

ANALISIS DAN DESAIN USER INTERFACE APLIKASI OLAH DATA ABSENSI MENGGUNAKAN CHATGPT 4.0

Oleh:

Marcelino Ponto¹, Syaiful Rahman^{2*}, Abdul Munir S³

¹²³Teknologi Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Kharisma
e-mail: ¹marcelinoponto_21@kharisma.ac.id, ²syaifulrahman@kharisma.ac.id,
³abdulmunir@kharisma.ac.id

Abstrak: *Belakangan ini, penggunaan ChatGPT dalam rekayasa perangkat lunak semakin populer, terutama dalam penulisan kode program. Pemanfaatan ChatGPT dapat menghemat waktu dan biaya secara signifikan jika dapat diterapkan. Penelitian ini bertujuan untuk mengeksplorasi kemampuan ChatGPT dalam berbagai aspek rekayasa perangkat lunak, khususnya dalam analisis kebutuhan, pembuatan diagram aktivitas, use case, diagram kelas, arsitektur sistem, serta desain antarmuka pengguna aplikasi olah data mesin fingerprint absensi. Hasil penelitian kemudian dicermati menggunakan metode content analysis menggunakan data primer yang dihasilkan oleh ChatGPT. Dari penelitian ini dapat disimpulkan bahwa ChatGPT dapat digunakan dalam proses analisis dan desain aplikasi olah data absensi, meskipun masih memiliki beberapa keterbatasan pada tahap-tahap tertentu dalam rekayasa perangkat lunak. Campur tangan pengguna diperlukan untuk memvalidasi, melengkapi, dan merevisi kesalahan dari jawaban yang dihasilkan.*

Kata kunci: ChatGPT, Rekayasa Perangkat Lunak, Olah Data, Absensi, Diagram UML

Abstract: *Recently, the use of ChatGPT in software engineering has become increasingly popular, particularly in code development. The utilization of ChatGPT can significantly save time and costs if implemented effectively. This study aims to explore the capabilities of ChatGPT in various aspects of software engineering, specifically in requirements analysis, activity diagram creation, use case modeling, class diagram development, system architecture design, and user interface design for a fingerprint attendance data processing application. The research results are examined using content analysis methods based on primary data generated by ChatGPT. Based on this study, it can be concluded that ChatGPT can be utilized in the analysis and design process of an attendance data processing application, although it still has several limitations in certain stages of software engineering. User intervention is required to validate, supplement, and revise errors in the generated responses.*

Keywords: ChatGPT, Software Engineering, Data Processing, Attendance, UML Diagrams

1. PENDAHULUAN

Large Language Models (LLM) saat ini telah berkembang pesat, salah satu diantaranya ialah ChatGPT yang telah dapat menyusun kalimat dengan sangat baik [1]. Chat GPT merupakan *Large Language Models (LLM)* yang dikembangkan oleh OpenAI, di mana GPT itu kependekan dari *Generative Pre-trained Transformer* yang berarti Chat GPT dilatih dengan memproses text yang diberikan. Dengan menggunakan 570 gigabyte text sebagai bahan latihan,

* Corresponding author : Syaiful Rahman (syaifulrahman@kharisma.ac.id)

ChatGPT versi 3 memiliki pengetahuan yang sangat luas dan mampu membuat konten dengan menggunakan pattern yang telah dipelajarinya [2][3]. Hasilnya, walaupun ChatGPT tidak didesain untuk rekayasa perangkat lunak, ChatGPT mampu dalam menggenerasi konten yang dibutuhkan dalam rekayasa perangkat lunak [4]. Pada Mei 2024, OpenAI merilis versi GPT-4o di mana o merupakan singkatan dari kata omni. versi terbaru ini dapat memproses gambar, suara, dan text jauh lebih cepat dan dapat meniru interaksi manusia [5].

Pembuatan aplikasi atau rekayasa perangkat lunak memerlukan biaya yang banyak [6]. Penerapan ChatGPT dalam proses rekayasa perangkat lunak dapat mengurangi biaya dan waktu yang diperlukan dalam pengembangan aplikasi. Namun ChatGPT, pada dasarnya hanyalah sebuah *Large Language Model* (LLM) dan memiliki kekurangan. Kekurangan yang akan berdampak dalam aktivitas pengembangan software adalah kemampuan ChatGPT dalam membuat konten. Kemampuan membuat konten ini juga sering disebut dengan halusinasi, di mana ChatGPT akan membuat konten yang tidak tepat, tidak masuk akal, atau tidak berhubungan dengan input yang diberi [7][8].

Dalam *Software Development Life Cycle*(SDLC), terdapat enam tahap secara lengkap adalah Planning, Analisis, Desain, Implementasi, Testing, dan Maintenance. Tahapan ini bertujuan untuk menghasilkan perangkat lunak yang berkualitas dengan memperhatikan waktu dan biaya yang digunakan. Analisis adalah tahap di mana masalah dievaluasi, tujuan sistem ditentukan, dan bagaimana cara mencapai tujuan tersebut dirancang. Sementara Desain merupakan tahap di mana perangkat lunak dan perangkat keras ditentukan untuk memenuhi kebutuhan sistem yang dibangun [9]. Tahap analisis juga akan menghasilkan beberapa diagram seperti use case diagram, activity diagram, dan class diagram aplikasi. Sementara pada tahap desain akan dihasilkan arsitektur sistem dan desain UI/UX aplikasi [6]. Diagram-dia tersebut dibuat dengan mengikuti standar UML yang merupakan singkatan dari *Unified Modeling Language*. UML adalah standarisasi bahasa yang digunakan untuk memvisualisasikan dan membangun suatu sistem informasi. UML memiliki banyak diagram, dan yang digunakan didalam penelitian ini yaitu:

- Activity diagram digunakan untuk memodelkan apa saja aktivitas yang dilakukan oleh user untuk mencapai tujuannya. diagram ini terdiri dari titik mulai, aktivitas aktivitas yang dihubungkan secara berurutan, dan titik selesainya.
- Use case diagram digunakan untuk mendeskripsikan fungsi aplikasi tanpa menjelaskan struktur internalnya. Use case terdiri dari dua komponen, yaitu aktor dan use case/fungsi aplikasi yang digunakan.
- Class diagram digunakan untuk merepresentasikan tiap use case yang ada dalam use case diagram, dalam bentuk kelas-kelas di dalam suatu aplikasi. Tiap kelas ini memiliki variabel, fungsi, dan relasinya dengan kelas lain.
- Diagram Arsitektur digunakan untuk merepresentasikan struktur dan komponen dalam suatu sistem, dan juga bagaimana komponen tersebut berinteraksi satu

sama lain. Komponen tersebut seperti server, aplikasi, basis data, dan lapisan jaringan.

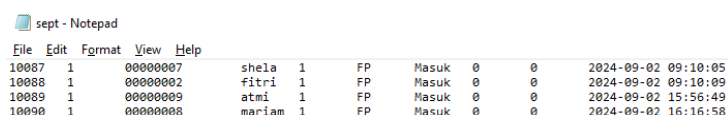
Implementasi UML dalam sebuah aplikasi dapat mempermudah pengembang yang memiliki latar belakang UML untuk memahami aplikasi tersebut dengan lebih mudah [10]. Berdasarkan pendahuluan diatas, tujuan penelitian ini adalah untuk menerapkan penggunaan ChatGPT dalam proses rekayasa perangkat lunak, khususnya pada analisis dan desain aplikasi. Penelitian ini mengangkat aplikasi yang berfungsi untuk mengolah data hasil mesin absensi fingerprint (sidik jari), sebagai objek rekayasa perangkat lunak yang akan dilakukan oleh ChatGPT versi 4.

Penelitian ini dilakukan dengan cara mendesain prompt, yaitu instruksi dalam bentuk teks yang sesuai dengan kebutuhan aplikasi, kemudian memberi instruksi tersebut kepada ChatGPT untuk melakukan rekayasa perangkat lunak, khususnya pada tahap analisis requirement, pembuatan diagram activity, use case, class, arsitektur, serta mendesain antarmuka pengguna (User Interface / UI) untuk aplikasi tersebut. Instruksi atau prompt ini akan diberikan berulang kali untuk menghasilkan data primer yang diproses menggunakan metode content analysis, isi data tersebut dianalisis apakah tujuan aplikasi tercapai, apakah isinya konsisten, apakah diagramnya sesuai standar UML, dan kelengkapan user interfacenya. Dalam pembuatan diagramnya, ChatGPT akan diberi instruksi untuk menggunakan format file *text* (.txt) yang dapat dibaca oleh aplikasi PlantUML untuk menjadi sarana visualisasi, sebagaimana dilakukan pada penelitian terkait oleh Ahmad et al. [11]. PlantUML adalah aplikasi yang mempermudah pembuatan berbagai macam diagram menggunakan bahasa yang intuitif dan sederhana [12]. Tingkat fleksibilitas ini yang menjadi alasan aplikasi ini dipilih sebagai sarana visualisasi diagram yang akan dibuat.

Dari langkah penelitian tersebut, maka penelitian ini diharapkan dapat memberikan pemahaman yang lebih mendalam mengenai penerapan ChatGPT dalam analisis dan desain perangkat lunak.

2. METODE PENELITIAN

Penelitian dimulai dengan mendeskripsikan data yang akan diolah disertai dengan fungsi apa saja yang dibutuhkan dalam penggunaan aplikasi. deskripsi ini disusun dalam bentuk paragraf yang kemudian akan dijadikan *prompt* untuk ChatGPT. Bentuk data yang akan diolah dapat lihat pada Gambar 1.



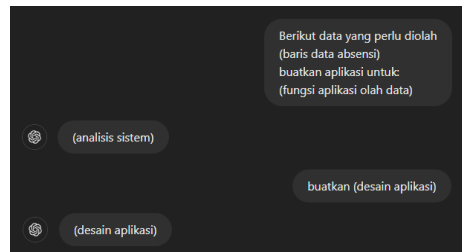
File	Edit	Format	View	Help
10087	1	00000007	shela	1 FP Masuk 0 0 2024-09-02 09:10:05
10088	1	00000002	fitri	1 FP Masuk 0 0 2024-09-02 09:10:09
10089	1	00000009	atmi	1 FP Masuk 0 0 2024-09-02 15:56:49
10090	1	00000008	mariam	1 FP Masuk 0 0 2024-09-02 16:16:58

Gambar 1. Data yang akan diolah

[sumber: data absensi Apotek Pelita Palu]

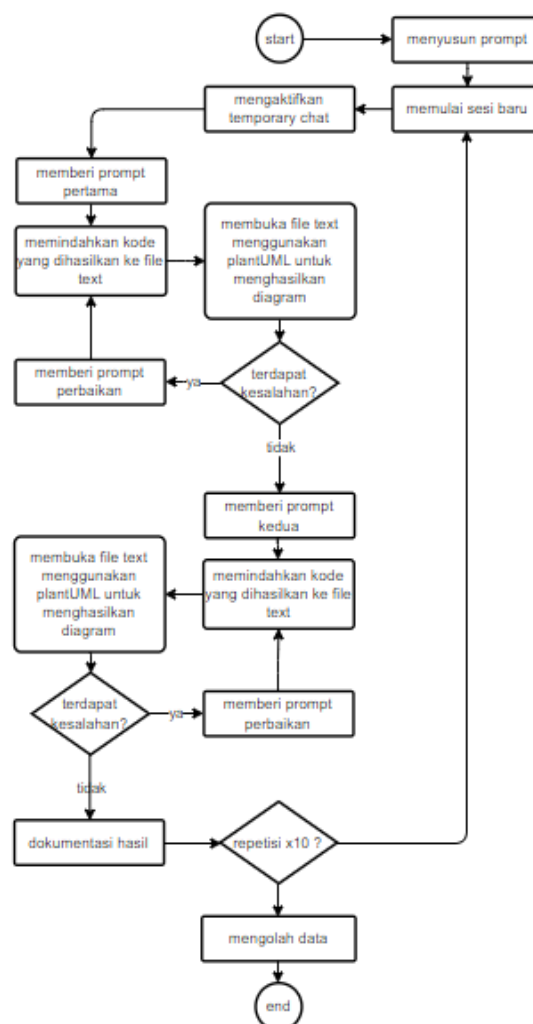
Analisis dan desain sistem dibuat dengan cara mengaktifkan fitur “*Temporary Chat*”(obrolan sementara), kemudian memberi *prompt* yang sebelumnya telah disusun dan

meminta ChatGPT untuk membuat requirement sistem, activity diagram, use case diagram, class diagram, serta arsitektur diagram dan desain UI untuk aplikasi tersebut seperti pada Gambar 2. Diagram-diagram tersebut akan divisualisasi oleh ChatGPT melalui file *text* yang dapat dibuka menggunakan PlantUML. Diagram yang dihasilkan kemudian dianalisis, dan *prompt* perbaikan akan diberikan jika terdapat kekeliruan yang perlu diperbaiki pada diagram yang dihasilkan.



Gambar 2. Memberi *prompt* kepada ChatGPT

Tahap ini akan diulang sebanyak sepuluh kali menggunakan *prompt* yang sama untuk menghasilkan data penelitian. Tahapan penelitian secara lengkap dapat dilihat pada Gambar 3.



Gambar 3. Langkah penelitian

3. HASIL DAN PEMBAHASAN

3.1. Penyusunan *Prompt*

Aplikasi yang akan dibuat adalah aplikasi yang membuka text file yang berisi barisan data absensi, mencari dan memasangkan jam masuk serta jam pulang setiap hari selama sebulan untuk setiap orang yang ada di dalam data tersebut. Aplikasi harus dapat membedakan absensi shift pagi yang dimulai dari jam 9 dan shift malam yang dimulai dari jam 4 sore. Laporan absensi tersebut harus menyertakan berapa jam total keterlambatan dan berapa total jam lembur.

sept - Notepad

File	Edit	Format	View	Help																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				</
------	------	--------	------	------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

Gambar 4. Baris text data absensi yang akan diolah.

Instruksi atau *prompt* juga perlu disertai dengan contoh format data yang akan diolah, contoh data ini dapat dilihat pada Gambar 4. Titik fokus pada data ini terdapat pada kolom 4 yang merupakan kolom nama, kolom 7 yang merupakan kolom deskripsi masuk atau pulang, kolom 11 yang merupakan tanggal, dan kolom 12 yang merupakan kolom jam dari log absensi. Berdasarkan dari kebutuhan tersebut maka deskripsi yang akan diberi sebagai input *prompt* kepada ChatGPT adalah sebagai berikut:

- *prompt* pertama

“berikut merupakan beberapa baris data dari file *text* yang dihasilkan oleh mesin absensi *fingerprint*:

```
“10099 1 00000007 shela 1 FP Masuk 0 0 2024-09-03 09:08:03
10101 1 00000001 adfais 1 FP Masuk 0 0 2024-09-03 16:01:31
10105 1 00000007 shela 1 FP Pulang 0 0 2024-09-03 16:25:26
10110 1 00000001 adfais 1 FP Pulang 0 0 2024-09-03 23:57:17”
```

di mana kolom ke 4 adalah nama pegawai, kolom ke 7 adalah deskripsi masuk atau pulang, kolom 11 adalah tanggal log, dan kolom 12 adalah waktu dari log tersebut. diperlukan sebuah aplikasi yang dapat mengimport file text yang akan dipilih, memasangkan dan menampilkan data jam masuk dan pulang, menghitung lama jam kerja serta lembur(jika lebih/kurang dari 7 jam) untuk semua pegawai pada tiap tanggal dan totalnya dalam bulan yang ditetapkan. buatlah analisis requirement, use case diagram, activity diagram, dan class diagramnya menggunakan PlantUML.”

- *prompt* kedua

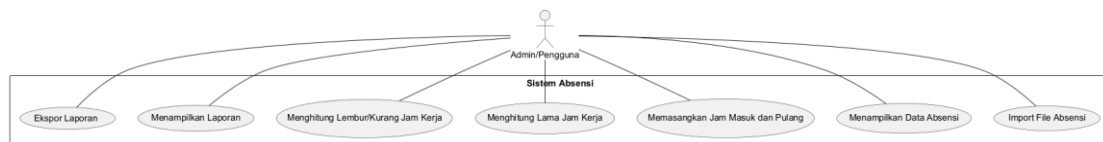
“Buatkan Desain arsitektur dalam PlantUML, serta Desain UI menggunakan bahasa pemrograman Python berdasarkan dari requirement, activity diagram, use case diagram, dan class diagram sebelumnya.”

Berdasarkan dari *prompt* tersebut, maka requirement yang perlu dipenuhi oleh aplikasi adalah sebagai berikut:

1. Mengimport file text.
2. Mencari jam pulang untuk absensi masuk setiap pegawai.
3. Menghitung total jam kerja serta lembur pegawai tiap hari dan total dalam sebulan.
4. Memfilter data berdasarkan bulan.
5. Menampilkan data yang telah diolah.

3.2. Hasil Analisis Aplikasi

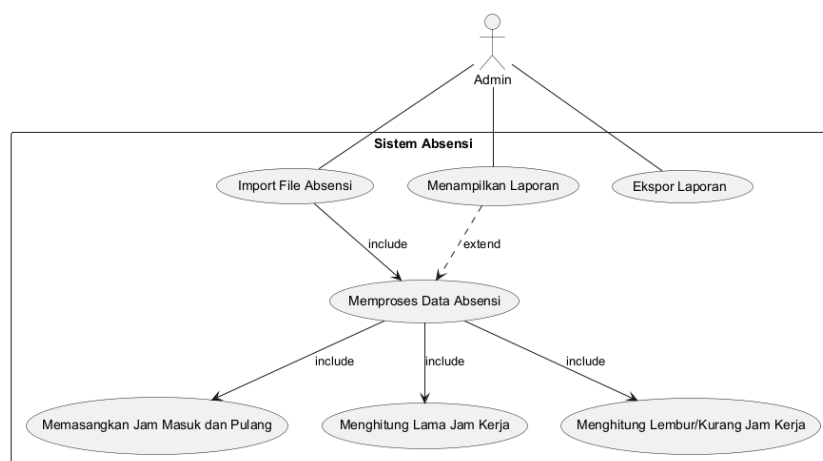
Berdasarkan dari deskripsi yang diberikan, ChatGPT dapat menghasilkan analisis requirement, use case diagram, activity diagram, dan class diagram walaupun terkadang terdapat kesalahan kecil pada pembuatan diagramnya. Analisis requirement yang dihasilkan memiliki kebutuhan fungsional dan non-fungsional yang lengkap. Pada use case diagram dapat terbuat kesalahan kecil di mana aktor diberi hubungan dengan use case sistem, ini dapat dilihat pada Gambar 5.



Gambar 5. Use case diagram sebelum diberi *prompt* perbaikan.

Gambar 5 merupakan use case diagram di mana use case diagram di mana user dapat melakukan pemasangan entry masuk dan pulang, hitung lama jam kerja, hitung lembur. ketiga use case tersebut, seharusnya fitur yang langsung dijalankan aplikasi setelah user mengimport file teks absensi. Untuk memperbaiki kesalahan ini, diberi *prompt* perbaikan sebagai berikut:

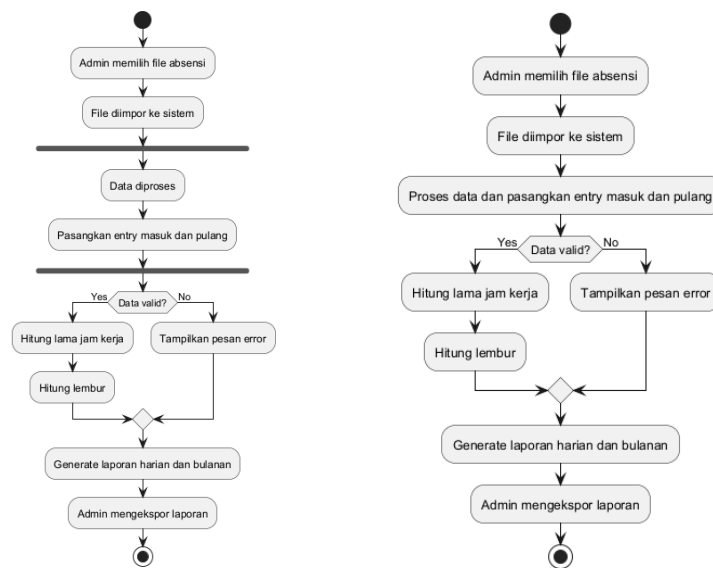
- “buat kembali use case yang lebih sesuai dengan activity diagram yg telah dibuat.”
- “perbaiki hubungan antar use case memproses data absensi dengan use case lainnya, menggunakan hubungan yang lebih cocok.”



Gambar 6. Use case diagram setelah diberi *prompt* perbaikan.

Prompt perbaikan pertama diberi untuk mendapatkan use case yang sesuai dengan alur penggunaan aplikasi. Hasil diagram tersebut masih terdapat kekeliruan kecil di mana hubungan

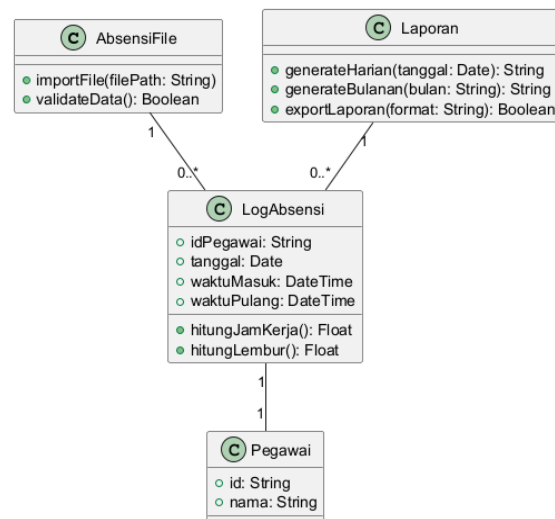
antara aktor dan use case menggunakan hubungan *generalization*. *Prompt* perbaikan kedua diberi untuk memperbaiki kekeliruan tersebut dan menghasilkan diagram seperti di Gambar 6.



Gambar 7. Activity diagram sebelum dan sesudah diberi *prompt* perbaikan.

Pada Activity diagram yang dihasilkan, ChatGPT menghasilkan alur activity yang memenuhi kebutuhan aplikasi. Dalam satu dari sepuluh percobaan terdapat kekeliruan penggunaan *Fork*, di mana *Fork* digunakan tanpa fungsi seperti yang dapat dilihat pada Gambar 7. *Prompt* perbaikan yang diberikan kekeliruan ini adalah sebagai berikut:

“perbaikan diperlukan pada diagram activity yang dihasilkan. terdapat penggunaan fork dan merge namun tidak terdapat pemecahan aktivitas di dalamnya.”

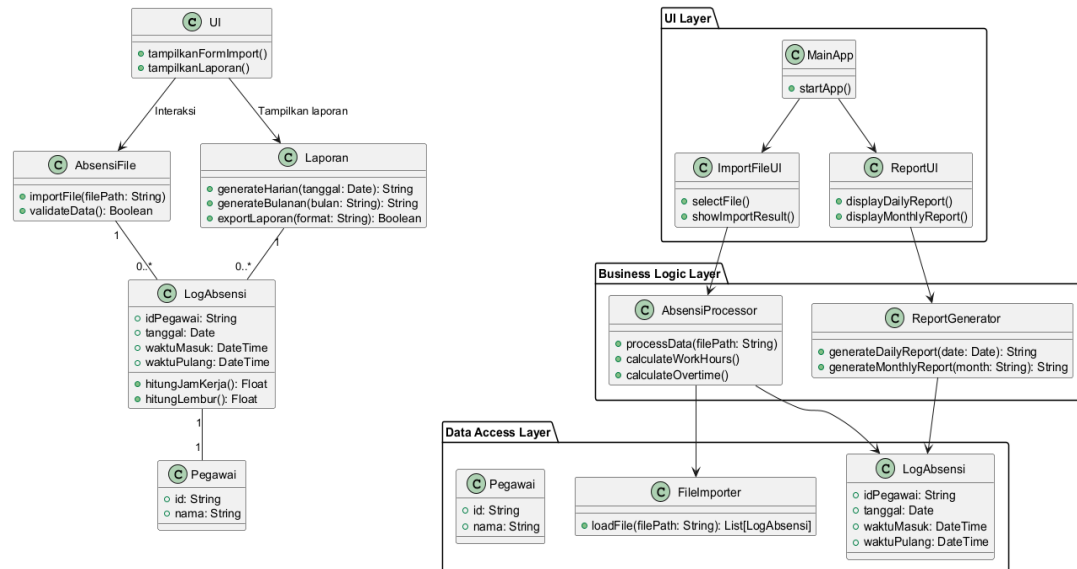


Gambar 8. Class diagram yang dihasilkan.

Pada Gambar 8 dapat dilihat class diagram yang dihasilkan oleh ChatGPT. Dari sepuluh class diagram yang dihasilkan, hanya terdapat satu class diagram yang menggunakan kardinalitas untuk menunjukkan jumlah objek yang dapat dimiliki atau dihubungkan oleh suatu kelas dalam suatu hubungan.

3.2. Hasil Desain Aplikasi

Dalam pembuatan Diagram arsitektur, ChatGPT menggunakan diagram dari class diagram untuk mendeskripsikan tiga komponen utama dalam aplikasi olah data di mana komponen tersebut adalah *User Interface* , *Business Layer* , dan *Data Layer*.

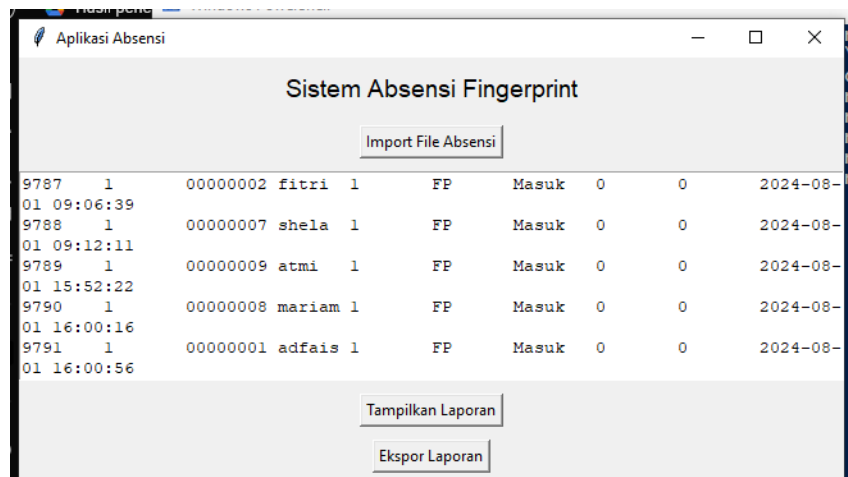


Gambar 9. Class diagram serta Arsitektur diagram yang dihasilkan.

Dalam pembuatan arsitektur diagram, ChatGPT terkadang menggunakan class diagram yang kurang konsisten dengan class diagram sebelumnya, di mana dibuat beberapa class baru dan terjadi perubahan fungsi dari kelas yang telah ada. Selain itu juga terdapat kelas yang tidak memiliki hubungan sama sekali. ini dapat dilihat pada Gambar 9, di mana kelas Pegawai tidak lagi berhubungan dengan kelas AbsensiFile yang namanya diubah menjadi AbsensiProcessor.

Dari sepuluh percobaan, terdapat empat percobaan di mana arsitektur diagram menggunakan class diagram yang kurang konsisten dengan class diagram sebelumnya. Selain itu juga, terdapat satu percobaan di mana *UI* dan *Business Layer* yang dihasilkan tidak memiliki hubungan dengan kelas yang berfungsi untuk mengimport file.

Desain UI merupakan tugas terakhir yang dikerjakan oleh ChatGPT, di mana ia membuat desain UI berdasarkan diagram yang telah dihasilkan sebelumnya menggunakan file Python. File Python ini dijalankan menggunakan windows powershell untuk menampilkan desain UI yang dihasilkan.



Gambar 10. Desain UI yang dihasilkan.

UI yang dihasilkan cukup sederhana, namun memenuhi kebutuhan yang telah ditetapkan. Dalam mendesain UI, ChatGPT menghasilkan beberapa variasi dalam menyajikan data laporan. Satu dari sepuluh percobaan langsung merekap data dan menuliskan rekap laporan. Sebagian besar, lima dari sepuluh percobaan menghasilkan UI yang memaparkan kembali data absensi ke dalam tabel sebelum di rekap, seperti yang dapat dilihat pada Gambar 10.

4. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa ChatGPT dapat digunakan untuk membuat *use case*, analisis kebutuhan, diagram aktifitas, dan desain antarmuka pengguna aplikasi sederhana seperti olah data absensi. Disisi lain, diagram yang dihasilkan tidak selalu akurat. Terdapat beberapa kesalahan kecil seperti penggunaan hubungan yang salah, penggunaan fungsi diagram yang kurang sesuai, dan kurangnya konsistensi antar diagram. Oleh karena itu, campur tangan pengguna masih diperlukan untuk mengoreksi serta menyesuaikan agar diagram memenuhi standar UML dan konsisten dari awal hingga akhir.

Meskipun pada penelitian ini ChatGPT berhasil digunakan untuk membuat analisis dan desain UI aplikasi olah data absensi, terdapat keterbatasan dalam tahap perbaikan diagram. Kekurangan ini terdapat pada tahap perbaikan diagram yang dihasilkan. Saat membuat perbaikan pada diagram, ChatGPT terkadang merubah komponen lain yang tidak perlu diperbaiki menghasilkan diagram yang kurang konsisten. Perbaikan ini sering dilakukan berulang kali hingga di titik dimana merubah diagram secara manual dengan cara mengedit kode PlantUML secara langsung jauh lebih cepat.

Untuk meningkatkan kualitas diagram yang dihasilkan, pengguna disarankan untuk melakukan validasi kembali diagram yang dibuat oleh ChatGPT dan mengoreksi kekurangan tersebut. Studi lebih lanjut juga dapat dilakukan untuk mengevaluasi kemampuan ChatGPT dalam membuat diagram untuk aplikasi berskala besar dan memiliki arsitektur yang lebih kompleks.

DAFTAR PUSTAKA

- [1] Z. Kenton, T. Everitt, L. Weidinger, I. Gabriel, V. Mikulik, and G. Irving, "Alignment of Language Agents," arXiv.org, 2021. <http://arxiv.org/abs/2103.14659>
- [2] A. Tamkin, M. Brundage, J. Clark, and D. Ganguli, "Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models," arXiv.org, 2021. <http://arxiv.org/abs/2102.02503>
- [3] N. M. S. Surameery and M. Y. Shakor, "Use Chat GPT to Solve Programming Bugs," International Journal of Information technology and Computer Engineering, vol. 3, no. 31, pp. 17–22, Jan. 2023, doi: <https://doi.org/10.55529/ijtc.31.17.22>.
- [4] S. Jalil, S. Rafi, T. D. LaToza, K. Moran, and W. Lam, "ChatGPT and Software Testing Education: Promises & Perils," IEEE Xplore, Apr. 01, 2023. <https://ieeexplore.ieee.org/abstract/document/10132255>
- [5] OpenAI. (2024). Chat GPT: Hello GPT-4o. OpenAI blog. <https://openai.com/index/hello-gpt-4o>
- [6] T. A. Pertiwi, N. T. Luchia, P. Sinta, R. Aprinastya, A. Dahlia, I. R. Fachrezi, and M. L. Hamzah. (2023). Perancangan dan Implementasi Sistem Informasi Absensi Berbasis Web Menggunakan Metode Agile Software Development. Jurnal Testing dan Implementasi Sistem Informasi Vol 1(1) 2023 : 53 - 66 <http://journal.al-matani.com/index.php/jtisi/article/view/325/285>
- [7] Y. Bang et al., "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity," arXiv.org, 2023. <http://arxiv.org/abs/2302.04023>
- [8] D. Kalla and N. Smith, "Study and Analysis of Chat GPT and its Impact on Different Fields of Study," papers.ssrn.com, Mar. 01, 2023. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4402499
- [9] Fania, V., Justin, F., Teresean, S., Vivian, O., & Adele, M.(2023). Mengenal Lebih Dalam Mengenai Analisis dan Desain Sistem. <https://sis.binus.ac.id/2023/03/08/mengenal-lebih-dalam-mengenai-analisis-dan-desain-sistem/>
- [10] N. Koch and A. Kraus, "The Expressive Power of UML-based Web Engineering 1." Accessed: Jun. 26, 2023. Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8d2ce9b7c2c82957715b6c7bb01f5b2924881122>
- [11] A. Ahmad, M. Waseem, P. Liang, M. Fahmideh, M. S. Aktar, and T. Mikkonen, "Towards human-bot collaborative software architecting with ChatGPT," ACM International Conference Proceeding Series, pp. 279–285, 2023. <https://doi.org/10.1145/3593434.3593468>
- [12] Plant UML, "Open-source tool that uses simple textual descriptions to draw beautiful UML diagrams," PlantUML.com. <https://plantuml.com/>