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

Kohesi: Jurnal Multidisiplin Saintek
Volume 10 No 7 Tahun 2025

PERANCANGAN DAN IMPLEMENTASI MODUL MANAJEMEN DATA (TAMBAH/HAPUS BARANG) PADA APLIKASI SMART REMINDER MENGGUNAKAN PENYIMPANAN LOKAL

Safira Azahra[1*], Ahmad Tabrani[2]
[1,2] Teknik Informatika, Universitas Islam Negeri Sultan Maulana Hasanuddin Banten
E-mail: gmailsafiraazahra023@gmail.com[1], ahmad.tabrani@uinbanten.ac.id[2]

ABSTRACT

*Flexible item management is an essential feature in reminder applications, especially when users need to adjust their lists according to daily activities. This study presents the design and implementation of a data management module (add and delete items) in the **Smart Reminder** application using a Graphical User Interface (GUI). The module enables users to add new items, remove unnecessary ones, and view list updates instantly. Local storage is employed to ensure that data remains available even after the application is closed. The system was developed using the Python programming language with Tkinter as the main interface framework. The results indicate that the data management module runs reliably, is easy to operate, and improves checklist flexibility. Local storage is proven effective because it does not require internet connectivity while maintaining data consistency. This module provides a foundation for future enhancements, including device synchronization and item categorization features.*

Keywords: data management, add/delete items, GUI, local storage, Smart Reminder.

ABSTRAK

Pengelolaan daftar barang yang fleksibel menjadi kebutuhan penting pada aplikasi pengingat, terutama ketika pengguna perlu menyesuaikan daftar sesuai aktivitas harian. Penelitian ini membahas perancangan dan implementasi modul manajemen data (tambah dan hapus barang) pada aplikasi **Smart Reminder** berbasis Graphical User Interface (GUI). Modul dirancang agar pengguna dapat menambahkan item baru, menghapus item yang tidak diperlukan, serta melihat perubahan daftar secara langsung. Sistem memanfaatkan penyimpanan lokal sehingga data tetap tersimpan meskipun aplikasi ditutup. Pengembangan dilakukan menggunakan bahasa pemrograman Python dengan pustaka Tkinter sebagai antarmuka utama. Hasil implementasi menunjukkan bahwa modul manajemen data bekerja stabil, mudah digunakan, dan mampu meningkatkan fleksibilitas daftar checklist. Fitur penyimpanan lokal dinilai efektif karena tidak memerlukan koneksi internet dan tetap menjaga konsistensi data. Modul ini diharapkan menjadi dasar pengembangan fitur lanjutan, seperti sinkronisasi lintas perangkat dan pengaturan kategori barang.

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

Kata kunci: manajemen data, tambah/hapus barang, GUI, penyimpanan lokal, Smart Reminder.	

1. PENDAHULUAN

Kegiatan sehari-hari sering menuntut seseorang membawa berbagai barang penting, seperti telepon seluler, dompet, kartu identitas, dan perangkat kerja. Dalam praktiknya, pengguna masih sering lupa mempersiapkan sebagian barang, terutama ketika sedang terburu-buru. Kondisi tersebut dapat menimbulkan ketidaknyamanan, menurunkan produktivitas, bahkan mengganggu kelancaran aktivitas. Oleh karena itu, diperlukan sistem pengingat yang tidak hanya menampilkan checklist, tetapi juga memungkinkan pengguna menyesuaikan daftar barang sesuai kebutuhan.

Aplikasi pengingat berbasis Graphical User Interface (GUI) memberikan kemudahan interaksi karena pengguna dapat mengelola data melalui tombol, daftar, dan elemen visual lain yang intuitif. Pada konteks ini, modul manajemen data (tambah/hapus barang) menjadi bagian penting. Tanpa kemampuan menambah dan menghapus item, daftar checklist bersifat statis dan tidak mampu mengikuti perubahan aktivitas pengguna.

Penelitian ini memfokuskan pada perancangan dan implementasi modul manajemen data pada aplikasi Smart Reminder. Modul dirancang agar pengguna dapat:

1. menambahkan item baru ke daftar,
2. menghapus item yang tidak diperlukan,
3. melihat perubahan daftar secara langsung,
4. menyimpan data secara lokal agar tetap tersedia setelah aplikasi ditutup.

Penyimpanan lokal dipilih karena lebih ringan, tidak membutuhkan koneksi internet, dan sesuai untuk aplikasi desktop berskala kecil. Pengembangan dilakukan menggunakan bahasa pemrograman Python dengan pustaka Tkinter sebagai antarmuka utama.

Dengan adanya modul ini, aplikasi Smart Reminder menjadi lebih fleksibel dan adaptif. Pengguna dapat mengatur daftar barang sesuai situasi, sehingga checklist yang digunakan benar-benar mencerminkan kebutuhan aktual. Penelitian ini diharapkan berkontribusi pada pengembangan aplikasi pengingat sederhana yang fungsional, mudah digunakan, dan dapat diperluas pada tahap selanjutnya.

Selain permasalahan kelupaan barang, tantangan lain yang muncul adalah kebutuhan pengguna yang selalu berubah. Pada waktu tertentu, pengguna mungkin hanya membawa barang dasar, namun pada kesempatan lain membutuhkan perlengkapan tambahan seperti alat kerja, perlengkapan presentasi, atau dokumen tertentu. Jika daftar barang bersifat tetap, pengguna akan cenderung mengabaikan aplikasi karena tidak lagi sesuai dengan kondisi nyata. Oleh sebab itu, fleksibilitas pengelolaan data menjadi faktor utama agar aplikasi tetap relevan digunakan.

Penggunaan penyimpanan lokal dalam modul manajemen data memberikan beberapa keuntungan. Data dapat diakses lebih cepat karena tersimpan langsung di perangkat, proses penyimpanan lebih sederhana, serta tidak bergantung pada jaringan internet. Hal ini penting terutama bagi pengguna yang bekerja secara offline atau menggunakan perangkat dengan keterbatasan konektivitas. Dengan pendekatan ini, aplikasi tetap dapat berfungsi optimal tanpa menambah beban infrastruktur.

Selain itu, perancangan modul tambah dan hapus barang juga memperhatikan aspek kemudahan penggunaan. Setiap fungsi dirancang dengan alur yang jelas, sehingga pengguna dapat memahami cara kerja aplikasi hanya dengan mencoba secara langsung. Kombinasi antara desain antarmuka yang sederhana, penyimpanan lokal yang stabil, dan logika manajemen data yang terstruktur diharapkan mampu meningkatkan pengalaman pengguna secara keseluruhan. Dengan demikian, penelitian ini tidak hanya menghasilkan prototipe aplikasi, tetapi juga memberikan gambaran mengenai pentingnya modularitas dalam pengembangan sistem pengingat.

Dengan adanya modul manajemen data yang terintegrasi pada Smart Reminder, proses pengembangan menjadi lebih terstruktur karena setiap fungsi dipisahkan sesuai perannya. Modul tambah/hapus barang bertanggung jawab terhadap pengelolaan data, sedangkan modul checklist berperan dalam memverifikasi kelengkapan barang. Pemisahan ini memudahkan proses pemeliharaan, pengujian, dan pengembangan lanjutan, misalnya penambahan kategori, prioritas barang, atau integrasi dengan sistem pengingat berbasis waktu. Dengan demikian, penelitian ini tidak hanya membahas implementasi teknis, tetapi juga menekankan pentingnya perancangan arsitektur yang rapi dalam membangun aplikasi pengingat yang dapat dikembangkan di masa depan.

Cuplikan Kode Utama

(Logika Reminder berbasis Checklist — disederhanakan)

Cuplikan berikut menggambarkan alur dasar: daftar barang disimpan, item dapat dicentang, status diperbarui, dan log aktivitas direkam.

```
import json
from datetime import datetime

class SmartReminder:
    def __init__(self, path="data.json"):
        self.path = path
        self.items = []
        self.checked = {}
        self.logs = []
        self.load()

    def add_item(self, name):
        if name not in self.items:
            self.items.append(name)
            self.checked[name] = False
            self.logs.append(f"[{self.now()}] tambah: {name}")
            self.save()

    def delete_item(self, name):
        if name in self.items:
            self.items.remove(name)
            self.checked.pop(name, None)
            self.logs.append(f"[{self.now()}] hapus: {name}")
            self.save()

    def toggle(self, name):
        self.checked[name] = not self.checked[name]
        self.logs.append(f"[{self.now()}] {name} -> {self.checked[name]}")
        self.save()

    def status(self):
        done = sum(1 for i in self.items if self.checked.get(i))
        return f"{done}/{len(self.items)} sudah dicentang"

    def save(self):
        data = {"items": self.items, "checked": self.checked, "logs": self.logs}
        with open(self.path, "w") as f:
            json.dump(data, f)

    def load(self):
        try:
            with open(self.path) as f:
                data = json.load(f)
                self.items = data.get("items", [])
                self.checked = data.get("checked", {})
                self.logs = data.get("logs", [])
        except FileNotFoundError:
            self.save()

    def now(self):
        return datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

Inti logika yang terlihat:

- daftar barang disimpan secara lokal (json),
- item bisa ditambah dan dihapus,
- status checklist diperbarui setiap kali ada perubahan,
- setiap tindakan dicatat dalam log,
- data tetap tersedia meskipun aplikasi ditutup.

Struktur ini kemudian dihubungkan dengan GUI (Tkinter), sehingga pengguna berinteraksi melalui tombol tambah, hapus, dan checklist.

2. METODE PENELITIAN

1. Desain Penelitian

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan fokus pada perancangan dan implementasi modul manajemen data. Tujuannya adalah menghasilkan modul tambah dan hapus barang yang stabil, mudah digunakan, serta mampu menyimpan perubahan secara lokal.

Tahapan penelitian disusun sebagai berikut:

- analisis kebutuhan pengguna,
- perancangan arsitektur dan alur sistem,
- implementasi modul tambah/hapus barang,
- integrasi dengan checklist dan penyimpanan lokal,
- pengujian fungsional dan evaluasi sederhana.

Pendekatan ini dipilih agar proses penelitian tidak hanya menghasilkan konsep teoretis, tetapi juga prototipe yang dapat dijalankan.

2. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengetahui fungsi minimum yang harus tersedia pada modul manajemen data. Dari hasil analisis, diperoleh beberapa kebutuhan utama:

- pengguna dapat menambahkan item baru ke daftar,
- pengguna dapat menghapus item yang tidak diperlukan,
- daftar barang diperbarui secara langsung pada antarmuka,
- perubahan data tersimpan secara otomatis,
- data tetap tersedia ketika aplikasi dibuka kembali.

Selain kebutuhan fungsional, diperhatikan pula kebutuhan nonfungsional, seperti kemudahan penggunaan, kejelasan tampilan, dan respon sistem yang cepat.

3. Alat dan Lingkungan Pengembangan

Pengembangan sistem dilakukan pada lingkungan berikut:

- Bahasa pemrograman : Python
- Framework GUI : Tkinter
- Mekanisme penyimpanan : file penyimpanan lokal (JSON)
- Editor pengembangan : Visual Studio Code
- Sistem operasi : Windows

Pemilihan mekanisme penyimpanan lokal dilakukan karena prosesnya sederhana, ringan, dan sesuai untuk aplikasi desktop berskala kecil.

4. Perancangan Sistem

Perancangan sistem difokuskan pada pemisahan modul agar mudah dikelola.

Komponen utama meliputi:

1. **Modul Manajemen Data**
   Mengelola proses tambah dan hapus barang, sekaligus memperbarui file penyimpanan.
2. **Modul Checklist**
   Menampilkan daftar barang dan memberikan tanda pada item yang telah dipersiapkan.
3. **Modul Penyimpanan Lokal**
   Bertugas membaca dan menulis data ke file sehingga perubahan tetap tersimpan.
4. **Modul Log Aktivitas**
   Mencatat setiap perubahan sebagai riwayat penggunaan.

Dengan pembagian tersebut, setiap komponen memiliki tanggung jawab yang jelas.

5. Alur Kerja Sistem

Secara ringkas, alur kerja modul manajemen data adalah sebagai berikut:

- Pengguna memasukkan nama barang yang ingin ditambahkan.
- Sistem memeriksa apakah barang sudah ada pada daftar.
- Jika belum ada, barang disimpan ke dalam penyimpanan lokal.
- Daftar pada antarmuka diperbarui secara otomatis.
- Jika pengguna memilih item tertentu untuk dihapus, sistem menghapusnya dari daftar.
- Perubahan kembali disimpan dan diperbarui pada tampilan.

Alur ini memastikan bahwa data yang tampil selalu sama dengan data yang tersimpan.

6. Implementasi Modul

Tahap implementasi dilakukan secara bertahap:

- Membuat struktur data untuk daftar barang dan status checklist.
- Mengembangkan fungsi tambah dan hapus barang.
- Mengintegrasikan penyimpanan lokal menggunakan file JSON.
- Menghubungkan modul dengan antarmuka GUI.
- Menambahkan log aktivitas untuk setiap perubahan.

Dengan pendekatan bertahap, kesalahan lebih mudah ditemukan dan diperbaiki.

7. Pengujian Sistem

Pengujian dilakukan menggunakan metode black-box testing, yaitu dengan menguji keluaran berdasarkan input pengguna, tanpa melihat kode secara langsung.

Skenario pengujian meliputi:

- menambah barang baru,
- menghapus barang dari daftar,
- memastikan perubahan tersimpan setelah aplikasi ditutup,
- memeriksa apakah daftar tampil sesuai dengan data yang tersimpan,
- memastikan tidak ada duplikasi data.

Sistem dinyatakan berhasil apabila hasil yang muncul sesuai dengan perilaku yang diharapkan.

8. Evaluasi

Evaluasi sederhana dilakukan melalui pengamatan terhadap:

- kejelasan alur penggunaan,
- kecepatan respon ketika data diperbarui,
- konsistensi data antara tampilan dan file penyimpanan.

Hasil evaluasi digunakan sebagai dasar perbaikan dan bahan pertimbangan untuk pengembangan pada tahap berikutnya.

3. HASIL DAN PEMBAHASAN

1. Hasil Implementasi Sistem

Modul manajemen data berhasil diintegrasikan dengan aplikasi Smart Reminder sehingga pengguna dapat menambah dan menghapus barang secara dinamis.

Setiap perubahan daftar langsung tampil pada antarmuka, dan secara otomatis tersimpan pada penyimpanan lokal.

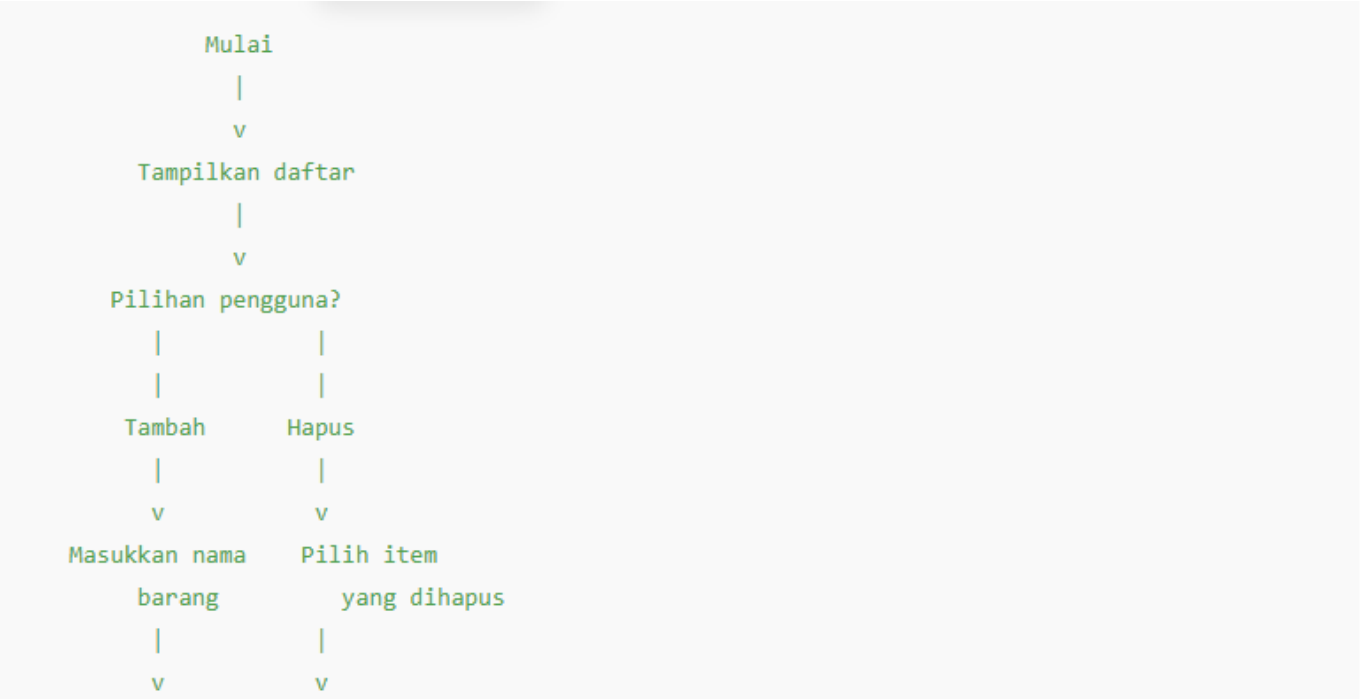
Fungsi utama yang berhasil diterapkan, yaitu:

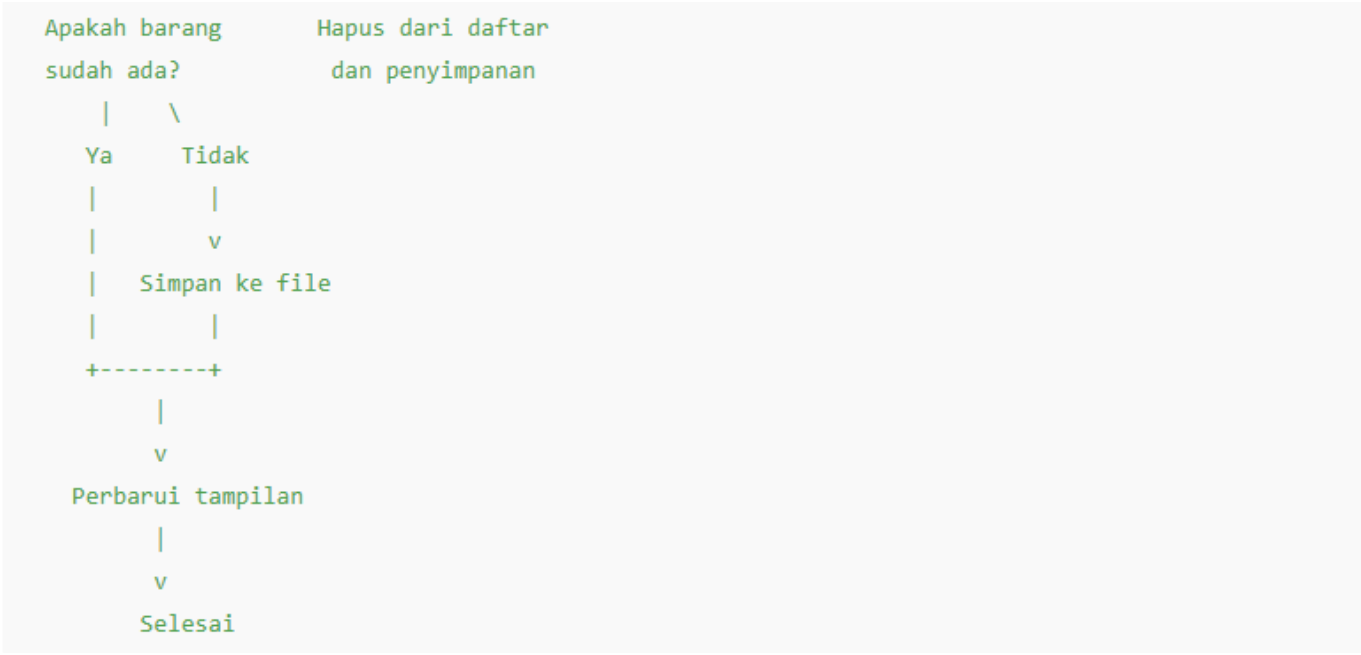
1. Tambah barang – pengguna memasukkan nama barang, kemudian item muncul pada daftar checklist.
2. Hapus barang – item tertentu dapat dihapus ketika sudah tidak digunakan.
3. Pembaruan otomatis – daftar pada GUI langsung menyesuaikan tanpa perlu memulai ulang aplikasi.
4. Penyimpanan lokal – daftar tetap tersimpan meskipun aplikasi ditutup dan dibuka kembali.
5. Log aktivitas – setiap perubahan dicatat sebagai riwayat.

Dengan hasil tersebut, aplikasi menjadi lebih fleksibel dan sesuai dengan kebutuhan pengguna yang berubah-ubah.

2. Flowchart Modul Tambah/Hapus Barang

Diagram berikut menggambarkan alur kerja modul secara sederhana:





Flowchart menunjukkan bahwa sistem selalu memastikan:

- tidak ada data duplikat
- perubahan langsung disimpan
- tampilan selalu mengikuti data terbaru

3. Tabel Ringkasan Hasil Pengujian

Tabel 1. Hasil Pengujian Modul Manajemen Data

No	Fitur yang Diuji	Skenario Pengujian	Hasil Diharapkan	Hasil Aktual	Status
1	Tambah barang	Pengguna memasukkan nama barang baru	Barang muncul pada daftar	Item tampil dan tersimpan	Berhasil
2	Duplikasi barang	Pengguna menambahkan barang yang sama	Sistem menolak/abaikan	Barang tidak digandakan	Berhasil
3	Hapus barang	Pengguna memilih item lalu menghapus	Item hilang dari daftar	Item terhapus dan tidak muncul kembali	Berhasil

No	Fitur yang Diuji	Skenario Pengujian	Hasil Diharapkan	Hasil Aktual	Status
4	Penyimpanan lokal	Aplikasi ditutup dan dibuka kembali	Data tetap tersedia	Data muncul sesuai terakhir	Berhasil
5	Pembaruan tampilan	Tambah/hapus dilakukan berulang	Tampilan mengikuti perubahan	Tampilan sinkron dengan data	Berhasil
6	Log aktivitas	Lakukan tambah dan hapus barang	Riwayat tercatat	Semua aktivitas tercatat	Berhasil

4. Pembahasan

Hasil pengujian menunjukkan bahwa modul manajemen data bekerja sesuai perancangan. Pertama, kemampuan tambah dan hapus barang membuat daftar menjadi dinamis. Pengguna dapat mengatur daftar sesuai kondisi, sehingga aplikasi tetap relevan digunakan dalam berbagai situasi. Kedua, penggunaan penyimpanan lokal terbukti efektif. Data tetap tersimpan tanpa membutuhkan internet, serta proses baca/tulis berlangsung cepat dan stabil. Ketiga, integrasi modul dengan checklist berjalan baik. Setiap perubahan pada daftar langsung berpengaruh pada status checklist, sehingga pengguna memperoleh gambaran kelengkapan yang akurat.

Selain keunggulan tersebut, terdapat beberapa catatan untuk pengembangan lanjutan:

- perlu ditambahkan validasi nama barang yang lebih ketat,
- perlu fitur kategori atau pengelompokan barang,
- perlu opsi pemulihan (undo) jika pengguna salah menghapus item.

Secara keseluruhan, modul ini meningkatkan fleksibilitas, kemudahan penggunaan, dan nilai fungsional Smart Reminder.

4. KESIMPULAN

Penelitian ini telah berhasil merancang dan mengimplementasikan modul manajemen data pada aplikasi Smart Reminder dengan kemampuan menambah dan menghapus barang serta menyimpan perubahan secara lokal. Modul yang dikembangkan memungkinkan pengguna menyesuaikan daftar barang sesuai kebutuhan, sehingga checklist tidak bersifat statis dan tetap

relevan untuk berbagai aktivitas. Integrasi dengan antarmuka berbasis GUI membuat proses pengelolaan data menjadi mudah dipahami dan nyaman digunakan.

Hasil pengujian menunjukkan bahwa seluruh fungsi utama berjalan sesuai dengan rancangan. Proses tambah dan hapus barang dapat dilakukan secara langsung, tampilan daftar diperbarui secara otomatis, dan data tetap tersimpan meskipun aplikasi ditutup. Penggunaan penyimpanan lokal terbukti efektif karena tidak bergantung pada koneksi internet, sekaligus menjaga konsistensi data antara tampilan dan file penyimpanan. Selain itu, pencatatan log aktivitas membantu mendokumentasikan setiap perubahan yang dilakukan pengguna.

Meskipun demikian, penelitian ini masih memiliki keterbatasan. Sistem belum menyediakan pengaturan kategori barang, fitur pemulihan (undo), maupun integrasi ke perangkat lain. Pada penelitian selanjutnya, modul dapat dikembangkan dengan penambahan sinkronisasi lintas perangkat, notifikasi cerdas, serta evaluasi usability dengan melibatkan lebih banyak responden. Dengan pengembangan lanjutan tersebut, Smart Reminder diharapkan dapat menjadi aplikasi pengingat yang lebih komprehensif, fleksibel, dan bermanfaat dalam mendukung aktivitas harian pengguna.

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 meningkatkan 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.

Lestari, D., & Ramadhan, F. (2022). Penerapan penyimpanan lokal pada aplikasi desktop untuk menjaga konsistensi data pengguna. *Jurnal Teknologi dan Sistem Informasi*, 8(2), 89–98.

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.

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, N. (2021). Validasi fungsional modul tambah–hapus data pada aplikasi berbasis desktop. *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.