1922
2. Maven Helios Agathon Yesstian - X RPL - (+62) 812-3130-1294

Tabel Log Book penggunaan AI

Tanggal dan Waktu	AI yang digunakan	Prompt / Pertanyaan	Respon AI (Ringkasan / code)	Implementasi & Refleksi team
16 November 2025, 18:23	ChatGPT (5.1)	Saya ingin membuat sebuah program python pencari file yang berjalan di terminal dan dibuat hanya menggunakan library bawaan python. Pada program tersebut user dapat memilih untuk memasukan Judul(Keyword) dari file atau menginputkan keyword di dalam file atau memilih menggunakan keduanya. dan jangan lupa untuk menjelaskan tiap code yang kamu buat.	AI memberikan kami sebuah program pencari file yang sudah memiliki fitur fitur yang kuminta, lalu ia memberikan kami cara code itu bekerja mulai dari awal sampai akhir. AI juga memberikan rekomendasi terhadap saya apakah ingin mengimplementasikan fitur tambahan seperti preview file yang dicari, buka file/folder yang dicari, dan semacamnya. potongan kode : import os import difflib def clear(): os.system('cls' if os.name == 'nt' else 'clear') def search_by_name(keyword, root): results = [] for dirpath, _, filenames in os.walk(root): for f in filenames: full_path = os.path.join(dirpath, f) results.append(full_path) # fuzzy match untuk mencocokkan nama file matches = difflib.get_close_matches(keyword, results, n=50, cutoff=0.1)	Implementasi: Kami membaca dan mencermati code serta penjelasan yang diberikan oleh AI terlebih dahulu, setelah melakukan analisa kami memutuskan untuk menerapkan code tersebut di dalam program kami tepatnya pada fitur “File Finder”. Refleksi : Kami belajar banyak hal melalui program yang dibuat oleh AI. Kami menemukan bahwa untuk membuat sebuah program pencari file, kita tidak perlu menggunakan library dari luar dan bisa hanya dengan menggunakan library bawaan saja. Library utama yang digunakan adalah os, pathlib, dan pickle. Kami belajar banyak tentang fungsi dan cara kerja tiap impor yang digunakan

			<pre> return matches def search_by_content(keyword, root): found = [] for dirpath, _, filenames in os.walk(root): for f in filenames: full_path = os.path.join(dirpath, f) # coba buka sebagai file teks try: with open(full_path, "r", errors="ignore") as file: content = file.read() if keyword.lower() in content.lower(): found.append(full_path) except: # file binary atau rusak → skip continue return found </pre>	
17 November 2025, 08:41	ChatGPT (5.1)	Setelah saya melakukan uji coba pada program-mu (Program file finder) saya menemukan bahwa program milik kamu berjalan sangat lambat dan tidak efisien sehingga user akan bosan menunggu. Sekarang perbaiki permasalahan ini dan jelaskan bagaimana cara kamu memperbaikinya	AI memberikan kita berbagai opsi yang dapat kita pilih untuk membuat program ini berjalan lebih cepat dan efisien, fitur yang direkomendasikan AI adalah menggunakan “Cache”, menggunakan “Threading” dan juga dengan cara membaca sekilas file untuk menentukan apakah file tersebut layak dibaca (Lanjut ke program) atau tidak (skip). AI juga memberikan kami sebuah fitur load bar agar user tidak merasa “bosan” atau menganggap program error. (Code terlalu panjang untuk diletakan disini)	<p>Implementasi : Kami memutuskan untuk memahami terlebih dahulu bagaimana “fitur” ini bekerja, setelah memahami lebih lanjut kita membuat mengimplementasikan ini pada program kita dan hasilnya adalah program kami berjalan lebih cepat.</p> <p>Refleksi : Pada awalnya program pencari file ini berjalan lambat karena harus mengecek file satu per satu. Tetapi dengan fungsi ini Kami mampu untuk mengurangi waktu yang dibutuhkan dalam pencarian file. Kami juga belajar berbagai ilmu baru dengan fungsi fungsi ini</p>
18 November 2025, 16:52	ChatGPT (5.1)	Sekarang aku ingin kamu membuatkan aku fitur tambahan yang kamu bilang padaku sebelumnya	AI memberikan kita sebuah fungsi dimana user dapat melakukan preview file dan menunjukkan baris mana dari file yang memiliki keyword yang kuminta dengan menggunakan library bawaan python berupa “re” dan cara membuka folder/file	<p>Implementasi : Kami mengimplementasikan fungsi yang diberikan oleh chatGPT ke dalam program kami.</p> <p>Refleksi : Kami mempelajari beberapa cara “memanipulasi”</p>

		<p>(Preview file dan open folder/file). Seperti biasa aku ingin kamu membuatkan program sambil menjelaskan bagaimana program itu bekerja</p> <pre> menggunakan "open()" potongan code : import os import chardet import subprocess def safe_read_text(path): """Membaca teks dengan encoding detection. Mengembalikan None jika biner / tidak bisa dibaca.""" try: with open(path, "rb") as f: raw = f.read() if b"\x00" in raw: # indikasi file biner return None enc = chardet.detect(raw)["encoding"] or "utf-8" return raw.decode(enc, errors="replace") except: return None def preview_context(text, keyword, context_lines=1): """Mengambil 3 baris konteks: sebelum, baris keyword, sesudah.""" lines = text.splitlines() results = [] for i, line in enumerate(lines): if keyword.lower() in line.lower(): start = max(0, i - context_lines) end = min(len(lines), i + context_lines + 1) snippet = lines[start:end] results.append((i + 1, snippet)) # simpan nomor baris & snippet return results def open_path(path): """Open file/folder cross-platform.""" if os.name == "nt": os.startfile(path) elif "darwin" in os.sys.platform: subprocess.Popen(["open", path]) else: subprocess.Popen(["xdg-open", path]) </pre>	<p>folder menggunakan script python. Hasilnya kami mampu membuat program kami lebih efektif dan tepat sasaran terhadap tujuan dibuatnya program kami.</p>
--	--	---	---

```

def search_by_content(root,
keyword):
    results = []
    all_files = []

    # Kumpulkan file terlebih dahulu
    # (untuk progress)
    for dirpath, _, filenames in
os.walk(root):
        for f in filenames:

            all_files.append(os.path.join(dirpath,
f))

    total = len(all_files)
    print(f"Total files: {total}")

    scanned = 0

    for path in all_files:
        scanned += 1
        print(f"\rScanning {scanned} out
of {total}", end="")

        text = safe_read_text(path)
        if not text:
            continue

        context_hits =
preview_context(text, keyword)
        if context_hits:
            results.append((path,
context_hits))

    print("\n\nScan finished.\n")
    return results

def show_results(results):
    for idx, (path, contexts) in
enumerate(results, 1):
        print(f"[{idx}] {path}")

        # tampilkan hanya 1 snippet
        pertama per file
        line_num, snippet = contexts[0]
        print(f" Line {line_num}:")
        for line in snippet:
            print(f"  {line}")
        print()

    if not results:
        print("No results found.")
    return

while True:
    choice = input("Select result

```

			<pre> number to open (or press Enter to skip): ").strip() if not choice: break if not choice.isdigit() or not (1 <= int(choice) <= len(results)): print("Invalid choice.") continue _, (file_path, _) = list(enumerate(results, 1))[int(choice) - 1] print("\n1. Open this file") print("2. Open folder") print("3. Back") action = input("Choose: ") if action == "1": open_path(file_path) elif action == "2": open_path(os.path.dirname(file_path)) else: continue </pre>	
17 November 2025, 20:12	ChatGPT (5.1)	cara membuat code untuk membuat program file organizer menggunakan python (CLI) menggunakan standard library saja	<p>AI memberikan tujuan dibuatnya program file organizer, lalu memberikan code beserta penjelasan dan langkah-langkahnya. Penjelasan tersebut memuat:</p> <ul style="list-style-type: none"> - Struktur dasar program, yakni fitur-fitur yang diperlukan untuk membuat program File Organizer -Cara menjalankan program -Hasil yang didapat setelah menjalankan program <p>Potongan kode :</p> <pre> import os import shutil import sys </pre> <pre> FILE_CATEGORIES = { "Images": [".jpg", ".jpeg", ".png", ".gif", ".bmp", ".svg"], "Documents": [".pdf", ".docx", ".doc", ".txt", ".xlsx", ".pptx"], "Videos": [".mp4", ".mkv", ".mov", ".avi"], "Audio": [".mp3", ".wav", ".flac"], "Archives": [".zip", ".rar", ".tar", ".gz"], "Code": [".py", ".js", ".html", ".css", ".php", ".cpp", ".java"] } </pre>	<p>Implementasi : Memahami bagaimana code-nya bekerja lalu mengimplementasikannya pada program kami. Setelah itu, kami memberikan penjelasan ke beberapa bagian yang dipahami dan menambah atau modifikasi beberapa fitur.</p> <p>Refleksi : Library utama yang digunakan untuk membuat program ini adalah os dan shutil untuk mendapatkan nama path file dan folder serta memindahkan file ke folder yang sesuai dengan kelompoknya. Sebelumnya kami kurang paham mengenai fungsi os dan shutil, setelah diberikan penjelasan, saya mulai paham kalau os dan shutil digunakan untuk memeriksa dan mengatur path file atau folder dalam komputer.</p>

```

def get_category(file_name):
    ext =
os.path.splitext(file_name)[1].lower()
    for category, extensions in
FILE_CATEGORIES.items():
        if ext in extensions:
            return category
    return "Others" # default jika
ekstensi tidak dikenali

def organize(folder_path):
    if not os.path.isdir(folder_path):
        print("Error: Path tidak valid.")
        return

    for file_name in
os.listdir(folder_path):
        file_path =
os.path.join(folder_path, file_name)

        # Lewati folder
        if os.path.isdir(file_path):
            continue

        category =
get_category(file_name)
        category_folder =
os.path.join(folder_path, category)

        os.makedirs(category_folder,
exist_ok=True)

        shutil.move(file_path,
os.path.join(category_folder,
file_name))
        print(f"[OK] {file_name} ->
{category}")

def main():
    if len(sys.argv) < 2:
        print("Usage: python
organizer.py <folder_path>")
        sys.exit(1)

    target_folder = sys.argv[1]
    organize(target_folder)

if __name__ == "__main__":
    main()

```

18 November 2025, 14:42	ChatGPT (5.1)	<p>berikan saya salinan semua format file dari gambar hingga code, misal:</p> <pre>["format", ".format", ".format"]</pre> <p>AI memberikan salinan nama-nama format/ekstensi file umum dengan kategorinya (ada 13 kategori), misalnya .jpg dan .png termasuk kategori gambar. Kategorinya adalah gambar, video, audio, dokumen, arsip, executable, code, database, font, E-Book, 3D Model, dan GIS/Map data. Jika suatu file ekstensinya tidak terdapat dalam 12 kategori tersebut, maka file akan masuk kategori "Lainnya".</p> <p>Potongan code :</p> <pre>FILE_CATEGORIES = { "Gambar": [".jpg", ".jpeg", ".png", ".gif", ".bmp", ".tiff", ".tif", ".svg", ".webp", ".ico", ".heic", ".heif", ".raw", ".arw", ".cr2", ".nef", ".orf", ".rw2", ".psd", ".ai", ".eps"], "Dokumen": [".pdf", ".doc", ".docx", ".txt", ".rtf", ".odt", ".xls", ".xlsx", ".csv", ".tsv", ".ppt", ".pptx", ".epub", ".md"], "Video": [".mp4", ".mkv", ".avi", ".mov", ".wmv", ".flv", ".webm", ".mpeg", ".mpg", ".3gp", ".m4v"], "Suara": [".mp3", ".wav", ".flac", ".aac", ".ogg", ".oga", ".wma", ".m4a", ".amr", ".aiff"], "Arsip": [".zip", ".rar", ".7z", ".tar", ".gz", ".bz2", ".xz", ".iso", ".lz", ".zst"], "Kode": [".py", ".js", ".ts", ".html", ".css", ".php", ".java", ".cpp", ".c", ".h", ".hpp", ".cs", ".rb", ".go", ".rs", ".kt", ".swift", ".m", ".lua", ".sql", ".xml", ".json", ".yaml", ".yml"] }</pre>	<p>Implementasi : Copy dan Paste salinan yang diberikan untuk menambah data atau nilai pada konstan/list kategori file.</p> <p>Refleksi : Ekstensi file hasil dari prompt AI tersebut sebenarnya sudah lengkap untuk ekstensi umum, hanya saja ada ekstensi file lain yang bersifat kustom (seperti file yang di-save dari sebuah software umum/tidak umum biasanya menggunakan ekstensi kustom) dan ekstensi file korup yang sifatnya tidak umum.</p>
----------------------------------	------------------	---	--

```
".yaml", ".yml"
],
"3DModel": [
    ".obj", ".fbx", ".stl", ".dae",
    ".blend",
    ".gltf", ".glb"
],
"Aplikasi-Executable-Installer": [
    ".exe", ".msi", ".bat", ".cmd",
    ".sh", ".apk",
    ".app", ".deb", ".rpm"
],
"Database": [
    ".db", ".sqlite", ".sqlite3", ".mdb",
    ".accdb", ".sql", ".dbf"
],
"GIS-MapData": [
    ".shp", ".kml", ".kmz", ".geojson",
    ".gpx"
],
"Font": [
    ".ttf", ".otf", ".woff", ".woff2"
],
"EBook": [
    ".epub", ".mobi", ".azw3", ".fb2"
]
}
```