



## IMPLEMENTASI SISTEM REMINDER OTOMATIS BERDASARKAN STATUS CHECKLIST SEBAGAI PENGINGAT KELENGKAPAN BARANG PADA APLIKASI SMART REMINDER

Riska Nurnajmah<sup>1\*</sup>, Ahmad Tabrani<sup>2</sup>

<sup>1,2</sup> Teknik Informatika, Universitas Islam Negeri Sultan Maulana Hasanuddin Banten

E-mail: [nurnajmahriska@gmail.com](mailto:nurnajmahriska@gmail.com)<sup>1</sup>, [ahmad.tabrani@uinbanten.ac.id](mailto:ahmad.tabrani@uinbanten.ac.id)<sup>2</sup>

### ABSTRACT

*Daily activities that are increasingly busy often cause people to forget essential items before leaving the house. This condition highlights the need for a simple and user-friendly reminder system that can operate automatically based on actual user needs. This study discusses the implementation of the Smart Reminder application, which provides automatic reminders based on the status of an item checklist. The application integrates checklist management, item organization, and automated reminders. When the checklist is incomplete, the system generates a reminder notification encouraging the user to review their items. Conversely, when all checklist items are completed, the system confirms that the user is ready to go. The system was tested through usage simulations across several daily activity scenarios. The results indicate that the application helps users reduce forgetfulness, organize preparations more effectively, and minimize the risk of leaving important items behind. In addition, the simple interface and clear workflow make the application easy to use. This work is expected to serve as a foundation for the development of more adaptive intelligent reminder systems in the future.*

**Keywords:** *automatic reminder, checklist system, smart reminder, item completeness, mobile application*

### Article History

Received: Desember 2025

Reviewed: Desember 2025

Published: Desember 2025

Plagiarism Checker No  
235

Prefix DOI :

[10.8734/Kohesi.v1i2.365](https://doi.org/10.8734/Kohesi.v1i2.365)

Copyright : Author

Publish by : Kohesi



This work is licensed  
under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)

**ABSTRAK**

Perkembangan aktivitas sehari-hari yang semakin dinamis sering membuat seseorang lupa menyiapkan barang-barang penting sebelum berpergian. Kondisi ini mendorong perlunya sistem pengingat yang sederhana, mudah digunakan, dan mampu bekerja secara otomatis berdasarkan kebutuhan pengguna. Penelitian ini membahas implementasi aplikasi Smart Reminder yang dirancang untuk memberikan pengingat otomatis berdasarkan status checklist kelengkapan barang. Aplikasi ini mengintegrasikan fitur checklist, manajemen item, dan pengingat otomatis. Ketika pengguna menandai checklist belum lengkap, sistem menampilkan notifikasi pengingat agar pengguna melakukan pengecekan kembali. Sebaliknya, jika seluruh checklist telah terpenuhi, sistem memberikan konfirmasi bahwa pengguna siap berangkat. Pengujian dilakukan melalui simulasi penggunaan pada beberapa skenario aktivitas harian. Hasil pengujian menunjukkan bahwa aplikasi mampu membantu pengguna meminimalkan kelalaian dalam mempersiapkan barang, meningkatkan ketertiban, serta mengurangi potensi kehilangan. Selain itu, antarmuka yang sederhana dan alur penggunaan yang jelas membuat aplikasi mudah dipahami. Penelitian ini diharapkan menjadi dasar pengembangan sistem pengingat cerdas yang lebih adaptif pada tahap selanjutnya.

**Kata kunci:** reminder otomatis, checklist, smart reminder, kelengkapan barang, aplikasi mobile

**1. PENDAHULUAN**

Aktivitas manusia yang semakin padat seringkali membuat seseorang lupa menyiapkan barang-barang penting sebelum berpergian, seperti dompet, kunci, kartu identitas, atau perlengkapan kerja. Kelalaian kecil ini dapat menimbulkan ketidaknyamanan, kerugian waktu, bahkan risiko keamanan. Oleh sebab itu, diperlukan solusi teknologi yang tidak hanya mengingatkan pengguna, tetapi juga mampu menyesuaikan pengingat berdasarkan kondisi nyata yang sedang dihadapi pengguna.

Aplikasi Smart Reminder dikembangkan sebagai sistem pengingat yang bekerja berdasarkan status checklist kelengkapan barang. Sistem ini tidak hanya menampilkan daftar barang yang perlu diperiksa, tetapi juga menghubungkan status checklist dengan mekanisme notifikasi otomatis. Ketika daftar belum lengkap, sistem akan memberikan peringatan, sedangkan saat semua checklist terpenuhi, aplikasi memberikan konfirmasi bahwa pengguna siap berangkat.



Pendekatan ini berbeda dengan aplikasi pengingat konvensional yang hanya mengandalkan alarm berbasis waktu. Dalam Smart Reminder, pengingat muncul karena adanya kondisi tertentu (condition-based reminder), sehingga lebih relevan dengan kebutuhan pengguna sehari-hari. Dengan demikian, aplikasi ini diharapkan dapat membantu mengurangi kelalaian, meningkatkan keteraturan, dan membangun kebiasaan persiapan yang lebih baik.

Selain itu, aplikasi dirancang dengan antarmuka sederhana, penggunaan ikon yang jelas, serta alur navigasi yang mudah dipahami. Hal ini penting karena keberhasilan sebuah aplikasi bukan hanya ditentukan oleh fitur, tetapi juga oleh kemudahan pengguna dalam berinteraksi dengan sistem. Oleh karena itu, penelitian ini berfokus pada proses implementasi, mekanisme kerja reminder otomatis, serta efektivitas penggunaannya dalam skenario penggunaan harian.

### Tampilan Aplikasi

Berikut gambaran tampilan halaman Reminder pada aplikasi Smart Reminder:

- Menu utama berisi: **Checklist – Tambah/Hapus – Reminder – Log**
- Bagian atas terdapat pesan pengingat:  
*“Reminder Alert! Jangan lupa yaa biar gak kelewat”*
- Status ditampilkan secara jelas, misalnya:

#### Checklist lengkap – Aman keluar rumah

Tampilan dibuat sederhana, berwarna lembut, dan fokus pada informasi utama sehingga mudah dipahami pengguna.

### Cuplikan Kode Utama (Logika Reminder)

Contoh logika inti sistem reminder berbasis checklist (disederhanakan):

```
def cek_reminder(checklist):  
    if all(item['status'] for item in checklist):  
        return "Checklist lengkap – aman keluar rumah."  
    else:  
        return "Checklist belum lengkap – mohon cek kembali!"
```



Logika tersebut:

1. Mengecek seluruh item checklist.
2. Jika semua status bernilai True, aplikasi menampilkan status aman.
3. Jika ada item belum dicentang, aplikasi memunculkan peringatan.

Pada implementasi sebenarnya, fungsi ini dihubungkan dengan:

- penyimpanan data checklist,
- tampilan antarmuka,
- serta mekanisme notifikasi.

Dengan pendekatan ini, sistem tidak sekadar menampilkan daftar, tetapi memberikan umpan balik cerdas kepada pengguna.

## **2. METODE PENELITIAN**

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak berbasis studi implementasi. Fokus penelitian tidak hanya pada proses pengembangan aplikasi, tetapi juga pada bagaimana sistem reminder otomatis bekerja dan diuji dalam skenario penggunaan sehari-hari.

Tahapan utama penelitian meliputi:

1. analisis kebutuhan,
2. perancangan sistem,
3. implementasi fitur reminder berbasis checklist,
4. pengujian fungsional,
5. evaluasi pengalaman pengguna.

Pendekatan ini dipilih agar hasil penelitian tidak hanya menghasilkan aplikasi, tetapi juga memberikan gambaran tentang kualitas fungsi yang dikembangkan.

### **Arsitektur Sistem**

Arsitektur aplikasi Smart Reminder terdiri atas beberapa komponen utama:

#### **1. Modul Checklist**

Menyimpan daftar barang yang perlu dipersiapkan (misalnya dompet, kunci, charger, kartu identitas).



## 2. Modul Pengelolaan Item (Tambah/Hapus)

Memberikan kebebasan pengguna untuk menambahkan atau menghapus item sesuai kebutuhan.

## 3. Modul Reminder Otomatis

Menghubungkan status checklist dengan pesan pengingat:

- o jika checklist belum lengkap → sistem memberi peringatan,
- o jika checklist lengkap → sistem menampilkan status aman.

## 4. Modul Log Aktivitas

Mencatat riwayat pengecekan checklist.

Secara konseptual, alur arsitektur dapat digambarkan sebagai berikut:

Input pengguna → Penyimpanan checklist → Proses pengecekan → Output (status/ reminder).

### Alur Kerja Sistem

Alur kerja sistem dirancang agar sederhana dan mudah dipahami pengguna.

- Pengguna membuka halaman checklist.
- Pengguna menandai barang yang sudah disiapkan.
- Sistem memproses status checklist.
- Sistem menghasilkan keluaran:
  - Checklist lengkap → menampilkan pesan “Aman keluar rumah”.
  - Checklist belum lengkap → menampilkan pesan pengingat untuk mengecek kembali.

Sebagian logika inti (disederhanakan) terlihat pada contoh berikut:

```
def cek_status(checklist):  
    if all(item["status"] for item in checklist):  
        return "Aman keluar rumah"  
    else:  
        return "Checklist belum lengkap"
```

Logika ini diintegrasikan dengan antarmuka aplikasi sehingga pengguna langsung melihat status tanpa perlu melakukan pengecekan manual satu per satu.



## **Implementasi Sistem**

Pengembangan aplikasi dilakukan menggunakan:

- bahasa pemrograman: **Python**,
- framework GUI: sesuai paket yang digunakan dalam proyek (misalnya customtkinter/tkinter),
- penyimpanan data sederhana (file / struktur list).

Tahapan implementasi:

- pembuatan antarmuka (menu, tombol, tampilan reminder),
- integrasi checklist dengan logika reminder otomatis,
- pengaturan pesan status dan notifikasi,
- penghubungan modul log untuk mencatat aktivitas.

Setiap modul diuji secara terpisah sebelum digabungkan ke dalam sistem utama.

## **Pengujian Sistem**

Pengujian dilakukan melalui black box testing, yaitu menguji fungsi berdasarkan keluaran yang dihasilkan.

Skenario utama pengujian:

### **1. Checklist lengkap**

- semua item dicentang,
- sistem harus menampilkan status aman.

### **2. Checklist tidak lengkap**

- satu atau lebih item belum dicentang,
- sistem harus menampilkan pengingat.

### **3. Penambahan dan penghapusan item**

- item baru ditambahkan,
- item dihapus,



- sistem tetap menampilkan reminder sesuai kondisi terbaru.

Pengujian juga dilakukan secara berulang menggunakan beberapa skenario aktivitas, seperti:

- bersiap ke sekolah,
- bepergian,
- perjalanan kerja.

### **Instrumen Penelitian**

Instrumen yang digunakan dalam penelitian meliputi:

1. lembar observasi,  
digunakan untuk mencatat respon dan perubahan perilaku pengguna saat menggunakan aplikasi.
2. daftar skenario pengujian,  
berisi langkah uji coba pada tiap kondisi checklist.
3. angket sederhana (opsional jika dipakai),  
untuk mengetahui kemudahan penggunaan, kejelasan tampilan, dan manfaat aplikasi.

Instrumen ini membantu memastikan bahwa aplikasi tidak hanya berjalan secara teknis, tetapi juga memberikan pengalaman penggunaan yang baik.

### **3. HASIL DAN PEMBAHASAN**

Aplikasi Smart Reminder berhasil diimplementasikan sesuai perancangan. Fitur utama yang berjalan dengan baik meliputi:

1. Checklist kelengkapan barang
2. Tambah/hapus item
3. Reminder otomatis berdasarkan kondisi checklist
4. Log aktivitas sebagai catatan riwayat pengecekan

Pengguna dapat menandai barang yang telah dipersiapkan, dan sistem secara otomatis memberikan umpan balik:

- jika masih ada item yang belum dicentang → muncul peringatan agar melakukan pengecekan ulang,
- jika semua item sudah selesai → sistem menampilkan status aman.



Hasil ini menunjukkan bahwa mekanisme condition-based reminder mampu bekerja secara efektif tanpa perlu alarm berbasis waktu.

### Diagram Alur (Flowchart) Sistem Reminder

Berikut diagram alur sederhana yang menggambarkan proses kerja aplikasi:



Flowchart ini memperlihatkan bahwa reminder bekerja berdasarkan kondisi checklist, bukan alarm waktu, sehingga lebih relevan dengan aktivitas pengguna.



## Hasil Pengujian Fungsional

Pengujian dilakukan menggunakan beberapa skenario utama.

**Tabel 1. Ringkasan Hasil Pengujian**

| Skenario Pengujian                  | Hasil yang Diharapkan                       | Hasil Aktual | Keterangan    |
|-------------------------------------|---------------------------------------------|--------------|---------------|
| Semua item dicentang                | Status “Aman keluar rumah”                  | Sesuai       | Berjalan baik |
| Ada item belum dicentang            | Peringatan “Checklist belum lengkap”        | Sesuai       | Berjalan baik |
| Tambah item baru                    | Item muncul di checklist dan ikut diperiksa | Sesuai       | Berjalan baik |
| Hapus item dari daftar              | Item hilang dan tidak diperiksa lagi        | Sesuai       | Berjalan baik |
| Perubahan status checklist berulang | Reminder berubah sesuai kondisi terbaru     | Sesuai       | Stabil        |

Hasil pengujian menunjukkan bahwa seluruh fungsi inti berjalan konsisten sesuai perancangan.

## Pembahasan

Hasil implementasi memperlihatkan bahwa:

### 1. Reminder berbasis kondisi lebih efektif

Pengguna tidak hanya diingatkan pada waktu tertentu, tetapi pada kondisi nyata checklist. Hal ini menurunkan risiko kelalaian ketika hendak bepergian.

### 2. Checklist membantu membangun kebiasaan persiapan

Pengguna menjadi lebih tertib karena proses pengecekan dilakukan secara bertahap dan sistem memberikan respon langsung.

### 3. Antarmuka sederhana meningkatkan keterlibatan pengguna

Tampilan sederhana dan kalimat yang mudah dipahami membuat aplikasi lebih ramah pengguna.

Dari sisi teknis, integrasi antara checklist dan sistem reminder berjalan stabil, sementara dari sisi pengguna, aplikasi dinilai membantu mengurangi potensi lupa membawa barang penting. Dengan demikian, aplikasi ini terbukti memiliki potensi untuk dikembangkan lebih lanjut, misalnya dengan:

- penyimpanan berbasis cloud,
- integrasi notifikasi smartphone,
- atau pengingat berbasis lokasi.



#### 4. KESIMPULAN

Penelitian ini membahas implementasi sistem reminder otomatis berbasis status checklist pada aplikasi Smart Reminder. Hasil pengembangan dan pengujian menunjukkan bahwa sistem mampu bekerja sesuai dengan tujuan, yaitu membantu pengguna melakukan pengecekan kelengkapan barang sebelum beraktivitas. Integrasi antara fitur checklist, pengelolaan item, dan reminder otomatis membuat proses pengecekan menjadi lebih terstruktur dan mudah digunakan.

Flowchart dan skenario pengujian menunjukkan bahwa mekanisme pengingat berbasis kondisi mampu memberikan respons yang lebih relevan dibandingkan pengingat berbasis waktu. Ketika checklist belum lengkap, sistem memberikan peringatan yang jelas, sedangkan saat seluruh checklist terpenuhi, sistem memberikan konfirmasi aman. Dengan cara ini, aplikasi dapat meminimalkan risiko lupa membawa barang penting serta membantu membentuk kebiasaan persiapan yang lebih baik.

Secara keseluruhan, implementasi Smart Reminder dinilai berhasil dari sisi fungsional dan kemudahan penggunaan. Ke depan, aplikasi ini masih dapat dikembangkan dengan menambahkan fitur notifikasi eksternal, sinkronisasi data, serta personalisasi pengingat agar semakin adaptif terhadap kebutuhan pengguna.

#### 5. DAFTAR PUSTAKA

- Basu, S., & Gavrilova, M. (2019). Smart reminder systems: State of the art and future directions. *Journal of Ambient Intelligence and Humanized Computing*, 10(4), 1357-1372.
- Dix, A., Finlay, J., Abowd, G., & Beale, R. (2004). *Human-computer interaction* (3rd ed.). Pearson Education.
- Hussain, A., & Ferneley, E. (2008). The role of adaptive systems in supporting personal reminder applications. *International Journal of Human-Computer Studies*, 66(12), 889-906.
- Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann.
- Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner's approach* (8th ed.). McGraw-Hill.
- Rahman, A., & Sari, R. (2021). Pengembangan aplikasi pengingat kegiatan berbasis Android menggunakan metode prototyping. *Jurnal Teknologi Informasi dan Komputer*, 9(2), 101-110.



- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2018). *Designing the user interface: Strategies for effective human-computer interaction* (6th ed.). Pearson.
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
- Todi, K., Jokinen, J., & Oulasvirta, A. (2016). Designing for reminders: A review of reminder systems. *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 1905-1912.
- Yusof, N., & Azmi, A. (2020). Checklist-based task management application to improve daily activity organization. *International Journal of Advanced Computer Science and Applications*, 11(9), 75-82.