

IMPLEMENTASI OTOMATISASI REKAP TO DO LIST OPENING DAN CLOSING DAY DI PT PRIMA SOLUSI COMPUTINDO DALAM Mendukung Efisiensi Operasional

Afiani Agus Abdillah¹, M. Fikri Avishena Parinduri², Muhamad Fahrizal³,
Seftia Della Fiisyatir Rodhiah⁴

dosen03164@unpam.ac.id¹, f.avishena@gmail.com², fzal3005@gmail.com³,
seftiadellafiisyatirrodhiah@gmail.com⁴

Universitas Pamulang

ABSTRAK

PT Prima Solusi Computindo (RADSoft) merupakan perusahaan pengembang perangkat lunak yang dalam kegiatan operasionalnya menghadapi kendala pada proses rekap To Do List Opening dan Closing Day yang dilakukan secara manual. Proses rekapitulasi belum memiliki standar waktu yang jelas, bergantung pada individu masing-masing, serta melibatkan banyak channel Discord dari tim Produk, Implementor, dan Development. Kondisi tersebut berpotensi menyebabkan ketidakkonsistenan data, keterlambatan laporan, dan inefisiensi operasional. Kerja praktik ini bertujuan untuk mengimplementasikan sistem otomatisasi rekap To Do List Opening dan Closing Day menggunakan Discord Bot berbasis Python. Sistem dikembangkan dengan pendekatan Software Development Life Cycle (SDLC) model Prototype, karena kebutuhan sistem belum sepenuhnya terdefinisi di awal sehingga dibangun secara bertahap dan dievaluasi bersama pengguna. Sistem terdiri dari fitur rekap otomatis terjadwal harian pukul 23:59 WIB, rekap manual melalui antarmuka web panel (Flask), pengolahan data dari berbagai channel dan sub-tim, serta pembuatan laporan dalam format file Excel (.xlsx) terstruktur menggunakan library openpyxl. Hasil implementasi menunjukkan bahwa sistem mampu berjalan secara otomatis dan terjadwal di server Linux Ubuntu 24, menghasilkan laporan per tim maupun laporan gabungan (all channel) dalam satu file. Sistem berhasil meningkatkan konsistensi data, mempercepat proses pembuatan laporan, serta mengurangi ketergantungan pada proses manual, sehingga mendukung efisiensi operasional perusahaan.

Kata kunci : Otomatisasi, Discord Bot, Rekap To Do List, Opening Day, Closing Day, Python, Flask, openpyxl, Efisiensi Operasional.

***Abstract -** PT Prima Solusi Computindo (RADSoft) is a software development company that faces operational challenges in the manual recapitulation process of the Opening and Closing Day To Do List. The recapitulation process lacks a standardized schedule, depends on individual initiative, and involves multiple Discord channels from the Product, Implementor, and Development teams. This situation potentially leads to data inconsistencies, report delays, and operational inefficiencies. This internship aims to implement an automated recapitulation system for the Opening and Closing Day To Do List using a Python-based Discord Bot. The system was developed following the Software Development Life Cycle (SDLC) Prototype model, as requirements were not fully defined at the outset and were refined iteratively with the development team. Key features include a daily scheduled automatic recap at 23:59 WIB, a manual recap feature through a Flask-based web panel, multi-channel data processing from various teams and sub-teams, and structured report generation in Excel format (.xlsx) using the openpyxl library. Implementation results demonstrate that the system successfully runs automatically and on schedule on a Linux Ubuntu 24 server, producing both per-team and consolidated (all-channel) reports in a single file. The system effectively improved data consistency, accelerated report generation, and reduced reliance on manual processes, thereby supporting the company's overall operational efficiency.*

Keywords: *Automation, Discord Bot, To Do List Recap, Opening Day, Closing Day, Python, Flask, openpyxl, Operational Efficiency*

1. PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat telah memberikan dampak signifikan pada berbagai aspek operasional perusahaan, termasuk dalam hal pengelolaan tugas dan koordinasi tim. Bagi perusahaan yang bergerak di bidang pengembangan perangkat lunak, efisiensi dalam pelaporan dan koordinasi aktivitas harian menjadi faktor krusial dalam mendukung kelancaran proyek.

PT Prima Solusi Computindo (RADSoft), sebagai perusahaan pengembang perangkat lunak yang berdiri sejak tahun 2013 dan berfokus pada solusi teknologi informasi di bidang manajemen keuangan, portofolio, asuransi, dan sistem informasi, menggunakan platform komunikasi Discord sebagai sarana koordinasi internal antar tim. Setiap hari, masing-masing anggota tim mencatat To Do List Opening dan Closing Day pada channel Discord yang telah ditentukan sebagai bentuk laporan aktivitas harian.

Namun, dalam praktiknya, proses rekapitulasi To Do List tersebut masih dilakukan secara manual. Proses ini belum memiliki standar waktu yang jelas (harian, mingguan, atau bulanan), bergantung pada individu masing-masing, serta melibatkan banyak channel dan tim yang terpisah meliputi tim Produk, Implementor, dan Development beserta sub-tim di bawahnya. Kondisi ini berpotensi menyebabkan ketidakkonsistenan data, keterlambatan dalam pembuatan laporan, serta inefisiensi operasional secara keseluruhan.

Sebagai upaya mengatasi permasalahan tersebut, dikembangkanlah sebuah sistem otomatisasi rekap To Do List berbasis Discord Bot menggunakan bahasa pemrograman Python. Sistem ini dirancang untuk membaca dan memproses pesan dari berbagai channel Discord, kemudian

menghasilkan laporan dalam format file Excel (.xlsx) yang terstruktur secara otomatis pada waktu yang telah ditentukan. Dengan adanya sistem ini, diharapkan proses rekapitulasi dapat berjalan secara konsisten, terstandarisasi, dan efisien tanpa bergantung pada proses manual.

Berdasarkan latar belakang yang telah diuraikan, identifikasi masalah mencakup: (1) proses rekapitulasi To Do List Opening dan Closing Day masih dilakukan secara manual sehingga memerlukan waktu dan berpotensi terjadi kesalahan; (2) banyaknya channel dan tim yang terpisah menyebabkan kesulitan dalam pengumpulan data secara terpusat dan konsisten; serta (3) belum adanya sistem otomatis yang dapat melakukan rekapitulasi data dan menghasilkan laporan secara terjadwal dan terstruktur.

Rumusan masalah dari penelitian ini adalah: bagaimana merancang sistem otomatisasi rekap To Do List yang dapat mengambil data dari berbagai channel Discord secara terjadwal, menghasilkan laporan rekap dalam format Excel yang terstruktur baik per tim maupun secara keseluruhan (all channel), serta menyediakan fitur manual berbasis web untuk melakukan rekap data secara fleksibel sesuai kebutuhan pengguna.

2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini meliputi observasi, wawancara, pengembangan sistem, dan studi pustaka. Observasi dilakukan untuk mengamati proses bisnis yang berjalan, khususnya alur pengisian dan rekap To Do List harian di berbagai channel Discord. Wawancara dilakukan langsung dengan pihak terkait di PT Prima Solusi Computindo untuk memahami kebutuhan sistem secara lebih mendalam.

Pengembangan Sistem menggunakan SDLC Prototype Model. Sistem dibangun secara bertahap dan dievaluasi langsung oleh tim pengembang maupun pengguna, sehingga kebutuhan yang belum terdefinisi di awal dapat diperjelas melalui iterasi prototipe. Model prototype dipilih karena kebutuhan sistem belum sepenuhnya jelas di awal pengembangan. Dengan adanya iterasi, sistem dapat dikembangkan secara bertahap sesuai dengan kebutuhan pengguna (Wibawa, 2023).

Studi pustaka dilakukan dengan mengkaji literatur, dokumentasi library, dan referensi teknis yang berkaitan dengan Discord API, Python, Flask, dan openpyxl. Sistem dikembangkan menggunakan bahasa pemrograman Python dengan library discord.py 2.x, framework Flask untuk web panel, serta library openpyxl untuk pembuatan laporan Excel. Sistem dijalankan pada server Linux Ubuntu 24 dengan spesifikasi CPU 1 Core, RAM 2 GB, dan Storage 60 GB.

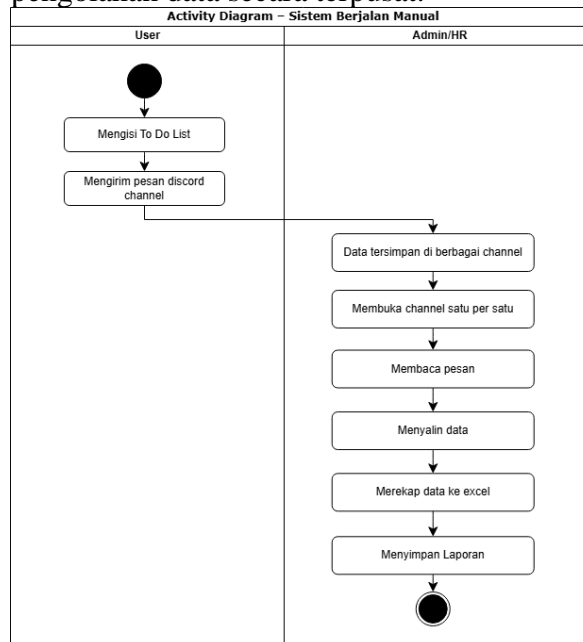
3. HASIL DAN PEMBAHASAN

3.1 Analisis Sistem Berjalan

Sistem yang berjalan saat ini dalam proses pelaporan To Do List masih dilakukan secara manual melalui platform Discord. Setiap anggota tim mencatat aktivitas harian berupa To Do List Opening dan Closing Day pada channel Discord yang telah ditentukan sesuai dengan tim masing-masing. Proses rekapitulasi laporan dilakukan oleh admin dengan cara membuka setiap channel Discord secara satu per satu, membaca dan menyalin data laporan dari masing-masing anggota tim, kemudian menyusunnya kembali ke dalam format laporan menggunakan aplikasi spreadsheet seperti Microsoft Excel.

Proses ini memiliki beberapa kelemahan, di antaranya memerlukan waktu yang cukup lama, berpotensi terjadi kesalahan pencatatan, serta tidak adanya standar waktu rekap yang konsisten. Selain

itu, data yang tersebar di berbagai channel juga menyulitkan proses pengumpulan dan pengolahan data secara terpusat.



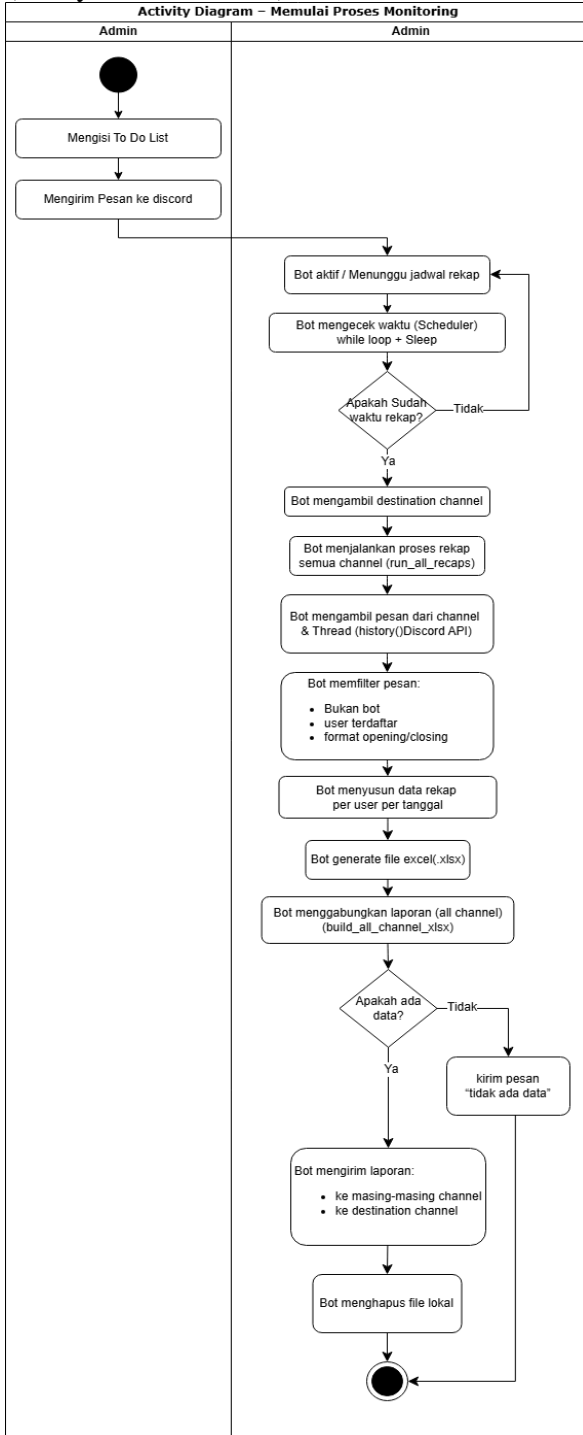
Gambar 1. Activity Diagram Sistem Berjalan

3.2 Analisis Sistem Usulan

Berdasarkan permasalahan pada sistem yang berjalan, diusulkan suatu sistem otomatisasi rekapitulasi To Do List berbasis Discord Bot. Sistem ini dirancang untuk menggantikan proses manual menjadi proses yang berjalan secara otomatis, terjadwal, dan terstruktur. Setiap anggota tim tetap mengirimkan To Do List melalui Discord seperti sebelumnya. Namun, data yang dikirimkan akan dibaca secara otomatis oleh Discord Bot yang telah dikonfigurasi untuk memantau channel tertentu. Bot kemudian melakukan proses parsing data untuk mengambil informasi yang dibutuhkan.

Selanjutnya, data yang telah dikumpulkan akan diolah dan disusun menjadi laporan dalam format file Excel menggunakan library openpyxl. Proses ini dapat berjalan secara otomatis berdasarkan jadwal yang telah ditentukan, yaitu setiap hari pada pukul 23:59 WIB. Selain itu, sistem juga menyediakan fitur manual melalui web panel berbasis Flask, sehingga operator dapat

menjalankan proses rekap kapan saja jika diperlukan. Setelah laporan selesai dibuat, file Excel akan dikirimkan secara otomatis ke channel Discord yang telah ditentukan (Wahyuni dkk., 2023).

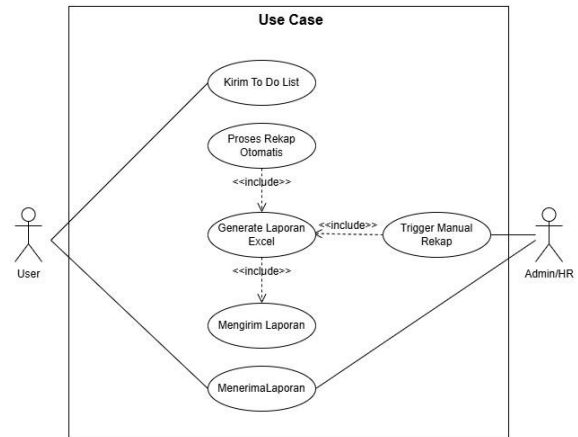


Gambar 2. Activity Diagram Sistem Usulan Berbasis Discord Bot

3.3 Perancangan Sistem (Diagram UML)

3.3.1 Use Case Diagram

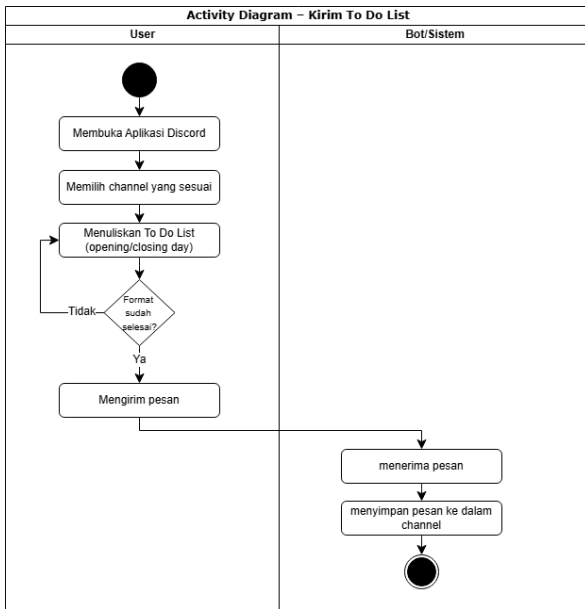
Use Case Diagram digunakan untuk menggambarkan interaksi antara aktor dengan sistem. Dalam sistem otomatisasi rekapitulasi To Do List berbasis Discord Bot ini, terdapat tiga aktor utama, yaitu User (anggota tim), Admin/Operator, dan Sistem (Discord Bot). User berperan dalam mengirimkan To Do List ke dalam channel Discord. Sistem (bot) bertugas membaca pesan, memproses data, serta menghasilkan laporan secara otomatis. Sedangkan Admin/Operator memiliki peran dalam melakukan monitoring serta menjalankan proses rekap secara manual melalui web panel jika diperlukan (Atmaja dkk., 2024).



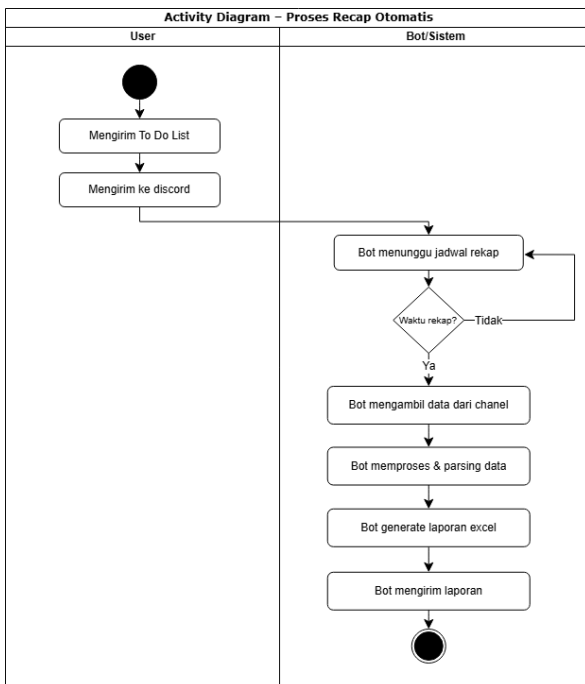
Gambar 3. Use Case Diagram Sistem Otomatisasi Rekapitulasi To Do List Berbasis Discord Bot

3.3.2 Activity Diagram

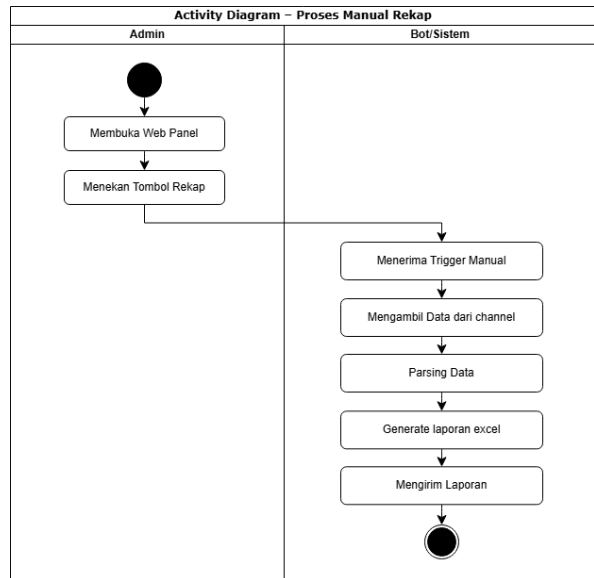
Activity Diagram digunakan untuk menggambarkan alur aktivitas atau proses yang terjadi dalam suatu sistem. Diagram ini menunjukkan urutan aktivitas dari awal hingga akhir, termasuk kondisi percabangan yang mungkin terjadi selama proses berlangsung. Dalam penelitian ini terdapat tiga Activity Diagram utama, yaitu: Activity Diagram Kirim To Do List, Activity Diagram Proses Rekap Otomatis, dan Activity Diagram Proses Manual (Trigger Admin) (Wahyuni dkk., 2023).



Gambar 4. Activity Diagram Kirim To Do List



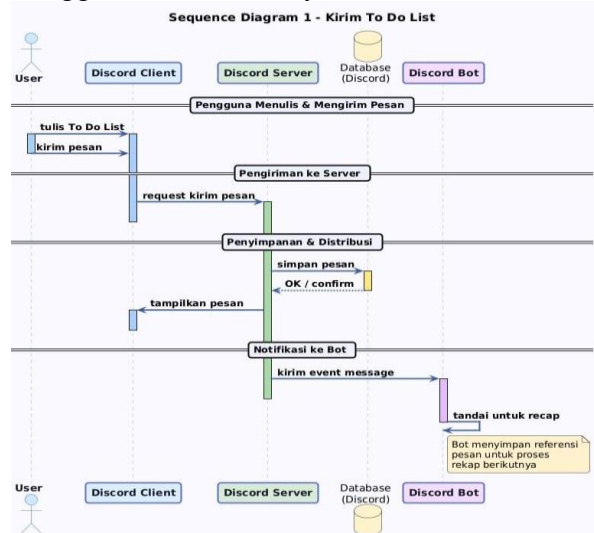
Gambar 5. Activity Diagram Proses Recap Otomatis



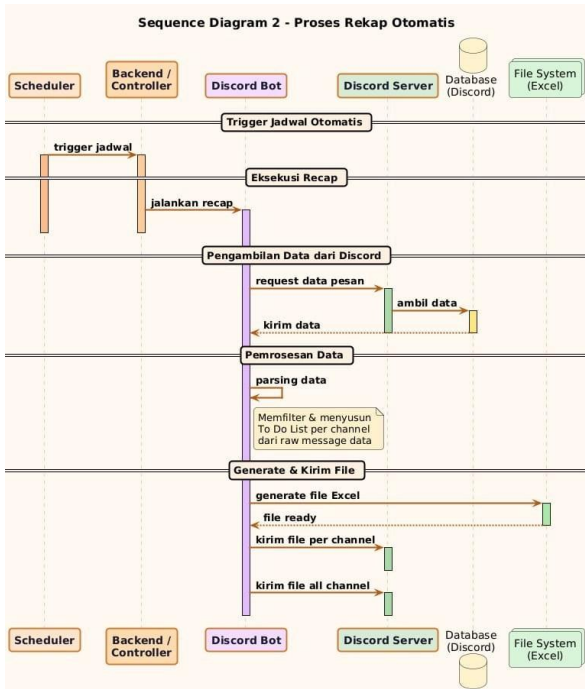
Gambar 6. Activity Diagram Proses Manual Recap (Trigger Admin)

3.3.3 Sequence Diagram

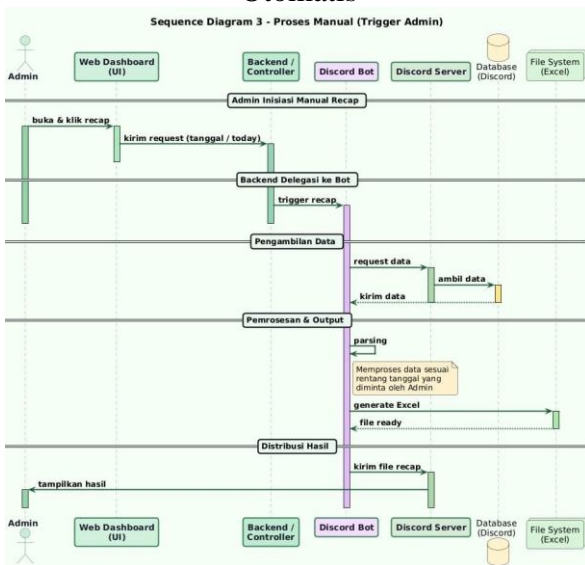
Sequence Diagram digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem berdasarkan urutan waktu. Diagram ini menunjukkan bagaimana pesan dikirim dan diterima antar komponen sistem secara berurutan. Dalam penelitian ini terdapat tiga Sequence Diagram, yaitu: Sequence Diagram Kirim To Do List, Sequence Diagram Proses Recap Otomatis, dan Sequence Diagram Proses Manual (Trigger Admin) (Wahyuni dkk., 2023).



Gambar 7. Sequence Diagram Kirim To Do List



Gambar 8. Sequence Diagram Proses Recap Otomatis

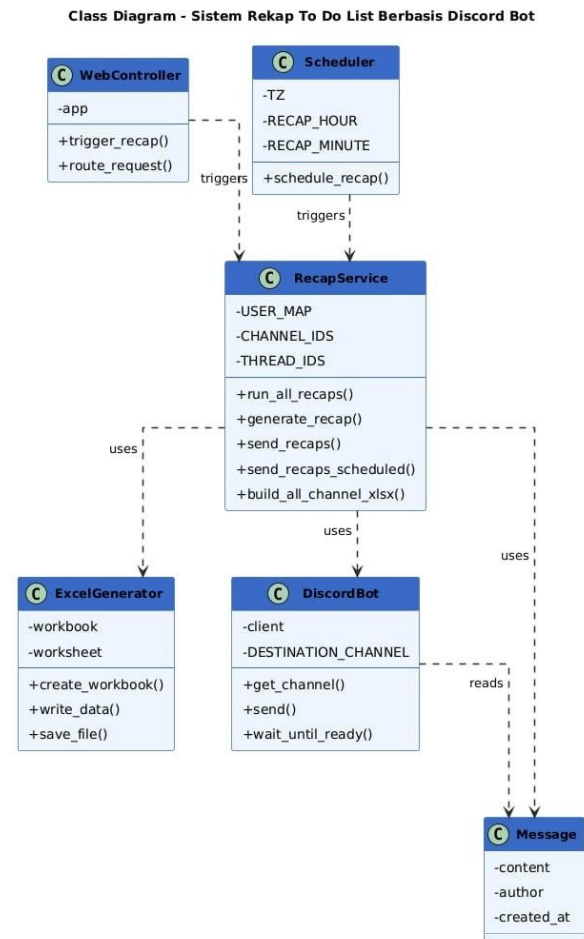


Gambar 9. Sequence Diagram Proses Manual (Trigger Admin)

3.3.4 Class Diagram

Class Diagram digunakan untuk menggambarkan struktur statis suatu sistem. Diagram ini menunjukkan kelas-kelas yang terdapat dalam sistem beserta atribut, metode (fungsi), serta hubungan antar kelas. Class Diagram pada sistem ini disusun berdasarkan struktur kode program, khususnya pada modul scheduler.py dan recap.py. Terdapat

lima kelas utama, yaitu: WebController, Scheduler, RecapService, ExcelGenerator, DiscordBot, dan Message (Wibawa, 2023).



Gambar 10. Class Diagram Sistem Recap To Do List Berbasis Discord Bot

3.4 Arsitektur Sistem

Sistem ini terdiri dari beberapa komponen utama: Discord Bot sebagai pengambil dan pemroses data; modul pengolahan data (recap.py) sebagai inti logika sistem; scheduler sebagai pengatur waktu eksekusi otomatis yang berjalan setiap pukul 23:59 WIB; serta web panel berbasis Flask sebagai antarmuka untuk menjalankan proses recap secara manual. Setiap komponen saling berinteraksi dalam satu lingkungan aplikasi berbasis Python yang berjalan secara asynchronous.

Konfigurasi sistem disimpan dalam file appsettings.json yang memuat token bot Discord, ID server (guild), daftar source

channel, ID destination channel untuk pengiriman laporan, serta pengaturan zona waktu (WIB/Asia/Jakarta). Modul recap.py merupakan inti dari sistem yang bertanggung jawab dalam proses pengolahan data dan pembuatan laporan. Fungsi `_collect_range()` digunakan untuk mengambil seluruh riwayat pesan dari satu source dalam rentang tanggal tertentu. Setiap pesan difilter berdasarkan zona waktu WIB, identitas pengguna, serta format pesan yang mengandung kata "OPENING" atau "CLOSING". Data yang telah difilter kemudian disusun dalam bentuk struktur dictionary bertingkat dengan format: `{tanggal: {user: {opening, closing, opening_late, closing_late}}}` (Direza & Fannani, 2025).

Web panel dikembangkan menggunakan framework Flask dan dijalankan sebagai thread terpisah menggunakan `threading.Thread` sehingga tidak mengganggu proses utama Discord Bot. Web panel menyediakan dua endpoint utama: GET / untuk menampilkan halaman antarmuka pengguna, dan POST /recap untuk menerima input tanggal dan memicu proses recap (Haeruddin dkk., 2025).

3.5 Perancangan Database

Pada sistem yang dikembangkan dalam kerja praktik ini, tidak digunakan database sebagai media penyimpanan data utama. Hal ini dikarenakan sistem dirancang untuk melakukan pengolahan data secara langsung (real-time processing) dengan mengambil data dari channel Discord, kemudian memprosesnya dan menghasilkan output berupa laporan dalam format file Excel (.xlsx). Data yang digunakan bersifat sementara (temporary), di mana data hanya diambil pada saat proses recap dilakukan, kemudian diolah dan langsung diekspor menjadi laporan tanpa disimpan dalam basis data permanen.

Sebagai pengganti database, sistem menggunakan: (1) Discord sebagai sumber data utama, yaitu tempat penyimpanan pesan To Do List yang dikirim oleh user; (2)

struktur data di dalam program (in-memory) untuk menyimpan hasil parsing sementara; serta (3) file Excel (.xlsx) sebagai media penyimpanan hasil akhir laporan.

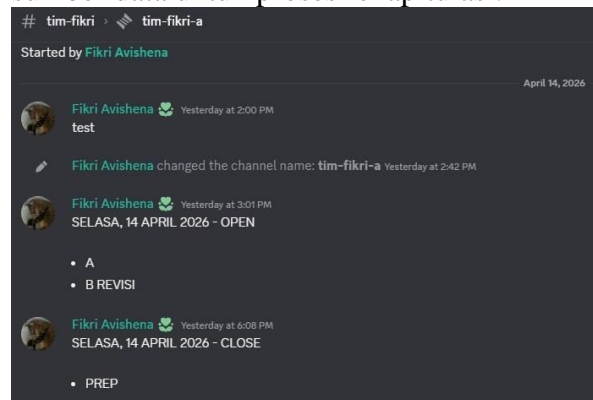
Tabel 1. Struktur Data Rekap Sistem

Field	Tipe Data	Keterangan
date	Date	Tanggal rekap data (format: YYYY-MM-DD)
source	String	Sumber data (Discord Channel)
user	String	Nama pengguna (user) yang mengirim pesan
opening	String	Status/jam Opening
opening_late	Boolean	Terlambat saat Opening (TRUE/FALSE)
closing	String	Status/jam Closing
closing_late	Boolean	Terlambat saat Closing (TRUE/FALSE)

3.6 Implementasi Sistem

3.6.1 Implementasi Input To Do List (Discord)

Pada tahap implementasi ini, user atau anggota tim mengirimkan To Do List harian ke dalam channel Discord yang telah ditentukan. Format pesan biasanya terdiri dari bagian Opening dan Closing yang berisi aktivitas yang dilakukan oleh masing-masing user. Sistem Discord Bot akan membaca pesan tersebut secara otomatis sebagai sumber data untuk proses rekapitulasi.

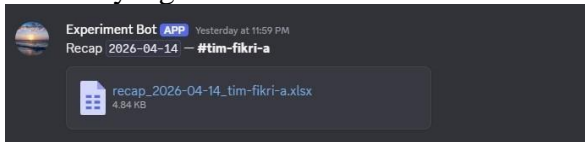


Gambar 11. Implementasi Input To Do List pada Discord

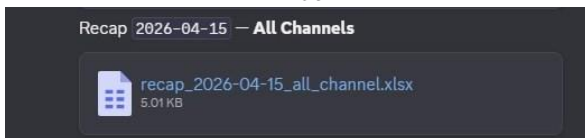
3.6.2 Implementasi Generate Laporan Excel

Setelah data berhasil dikumpulkan dan diproses, sistem akan menghasilkan laporan dalam bentuk file Excel (.xlsx) menggunakan library `openpyxl`. Laporan yang dihasilkan memiliki struktur kolom sebagai berikut:

Date, Source (Channel), User, Opening, Opening Late, Closing, dan Closing Late. Sistem menghasilkan dua jenis laporan, yaitu laporan per channel yang dikirim ke masing-masing channel, dan laporan gabungan all channel yang dikirim ke destination channel.



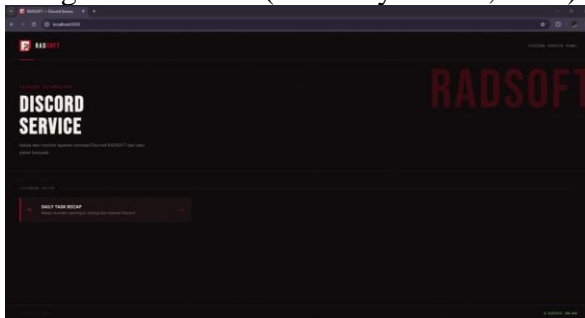
Gambar 12. Implementasi Hasil Laporan Excel



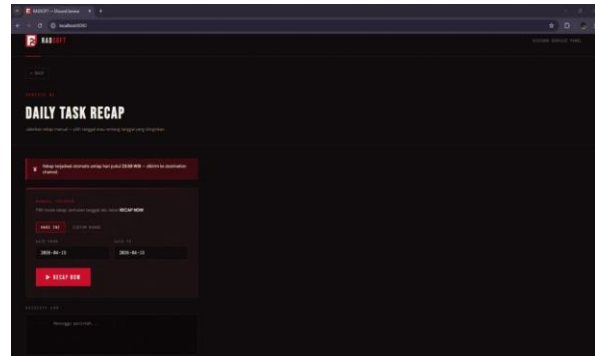
Gambar 13. Implementasi Laporan Gabungan All Channel

3.6.3 Implementasi Web Panel (Flask)

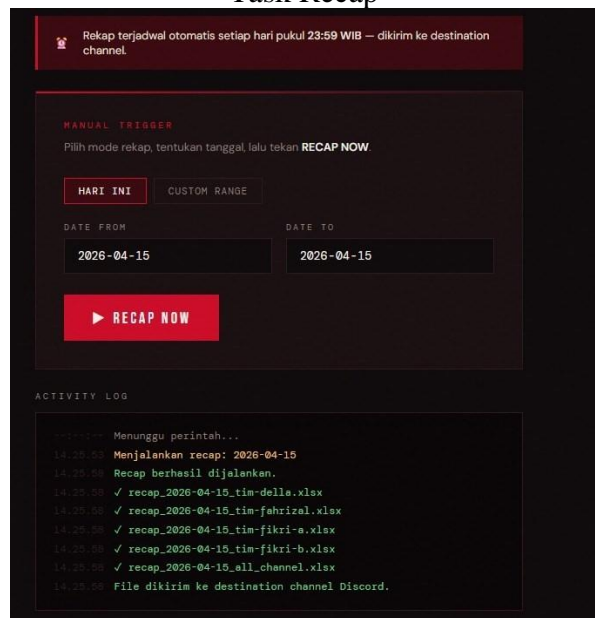
Web panel digunakan sebagai antarmuka bagi admin untuk menjalankan rekap secara manual di luar jadwal otomatis. Halaman ini menyediakan fitur: pemilihan tanggal (single date / range), tombol Recap Now, informasi jadwal otomatis, dan activity log. Sistem akan menampilkan status proses secara real-time ketika rekap dijalankan, sehingga admin dapat mengetahui apakah proses sedang berlangsung, berhasil, atau mengalami kendala (Mamuriyah dkk., 2025).



Gambar 14. Implementasi Halaman Dashboard Web Panel



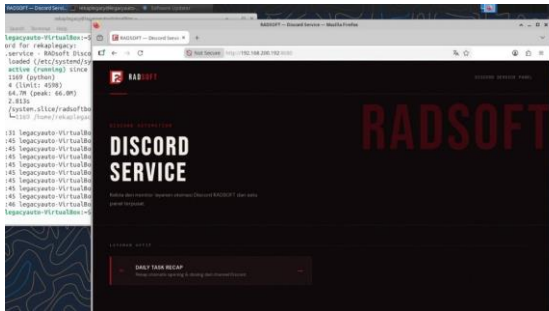
Gambar 15. Implementasi Halaman Daily Task Recap



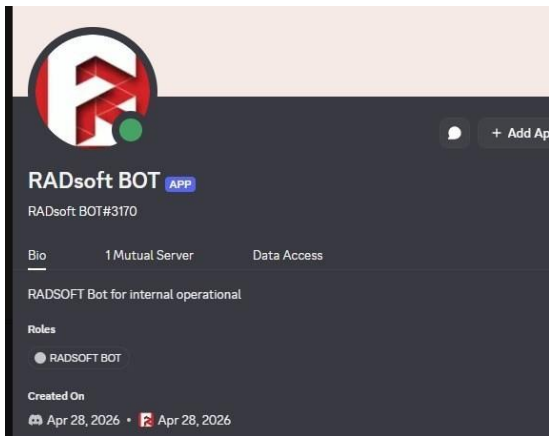
Gambar 16. Implementasi Proses Rekap pada Web Panel

3.6.4 Implementasi Deployment Sistem

Sistem otomatisasi rekapitulasi To Do List berbasis Discord Bot telah berhasil diimplementasikan dan dijalankan pada server berbasis Linux Ubuntu. Proses deployment dilakukan dengan menjalankan sistem sebagai service sehingga dapat berjalan secara terus-menerus (24/7) tanpa bergantung pada perangkat lokal. Sistem juga telah terintegrasi dengan Discord Bot yang aktif secara online, serta web panel berbasis Flask yang dapat diakses melalui jaringan.



Gambar 17. Tampilan Web Panel Discord Service yang Berjalan pada Server



Gambar 18. Status Discord Bot dalam Kondisi Online

3.7 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi dalam sistem berjalan sesuai dengan kebutuhan yang telah dirancang. Metode yang digunakan adalah Black-Box Testing, yaitu pengujian yang berfokus pada fungsi sistem tanpa melihat struktur internal kode program. Pengujian dilakukan pada fitur utama seperti pembacaan pesan Discord, proses rekap otomatis, pembuatan laporan Excel, serta pengiriman laporan ke channel tujuan. Hasil pengujian seluruh 27 skenario uji menunjukkan status berhasil sesuai output yang diharapkan.

Tabel 2. Hasil Pengujian Sistem (Black-Box Testing) — Ringkasan

Catatan: Total 27 skenario uji telah dilakukan. Tabel di atas merupakan ringkasan. Seluruh skenario menunjukkan hasil Berhasil sesuai output yang diharapkan.

3.8 Spesifikasi Server Deployment

No	Fitur yang Diuji	Skenario Pengujian	Output yang Diharapkan	Hasil	Status
1	Input To Do List	User mengirim pesan OPENING	Sistem membaca pesan	Sesuai	Berhasil
2	Input To Do List	User mengirim pesan CLOSING	Sistem membaca pesan	Sesuai	Berhasil
3	Filter User	User tidak terdaftar di USER_MAP	Data diabaikan	Sesuai	Berhasil
4	Parsing Data	Format pesan valid	Data berhasil diparsing	Sesuai	Berhasil
5	Validasi Opening	Opening sebelum jam 10:00	Status opening = valid	Sesuai	Berhasil
6	Validasi Opening	Opening setelah jam 10:00	Status opening_late	Sesuai	Berhasil
7	Generate Excel	Data tersedia	File .xlsx terbentuk	Sesuai	Berhasil
8	Scheduler	Waktu mencapai target 23:59 WIB	Rekap otomatis berjalan	Sesuai	Berhasil
9	Pengiriman Laporan	File tersedia	File terkirim ke Discord	Sesuai	Berhasil
10	Trigger Manual	Admin klik rekap	Rekap berjalan	Sesuai	Berhasil

Tabel 3. Spesifikasi Server Deployment.

Komponen	Spesifikasi
Sistem Operasi	Linux Ubuntu 24
CPU	1 Core
RAM	2 GB
Storage	60 GB
Python	3.10+
Bahasa Pemrograman	Python 3
Framework Web	Flask
Library Discord	discord.py 2.x
Library Excel	openpyxl

4. KESIMPULAN

Berdasarkan hasil analisis, perancangan, implementasi, dan pengujian sistem yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut. Pertama, sistem otomatisasi rekap To Do List dari berbagai channel Discord secara terjadwal telah berhasil dirancang dan diimplementasikan. Sistem yang dikembangkan menggunakan Discord Bot berbasis Python mampu membaca, mengumpulkan, dan memproses data dari berbagai channel secara otomatis dengan mekanisme scheduler harian pada pukul 23:59 WIB.

Kedua, sistem berhasil menghasilkan laporan rekap dalam format Excel yang terstruktur, baik per tim maupun secara keseluruhan (all channel). Laporan yang dihasilkan menggunakan library openpyxl memiliki format yang rapi dan sistematis, mencakup informasi penting seperti tanggal, user, opening, closing, serta status keterlambatan.

Ketiga, fitur manual berbasis web telah berhasil disediakan untuk mendukung fleksibilitas dalam proses rekap data. Web panel berbasis Flask memungkinkan admin atau operator untuk menjalankan proses rekap secara manual di luar jadwal otomatis, memberikan kemudahan kontrol serta fleksibilitas dalam kondisi tertentu tanpa mengganggu sistem utama.

Untuk pengembangan ke depan disarankan: penambahan fitur analitik dan

visualisasi data berupa dashboard grafik performa tim; integrasi dengan sistem manajemen proyek seperti Jira atau Trello; penambahan database untuk penyimpanan data historis; peningkatan keamanan dan manajemen akses pada web panel; serta pengembangan notifikasi otomatis jika terjadi error atau kegagalan proses.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, orang tua dan keluarga yang selalu memberikan doa dan dukungan, Ibu Afiani Agus Abdillah, S.Kom., M.Kom. selaku dosen pengampu mata kuliah Kerja Praktik, pihak PT Prima Solusi Computindo (RADSoft) yang telah memberikan kesempatan untuk melaksanakan kerja praktik, serta seluruh pihak yang telah membantu dalam penyusunan penelitian ini.

5. DAFTAR PUSTAKA

- Direza, D. R. J., & Fannani, I. (2025). Pengembangan sistem otomatisasi manajemen stok dan pelaporan gudang menggunakan n8n dan chat bot. *Jurnal Inovasi*.
<https://openjournal.unpam.ac.id/index.php/Inovasi/article/view/56100>
- Fransisco, dkk. (2022). Penerapan chatbot auto reply pada Telegram. *JTSI*.
<https://openjournal.unpam.ac.id/index.php/JTSI/article/view/14733>
- Haeruddin, dkk. (2025). Designing a chatbot application using the Flask framework and rule-based algorithm. *JTEKSIS*.
<https://jurnal.unidha.ac.id/index.php/jteksis/article/view/1820>
- Mamuriyah, N., dkk. (2025). Developing a chatbot system for PT. NG Tech Supplies based on the Python Flask framework. *JTEKSIS*.
<https://jurnal.unidha.ac.id/index.php/jteksis/article/view/1821>
- Pratama, R. Y. (2025). Pengembangan sistem chatbot cerdas berbasis NLP. *JITET*.
<https://journal.eng.unila.ac.id/index.php/jitet/article/view/8528>

11 Jurnal Ilmu Komputer dan Sistem Informasi (JIKSI)

- Putra, A. F. A., dkk. (2024). Pengembangan chatbot sederhana untuk informasi akademi menggunakan Python dan Flask. *Journal of Informatics and Business (JIBS)*.
<https://jurnal.ittc.web.id/index.php/jibs/article/view/1223>
- Vista, C. B., dkk. (2025). Pengembangan chatbot berbasis framework RASA. *Jurnal Informatika Polinema*.
<https://jurnal.polinema.ac.id/index.php/jip/article/view/9569>
- Wahyuni, I., dkk. (2023). Aplikasi chatbot akademik menggunakan Discord Bot. *ICIT*.
<https://ejournal.raharja.ac.id/index.php/icit/article/view/2952>
- Wibawa, S. (2023). Analisis chatbot otomatisasi tugas administratif dan manajemen dalam lingkungan digital dengan menggunakan Python. *INSANtek*.
<https://jurnal.bsi.ac.id/index.php/insantek/article/view/2190>
- Widiyatmoko, A., dkk. (2025). Smart temperature and humidity monitoring pada data center dengan laporan harian otomatis ke Discord. *KST*.
<https://jurnal.umnu.ac.id/index.php/kst/article/download/1742/823>