E-ISSN : 2988-1986
https://ejournal.cahayailmubangsa.institute/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 10 No 7 Tahun 2025

IMPLEMENTASI FITUR CHECKLIST BERBASIS GUI UNTUK MEMASTIKAN KELENGKAPAN BARANG SEBELUM BERAKTIVITAS PADA APLIKASI SMART REMINDER

Derisya Febriyanti[1*], Ahmad Tabrani[2]
[1,2] Teknik Informatika, Universitas Islam Negeri Sultan Maulana Hasanuddin Banten
E-mail: deriaya491@gmail.com[1], ahmad.tabrani@uinbanten.ac.id[2]

ABSTRACT

A simple and user-friendly reminder application is needed to help users ensure their belongings are complete before carrying out daily activities. This study implements a GUI-based checklist feature in the Smart Reminder application. The feature allows users to create an item list, mark prepared items, automatically view completion status, reset the checklist, and store activity logs. The system was developed using the Python programming language with Tkinter as the main graphical user interface framework. The results show that the checklist feature helps users organize their preparation process, reduce forgotten items, and provide clear visual feedback regarding item completeness. This work is expected to serve as a foundation for future enhancements, including intelligent notifications and integration with daily reminder features.

Keywords: checklist, GUI, Smart Reminder, item completeness, Python.

ABSTRAK

Aplikasi pengingat yang sederhana namun mudah digunakan dibutuhkan untuk membantu pengguna memastikan kelengkapan barang sebelum beraktivitas. Penelitian ini mengimplementasikan fitur checklist berbasis Graphical User Interface (GUI) pada aplikasi Smart Reminder. Fitur ini memungkinkan pengguna membuat daftar barang, memberi tanda pada item yang sudah dipersiapkan, melihat status kelengkapan secara otomatis, melakukan reset checklist, serta menyimpan riwayat aktivitas. Pengembangan dilakukan menggunakan bahasa pemrograman Python dengan pustaka Tkinter sebagai antarmuka utama. Hasil implementasi menunjukkan bahwa fitur checklist mampu membantu pengguna memantau persiapan secara lebih terstruktur, meminimalkan kelupaan, dan memberikan umpan balik visual yang jelas mengenai tingkat kelengkapan barang. Penelitian ini diharapkan menjadi dasar untuk pengembangan fitur lanjutan seperti notifikasi cerdas dan integrasi dengan pengingat aktivitas harian.

Kata kunci: checklist, GUI, Smart Reminder, kelengkapan barang, Python.

Article History
Received: Desember 2025
Reviewed:Desember 2025
Published: Desember 2025

Plagiarism Checker No 235
Prefix DOI : 10.8734/Kohesi.v1i2.365
Copyright : Author
Publish by : Kohesi

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

1. PENDAHULUAN

Aktivitas sehari-hari sering menuntut seseorang untuk membawa berbagai barang penting, seperti telepon seluler, dompet, kartu identitas, laptop, dan perlengkapan kerja lainnya. Namun dalam praktiknya, sebagian orang masih sering lupa menyiapkan beberapa barang, terutama ketika berada dalam situasi terburu-buru. Kelalaian tersebut tidak hanya menimbulkan ketidaknyamanan, tetapi juga dapat berdampak pada terganggunya produktivitas dan efektivitas kegiatan yang dijalankan. Oleh karena itu, diperlukan solusi sederhana yang mampu membantu pengguna mengingat dan memeriksa kembali kelengkapan barang sebelum beraktivitas.

Perkembangan teknologi informasi, khususnya pada perangkat komputer dan aplikasi pendukung, membuka peluang untuk menghadirkan sistem pengingat yang mudah digunakan. Salah satu bentuk solusi yang relevan adalah aplikasi berbasis Graphical User Interface (GUI) yang memungkinkan pengguna berinteraksi secara visual melalui tombol, daftar, dan indikator status. Aplikasi dengan pendekatan GUI umumnya lebih intuitif, sehingga dapat digunakan oleh berbagai kalangan tanpa memerlukan pengetahuan teknis yang rumit.

Penelitian ini berfokus pada implementasi fitur checklist pada aplikasi Smart Reminder. Fitur checklist dirancang untuk membantu pengguna mencentang item yang telah dipersiapkan, memantau status kelengkapan secara otomatis, serta melakukan pengaturan ulang (reset) apabila diperlukan. Selain itu, aplikasi juga mencatat riwayat aktivitas sehingga perubahan yang dilakukan pengguna dapat terdokumentasi dengan baik. Pendekatan ini diharapkan dapat membentuk kebiasaan persiapan yang lebih terstruktur dan konsisten.

Pengembangan aplikasi dilakukan menggunakan bahasa pemrograman Python dengan pustaka Tkinter sebagai antarmuka utama. Pemilihan Python didasarkan pada kemudahan implementasi, fleksibilitas pengembangan, serta dukungan komunitas yang luas. Dengan memanfaatkan fitur-fitur tersebut, aplikasi diharapkan mampu memberikan pengalaman penggunaan yang sederhana namun fungsional.

Secara umum, tujuan penelitian ini adalah: (1) merancang dan mengimplementasikan fitur checklist berbasis GUI pada aplikasi Smart Reminder; (2) menguji fungsi checklist dalam membantu pengguna memastikan kelengkapan barang; dan (3) memberikan gambaran awal

mengenai potensi pengembangan sistem pengingat yang lebih cerdas pada tahap selanjutnya. Hasil penelitian ini diharapkan dapat menjadi referensi dalam pengembangan aplikasi pengingat sederhana yang mendukung aktivitas harian secara praktis dan efektif.

Cuplikan Kode Utama (Logika Reminder) berikut menunjukkan alur dasar bagaimana sistem checklist bekerja: menyimpan daftar barang, menandai item yang sudah dicek, memperbarui status, dan mencatat riwayat perubahan.

```
import datetime

class ReminderChecklist:
    def __init__(self):
        # daftar barang yang akan dicek
        self.items = ["HP", "dompet", "tas", "laptop"]
        # status checklist disimpan dalam bentuk dictionary
        self.checked = {item: False for item in self.items}
        # log aktivitas
        self.logs = []

    def toggle_item(self, item):
        """Mengubah status checklist dan menyimpan riwayat."""
        self.checked[item] = not self.checked[item]
        stamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        self.logs.append(f"[{stamp}] {item} -> {self.checked[item]}")

    def status(self):
        """Menghitung berapa item yang sudah dicek."""
        done = sum(1 for i in self.items if self.checked[i])
        total = len(self.items)
        return f"{done}/{total} item sudah dip. apkan"

    def reset(self):
        """Mengembalikan checklist ke kondisi awal."""
        for item in self.items:
            self.checked[item] = False
        self.logs.append("Checklist di-reset")

# contoh penggunaan
rc = ReminderChecklist()
rc.toggle_item("HP")
rc.toggle_item("tas")
print(rc.status())
```

Cuplikan kode di atas menggambarkan beberapa konsep inti:

1. **Representasi item**
   Barang yang perlu dipersiapkan disimpan dalam daftar agar mudah diproses.
2. **Status checklist**
   Setiap item memiliki nilai True/False untuk menunjukkan sudah dicek atau belum.
3. **Pencatatan log**
   Setiap perubahan dicatat bersama waktu kejadian, sehingga riwayat aktivitas dapat ditelusuri.
4. **Reset checklist**
   Pengguna dapat mengembalikan seluruh item ke kondisi awal ketika akan melakukan persiapan ulang.

Struktur logika ini kemudian diintegrasikan dengan antarmuka GUI (Tkinter) sehingga pengguna dapat berinteraksi melalui tombol, checkbox, dan indikator status secara visual.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan fokus pada perancangan dan implementasi fitur checklist berbasis GUI pada aplikasi Smart Reminder. Pendekatan ini dipilih karena tujuan penelitian tidak hanya menghasilkan konsep, tetapi juga prototipe aplikasi yang dapat digunakan secara langsung oleh pengguna.

Alur penelitian terdiri atas beberapa tahap utama:

- analisis kebutuhan,
- perancangan sistem,
- implementasi,
- pengujian fungsional, dan
- evaluasi sederhana terhadap kemudahan penggunaan.

Alat dan Bahan Penelitian

Pengembangan sistem dilakukan menggunakan perangkat dan pustaka berikut:

- Bahasa pemrograman : Python
- Framework GUI : Tkinter
- Editor pengembangan : Visual Studio Code
- Sistem operasi : Windows

- Library pendukung : modul standar Python (datetime, messagebox, dan lain-lain)

Data yang digunakan berupa daftar barang yang umumnya dibawa pengguna dalam aktivitas harian. Daftar dapat ditambah, dihapus, dan diubah sesuai kebutuhan pengguna.

Arsitektur Sistem

Secara umum, sistem terdiri dari beberapa komponen:

1. **State Manager**
   Menyimpan daftar barang, status checklist, serta log aktivitas.
2. **Checklist Module**
   Menampilkan daftar item dalam bentuk checkbox, menerima input dari pengguna, dan memperbarui status.
3. **Reminder Logic**
   Menghitung kelengkapan barang, memberi informasi status (lengkap atau belum), serta menyediakan fitur reset.
4. **Log Module**
   Menyimpan riwayat perubahan agar aktivitas dapat ditelusuri kembali.

Komponen tersebut saling berinteraksi, sehingga setiap perubahan pada checklist langsung tercermin pada status dan log sistem.

Alur Kerja Sistem

Secara ringkas, alur kerja aplikasi adalah sebagai berikut:

1. Pengguna membuka aplikasi Smart Reminder.
2. Sistem menampilkan daftar barang yang telah tersimpan.
3. Pengguna memberi tanda (centang) pada item yang sudah dipersiapkan.
4. Sistem memperbarui status kelengkapan secara otomatis.
5. Setiap perubahan disimpan ke dalam log aktivitas.
6. Jika diperlukan, pengguna dapat melakukan reset untuk memulai persiapan baru.

Alur ini memastikan pengguna selalu mendapatkan umpan balik visual tentang tingkat kelengkapan barang.

Prosedur Implementasi

Tahapan implementasi dilakukan secara bertahap:

1. **Perancangan antarmuka**

- Menentukan tata letak halaman checklist.
- Menyusun struktur tombol, label, dan checkbox.

2. **Perancangan logika checklist**
   - Membuat struktur data untuk daftar item.
   - Menentukan mekanisme penandaan dan penyimpanan status.
3. **Integrasi log dan status**
   - Menambahkan pencatatan waktu pada setiap perubahan.
   - Menampilkan informasi status pada halaman utama.
4. **Integrasi penyimpanan sederhana**
   - Menyimpan perubahan agar tetap tersimpan saat aplikasi ditutup.

Pengujian Sistem

Pengujian dilakukan menggunakan uji fungsional (black-box). Setiap fitur diuji berdasarkan keluaran yang dihasilkan, tanpa melihat kode internal.

Fitur yang diuji meliputi:

- Menambah dan menampilkan daftar item
- Memberi tanda pada checklist
- Menampilkan status kelengkapan
- Melakukan reset checklist
- Menyimpan dan menampilkan log aktivitas

Setiap pengujian dinyatakan berhasil apabila sistem menghasilkan keluaran sesuai dengan skenario yang telah ditentukan.

Instrumen Penilaian

Instrumen evaluasi sederhana digunakan untuk melihat:

1. **Kesesuaian fungsi**
   Apakah sistem menjalankan fitur sesuai rancangan.
2. **Kemudahan penggunaan**
   Pengguna memberikan penilaian terhadap kejelasan tampilan, kemudahan interaksi, dan alur aplikasi.
3. **Keandalan checklist**

Apakah fitur membantu pengguna mengurangi kelupaan barang ketika beraktivitas.

Hasil evaluasi digunakan sebagai dasar untuk memberikan rekomendasi pengembangan pada tahap selanjutnya.

3. HASIL DAN PEMBAHASAN

1. Hasil Implementasi Sistem

Implementasi fitur checklist pada aplikasi Smart Reminder menghasilkan antarmuka yang sederhana dan mudah dipahami. Pada halaman checklist, pengguna dapat:

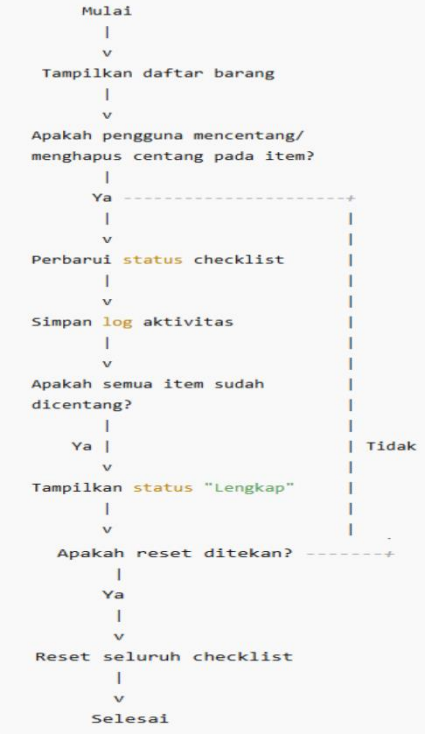
1. melihat daftar barang,
2. memberikan tanda centang pada item yang sudah dipersiapkan,
3. memantau status kelengkapan,
4. melakukan reset checklist,
5. melihat riwayat perubahan yang tersimpan di log.

Setiap perubahan langsung disimpan sehingga ketika aplikasi dibuka kembali, status checklist tetap sama seperti terakhir kali digunakan. Hal ini membuat proses persiapan lebih terstruktur karena pengguna tidak perlu mengingat kembali barang apa saja yang sudah dicek.

Fitur status juga bekerja secara otomatis. Ketika seluruh item telah dicentang, sistem menampilkan informasi “Status: Lengkap”. Sebaliknya, bila masih ada item yang belum dipersiapkan, sistem menampilkan jumlah checklist yang sudah dicentang dibandingkan total barang.

2. Diagram Alur Sistem (Flowchart)

Flowchart berikut menggambarkan alur utama kerja fitur checklist dalam aplikasi.



Flowchart tersebut menunjukkan bahwa sistem selalu melakukan dua hal penting setiap kali terjadi perubahan: memperbarui status dan menyimpan riwayat aktivitas. Dengan mekanisme ini, aplikasi dapat memberikan umpan balik yang konsisten kepada pengguna.

Tabel 1. Hasil Pengujian Fungsional Fitur Smart Reminder

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
1	Menampilkan daftar barang	Aplikasi dijalankan	Daftar barang muncul di layar	Daftar tampil dengan benar	Berhasil
2	Memberi tanda checklist	Pengguna mencentang salah satu item	Status item berubah menjadi “tercentang”	Status berubah sesuai input	Berhasil
3	Perhitungan status kelengkapan	Beberapa item dicentang	Sistem menampilkan jumlah item yang sudah dicek	Status tampil akurat (misal 2/5)	Berhasil
4	Status lengkap	Semua item dicentang	Sistem menampilkan status lengkap	Pesan “Status: Lengkap” muncul	Berhasil
5	Reset checklist	Tombol reset ditekan	Semua tanda checklist kembali kosong	Checklist kembali ke kondisi awal	Berhasil
6	Penyimpanan log aktivitas	Pengguna mencentang dan menghapus centang	Setiap perubahan tercatat	Log tersimpan dengan waktu kejadian	Berhasil
7	Penyimpanan status	Aplikasi ditutup kemudian dibuka kembali	Status checklist tetap sama	Status tersimpan dan kembali muncul	Berhasil

Berdasarkan hasil pengujian yang dilakukan, seluruh fitur utama berjalan sesuai dengan rancangan.

- **Fitur checklist** mampu merespons setiap perubahan secara langsung. Setiap kali pengguna memberi atau menghapus centang, sistem memperbarui status secara real time.
- **Perhitungan status kelengkapan** bekerja akurat. Sistem menampilkan jumlah item yang telah dipersiapkan dibandingkan total barang, sehingga pengguna dapat mengetahui progres persiapannya.
- **Fitur reset** berfungsi sebagaimana mestinya. Seluruh checklist kembali ke kondisi awal sehingga pengguna dapat memulai persiapan baru tanpa kebingungan.

- **Log aktivitas mencatat setiap perubahan**. Setiap tindakan pengguna disimpan bersama cap waktu (timestamp), sehingga riwayat penggunaan dapat ditelusuri kembali.
- **Status tersimpan meskipun aplikasi ditutup**. Hal ini menunjukkan bahwa mekanisme penyimpanan sudah berjalan dengan baik dan mendukung kenyamanan pengguna.

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem:

- Stabil
- Mudah digunakan
- Memberikan umpan balik yang jelas
- Mesuai dengan tujuan penelitian

Namun, seperti telah dibahas sebelumnya, masih terdapat ruang pengembangan pada sisi notifikasi cerdas, integrasi mobile, dan pengaturan prioritas barang.

3. Analisis Fungsional

Pengujian fungsional menunjukkan bahwa seluruh fitur utama dapat berjalan sesuai perancangan.

- Checklist berfungsi dengan baik: setiap centang langsung tercatat dan tersimpan.
- Status kelengkapan akurat: sistem menghitung jumlah item yang sudah disiapkan.
- Reset bekerja efektif: seluruh status kembali ke kondisi awal tanpa gangguan.
- Log aktivitas mencatat perubahan: memudahkan pengguna melihat riwayat penggunaan.

Pengguna dapat memahami fitur tanpa membutuhkan panduan khusus karena tampilan antarmuka dirancang sederhana dan konsisten.

4. Manfaat Penggunaan Checklist

Berdasarkan hasil uji coba awal, fitur checklist memberikan beberapa manfaat:

- membantu pengguna menyiapkan barang secara sistematis,
- mengurangi potensi lupa ketika beraktivitas,
- menanamkan kebiasaan memeriksa ulang sebelum berangkat,
- menyediakan catatan aktivitas yang bisa dilihat kembali.

Dengan demikian, implementasi GUI checklist pada Smart Reminder tidak hanya berfungsi sebagai pengingat, tetapi juga sebagai alat bantu pengorganisasian aktivitas pribadi.

5. Pembahasan

Hasil implementasi menunjukkan bahwa konsep checklist sederhana dapat diterapkan secara efektif melalui antarmuka GUI.

Kekuatan utama sistem terletak pada:

- interaksi visual yang jelas,
- proses penandaan yang cepat,
- pembaruan status secara real time,
- adanya log sebagai catatan digital.

Dibandingkan dengan pengingat manual (catatan kertas atau memori pribadi), aplikasi ini memberikan kontrol yang lebih baik karena setiap tindakan terekam dan dapat dilihat kembali. Meskipun demikian, sistem masih memiliki beberapa keterbatasan. Aplikasi belum dilengkapi dengan notifikasi otomatis, belum terhubung ke perangkat seluler, dan belum tersedia pengaturan prioritas barang. Hal ini membuka peluang pengembangan lebih lanjut pada penelitian berikutnya.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan fitur checklist berbasis GUI pada aplikasi Smart Reminder sebagai alat bantu untuk memastikan kelengkapan barang sebelum beraktivitas. Fitur yang dikembangkan memungkinkan pengguna menandai item yang telah dipersiapkan, melihat status kelengkapan secara otomatis, melakukan reset, serta menyimpan riwayat aktivitas. Hasil pengujian menunjukkan bahwa sistem dapat berjalan stabil dan memberikan pengalaman penggunaan yang sederhana namun fungsional.

Penerapan checklist dalam bentuk aplikasi memberikan manfaat nyata bagi pengguna. Proses persiapan menjadi lebih terstruktur, risiko lupa barang dapat berkurang, dan pengguna memperoleh umpan balik visual mengenai tingkat kelengkapan barang yang telah dicek. Dengan demikian, aplikasi Smart Reminder berpotensi digunakan sebagai solusi praktis dalam mendukung aktivitas harian.

Meskipun telah berjalan sesuai tujuan, sistem masih memiliki beberapa keterbatasan. Aplikasi belum dilengkapi dengan fitur notifikasi otomatis, sinkronisasi lintas perangkat, dan pengaturan prioritas barang. Oleh karena itu, penelitian berikutnya dapat mengembangkan fitur lanjutan, melakukan uji coba dengan responden yang lebih beragam, serta menganalisis dampak penggunaan aplikasi terhadap kebiasaan pengguna dalam jangka panjang.

5. DAFTAR PUSTAKA

Aini, N., & Putra, R. A. (2021). Pengembangan aplikasi pengingat aktivitas berbasis checklist pada perangkat desktop. *Jurnal Teknologi Informasi dan Komputer*, 9(2), 45–53.

Fitriani, D., & Kurniawan, H. (2020). Rancangan antarmuka pengguna berbasis GUI untuk mendukung kemudahan penggunaan aplikasi. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 7(1), 12–20.

Hidayat, M., & Prasetyo, A. (2022). Implementasi sistem pengingat berbasis aplikasi untuk mendukung aktivitas harian pengguna. *Jurnal Sistem Informasi*, 14(3), 210–219.

Mardiana, S., & Nugroho, A. (2022). Evaluasi usability pada aplikasi desktop menggunakan pendekatan pengujian fungsional. *Jurnal Informatika Terapan*, 6(2), 97–106.

Nasution, F., & Sari, R. (2023). Penerapan checklist digital pada aplikasi manajemen aktivitas untuk meminimalkan kelupaan pengguna. *Jurnal Ilmu Komputer dan Sistem Informasi*, 11(1), 28–36.

Purnama, Y., & Rahmawati, S. (2021). Pengembangan aplikasi pengingat pintar berbasis antarmuka grafis. *Jurnal Sistem Komputer dan Aplikasi*, 5(2), 77–86.

Septiani, A., & Wahyudi, T. (2020). Analisis pengalaman pengguna pada aplikasi pengingat menggunakan pendekatan usability. *Jurnal Teknologi Informasi Terapan*, 8(3), 134–142.

Siregar, R., & Widodo, B. (2022). Desain dan evaluasi aplikasi checklist untuk mendukung manajemen aktivitas harian. *Jurnal Teknologi Informasi*, 10(1), 55–64.

Tanjung, A., & Lestari, D. (2021). Pengaruh desain antarmuka terhadap kemudahan penggunaan aplikasi pengingat. *Jurnal Sistem Informasi dan Teknologi*, 9(2), 101–110.

Utami, E., & Handayani, N. (2023). Integrasi fitur reminder dan checklist pada aplikasi produktivitas berbasis desktop. *Jurnal Informatika dan Aplikasi Komputer*, 12(1), 42–51.

Wahid, M., & Riyanto, S. (2020). Evaluasi fungsional aplikasi pengingat menggunakan pendekatan black-box testing. *Jurnal Rekayasa Sistem Informasi*, 6(2), 89–96.

Yuliana, R., & Fadhilah, N. (2023). Pengembangan aplikasi pengingat tugas dengan pendekatan user-centered design. *Jurnal Sistem dan Teknologi Informasi*, 11(2), 75–84.