

## Implementasi Time Visit Monitoring Pada PT. Trans Retail Indonesia

<sup>1</sup>Joko Susilo, <sup>2</sup>Machudor Yusman

<sup>1</sup>Program Studi Informatika, Fakultas Komputer Universitas Mitra Indonesia

<sup>2</sup>Jurusan Ilmu Komputer, FMIPA, Universitas Lampung

Email: <sup>1</sup>jokosusilo.student@umitra.ac.id, <sup>2</sup>machudoryusman@yahoo.com

### Abstract

*This study aims to develop and implement a web-based Time Visit Monitoring system at PT Trans Retail Indonesia to improve the efficiency and accuracy of employee visit time recording. The company's main issues include manual visit recording, which is prone to errors, difficult to analyze, and causes delays in report preparation. The research employed a prototype development model consisting of requirements analysis, design, development, evaluation, refinement, and system implementation. The system design utilized UML (Use Case, Activity, Sequence, and Class Diagrams), while functional testing was carried out using the Black Box Testing method. The implementation results show that the system can record visit data in real-time, generate automated reports, and facilitate monitoring by management. This system has proven to enhance transparency, reduce the potential for fraud, and accelerate employee performance evaluation processes.*

**Keywords :** Time Visit Monitoring, Information System, Prototype, UML, Black Box

### Abstrak

*Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem Time Visit Monitoring berbasis web pada PT Trans Retail Indonesia guna meningkatkan efisiensi dan akurasi pencatatan waktu kunjungan karyawan. Permasalahan yang dihadapi perusahaan adalah proses pencatatan kunjungan yang masih manual, rentan kesalahan, sulit dianalisis, dan menyebabkan keterlambatan laporan. Metode penelitian menggunakan model pengembangan prototype yang melibatkan tahapan analisis kebutuhan, perancangan, pembuatan, evaluasi, perbaikan, hingga implementasi sistem. Desain sistem dibuat menggunakan UML (Use Case, Activity, Sequence, dan Class Diagram), dengan pengujian fungsional menggunakan metode Black Box Testing. Hasil implementasi menunjukkan bahwa sistem mampu mencatat data kunjungan secara real-time, menyajikan laporan otomatis, dan memudahkan proses pemantauan oleh manajemen. Sistem ini terbukti meningkatkan transparansi, mengurangi potensi kecurangan, serta mempercepat proses evaluasi kinerja karyawan.*

**Kata Kunci :** Time Visit Monitoring, Sistem Informasi, Prototype, UML, BlackBox

## 1. PENDAHULUAN

Teknologi Informasi adalah suatu teknologi yang digunakan untuk mengolah data, termasuk memproses, mendapatkan, menyusun, menyimpan, memanipulasi data dalam berbagai cara untuk menghasilkan informasi yang berkualitas, yaitu informasi yang relevan, akurat dan tepat waktu, yang digunakan untuk keperluan pribadi, bisnis, dan pemerintahan dan merupakan informasi yang strategis untuk pengambilan keputusan. Teknologi ini menggunakan seperangkat komputer untuk mengolah data, sistem jaringan untuk menghubungkan satu komputer dengan komputer yang lainnya sesuai dengan kebutuhan, dan teknologi telekomunikasi digunakan agar data dapat disebar dan diakses secara global.

Dalam dunia bisnis yang kompetitif, efektivitas dan efisiensi dalam pengelolaan tenaga kerja menjadi faktor penting bagi keberhasilan suatu perusahaan. Salah satu aspek krusial dalam manajemen sumber daya manusia adalah pemantauan waktu kunjungan karyawan ke berbagai lokasi operasional. PT Trans Retail Indonesia, sebagai perusahaan yang

bergerak di bidang ritel, memiliki kebutuhan untuk memastikan bahwa aktivitas operasional di setiap lokasi berjalan dengan optimal.

Dengan adanya sistem Time Visit Monitoring, perusahaan dapat mengotomatiskan proses pencatatan dan pemantauan kunjungan karyawan secara realtime. Sistem ini memungkinkan pencatatan waktu masuk dan keluar karyawan di lokasi tertentu secara akurat, serta menyediakan laporan yang lebih transparan dan mudah diakses oleh manajemen. Dengan implementasi sistem ini, perusahaan dapat meningkatkan efisiensi operasional, mengurangi potensi kecurangan, serta mempercepat proses evaluasi kinerja karyawan berdasarkan data yang valid dan objektif.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem Time Visit Monitoring di PT Trans Retail Indonesia. Sistem ini diharapkan dapat membantu perusahaan dalam meningkatkan akurasi pemantauan waktu kunjungan karyawan, mempercepat proses pelaporan, serta mendukung pengambilan keputusan yang lebih efektif dalam manajemen tenaga kerja. Dari penjelasan di atas maka penulis tertarik untuk membahas judul implementasi time visit monitoring pada PT Trans Retail Indonesia.

## **2. METODE PENELITIAN**

Pengumpulan data yang dilakukan pada penelitian ini adalah sebagai berikut :

### **1. Wawancara**

Pada metode wawancara penulis melakukan wawancara dengan beberapa pegawai dan atasan PT. Trans Retail Indonesia, penulis mengajukan beberapa pertanyaan seputar proses penilaian pegawai dan pendapat mereka tentang sistem yang akan di buat penulis.

### **2. Observasi Lapangan**

Pada metode observasi ini penulis melakukan pengamatan secara langsung di PT. Trans Retail Indonesia, dalam observasi ini penulis menguji coba bagaimana sistem yang berjalan sekarang dan yang akan penulis ajukan serta melakukan perbandingan secara langsung.

### **3. Studi Pustaka**

Mengumpulkan data, informasi serta pengetahuan dengan cara mencari dari buku-buku tentang teori bersangkutan dalam pembuatan website dan melakukan survey di PT. Trans Retail Indonesia.

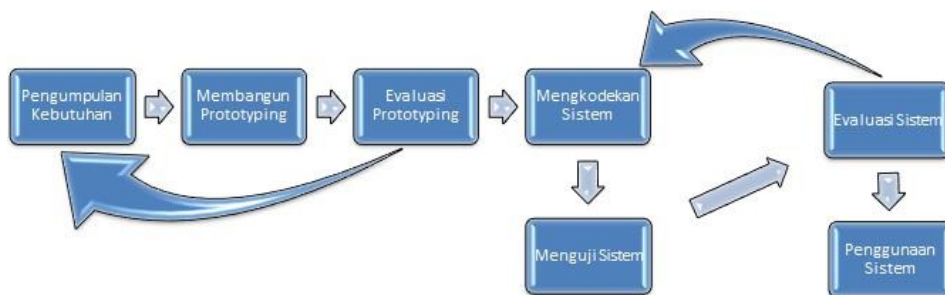
### **2.1 Metode Pengembangan Sistem**

Menurut (Ananda Muhamad Tri Utama, 2022) Model Prototipe merupakan suatu teknik dalam pengembangan perangkat lunak yang berfungsi untuk mengumpulkan informasi mengenai kebutuhan pengguna secara cepat. Proses ini berfokus pada penyajian aspek-aspek perangkat lunak yang akan terlihat oleh pelanggan atau pengguna. Sistem berbasis prototipe memungkinkan pengguna untuk memahami bagaimana sistem akan dibuat dan beroperasi. Pengguna berperan aktif dalam mengevaluasi prototipe, memberikan masukan, dan menyaring kebutuhan pengembangan perangkat lunak. Pendekatan prototipe mempermudah pengguna dalam mengenal sistem yang sedang dikembangkan, sekaligus memungkinkan perubahan dan penyesuaian selama proses pengembangan agar sistem sesuai dengan kebutuhan dan keinginan pengguna. Model prototype ini memiliki beberapa tahapan sebagai berikut:

- Requirements Gathering and Analysis (Analisis Kebutuhan)

Tahapan model prototype dimulai dari analisis kebutuhan. Dalam prosesnya, klien dan tim developer akan bertemu untuk mendiskusikan detail sistem seperti apa yang diinginkan oleh user.

- **Quick Design (Desain cepat)**  
Tahap kedua adalah pembuatan desain sederhana yang akan memberi gambaran singkat tentang sistem yang ingin dibuat. Tentunya berdasarkan diskusi dari langkah 1 di awal.
- **Build Prototype (Bangun Prototipe)**  
Setelah desain cepat disetujui selanjutnya adalah pembangunan prototipe sebenarnya yang akan dijadikan rujukan tim programmer untuk pembuatan program atau aplikasi.
- **User Evaluation (Evaluasi Pengguna Awal)**  
Di tahap ini, sistem yang telah dibuat dalam bentuk prototipe di presentasikan pada klien untuk di evaluasi. Selanjutnya klien akan memberikan komentar dan saran terhadap apa yang telah dibuat.
- **Refining Prototype (Memperbaiki Prototipe)**  
Jika klien tidak mempunyai catatan revisi dari prototipe yang dibuat, maka tim bisa lanjut pada tahapan 6, namun jika klien mempunyai catatan untuk perbaikan sistem, maka fase 4-5 akan terus berulang sampai klien setuju dengan sistem yang akan dikembangkan.
- **Implement Product and Maintain (Implementasi dan Pemeliharaan)** Pada fase akhir ini, produk akan segera dibuat oleh para programmer berdasarkan prototipe akhir, selanjutnya sistem akan diuji dan diserahkan pada klien.



**Gambar 1 Model Prototype**

## 2.2 Metode Perancangan Sistem

Menurut (Destriana et al., 2022) UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corps. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Pembagian kategori dan macam-macam diagram UML dapat dilihat di bawah ini:


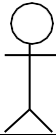

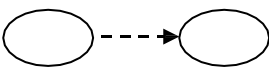
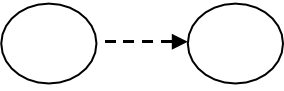
1. Structure diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. Behavior diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. Interaction diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

Diagram-diagram yang digunakan dalam UML berupa Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram. Diagram yang merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas, juga merupakan dokumentasi yang baik. Tujuan menggunakan diagram-diagram UML ini adalah memberikan pandangan umum sistem, memperlihatkan sebuah proses yang berinteraksi dengan lingkungannya. empat diagram yang digunakan, yaitu terdiri dari:

### 1. Use case diagram

Pemodelan dari behavior diagrams untuk kelakuan sistem informasi yang akan dibuat. Mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Simbol-simbol yang digunakan untuk pembuatan use case diagram dapat dilihat pada tabel 3.1 sebagai berikut:



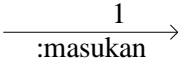
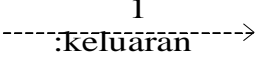
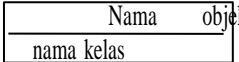

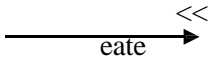
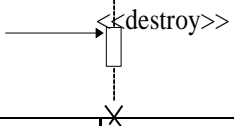

**Tabel 1 Simbol - Simbol Use Case Diagram**

Simbol	Keterangan
	<i>Use Case</i> menggambarkan bagaimana seseorang akan menggunakan/memanfaatkan sistem.
	Aktor seseorang/sesuatu yang berinteraksi dengan sistem yang sedang kita kembangkan.
	Asosiasi komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.
	<i>Include</i> memungkinkan suatu <i>usecase</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>usecase</i> yang lainnya.
	<i>Extend</i> memungkinkan <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh <i>usecase</i> yang lainnya.

## 2. Sequence Diagram

Menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan untuk pembuatan sequence diagram dapat dilihat pada tabel 3.2 sebagai berikut :

**Tabel 2 Simbol – Simbol Sequence Diagram**

Simbol	Keterangan
	Aktor adalah seseorang / sesuatu yang berinteraksi dengan sistem yang sedang kita kembangkan.
	Pesan tipe <i>call</i> menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
	Pesan tipe <i>send</i> menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya.
	Pesan tipe <i>return</i> menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.
	Objek menyatakan objek yang berinteraksi pesan
	Waktu aktif menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
	Pesan tipe <i>create</i> menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
	Pesan tipe <i>destroy</i> menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .
	Garis hidup/ <i>lifeline</i> menyatakan kehidupan suatu objek.

## 3. Activity Diagram

Teknik untuk menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Simbol-simbol yang digunakan untuk pembuatan activity diagram dapat dilihat pada tabel 3.3 sebagai berikut:

**Tabel 3 Simbol - Simbol Activity Diagram**

Simbol	Keterangan
●	Status Awal
○	Status Akhir
▭	Aktivitas
◇	Percabangan/decision: asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
▬	Penggabungan/join: asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

#### 4. Class Diagram

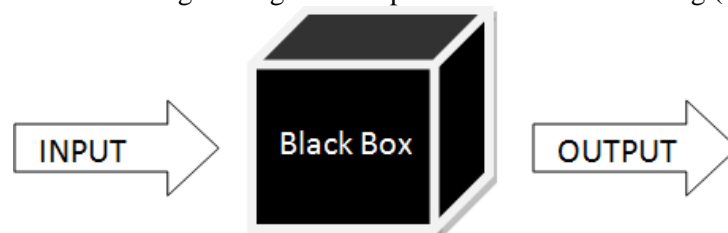
Pemodelan dari *structure diagrams*. Menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.



Gambar 2 Class Diagram

#### 2.3 Metode Pengujian Sistem

Menurut (Kusuma Dewi et al., 2022) Black Box Testing merupakan pengujian alternatif sebagai pelengkap setelah melakukan pengujian White Box Testing, dimana pengujian hanya berfokus pada spesifikasi kebutuhan fungsional saja dengan tujuan mengetahui dan memastikan apakah fungsi, input, serta output dari aplikasi tersebut sudah sesuai berdasarkan spesifikasi aplikasi pada awalnya. Black Box Testing mudah untuk diimplementasikan sebab memfokuskan pada spesifikasi dari sisi fungsional sebuah aplikasi dan tidak mementingkan bagaimana aplikasi tersebut dirancang (coding).



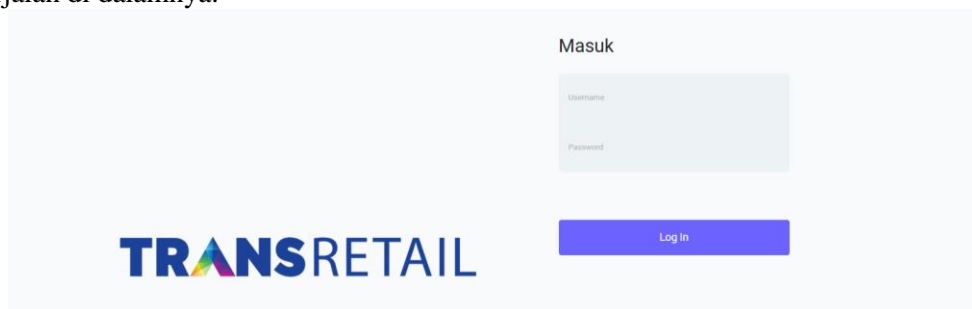
Gambar 3 Metode Pengujian Black Box

Tujuan Black Box Testing adalah untuk mencari kesalahan/kegagalan dalam operasi tingkat tinggi, yang mencakup kemampuan dari perangkat lunak, operasional/tata laksana, skenario pemakai.

- Tester tidak memerlukan Skill Programming yang tinggi
- Jenis Testing yang digunakan: User Acceptance Test, System Testing, UI Testing
- Melakukan pengujian berdasarkan apa yang dilihat, hanya fokus terhadap fungsionalitas dan output. Pengujian lebih ditujukan pada desain software sesuai standar dan reaksi apabila terdapat celah-celah bug/vulnerabilitas pada program aplikasi tersebut setelah dilakukan white box testing.

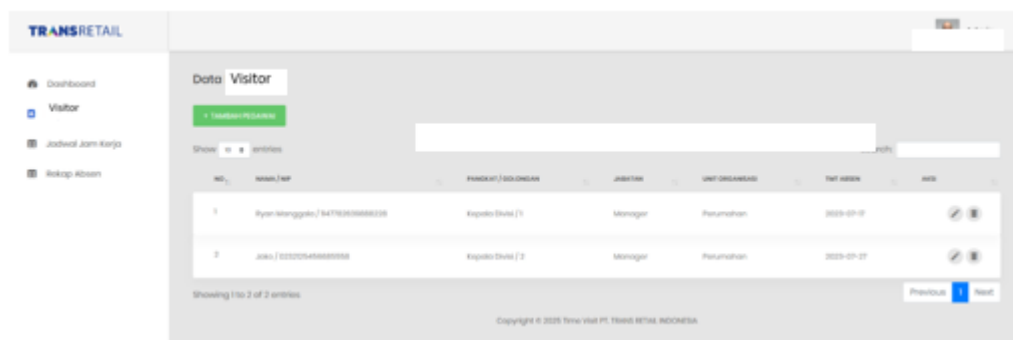
### 3 HASIL PENELITIAN

Tampilan Program pada gambar 4 tampilan program Login beserta fungsi fungsi yang berjalan di dalamnya.



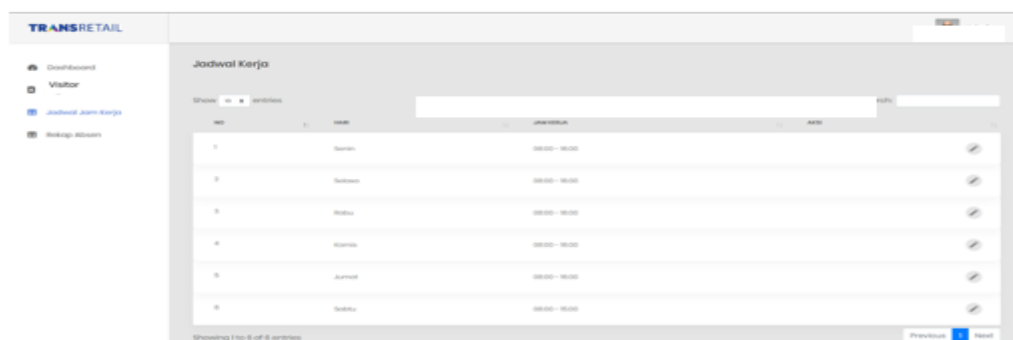
Gambar 4 Tampilan Login Utama

Tampilan Program pada gambar 5 tampilan program Data Visitor beserta fungsi fungsi yang berjalan di dalamnya.



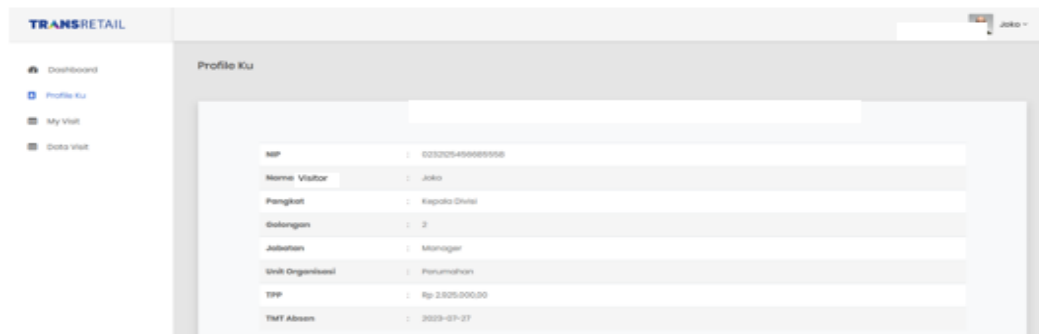
Gambar 5 Tampilan Data Visitor

Tampilan Program pada gambar 6 tampilan jadwal beserta fungsi fungsi yang berjalan didalamnya.



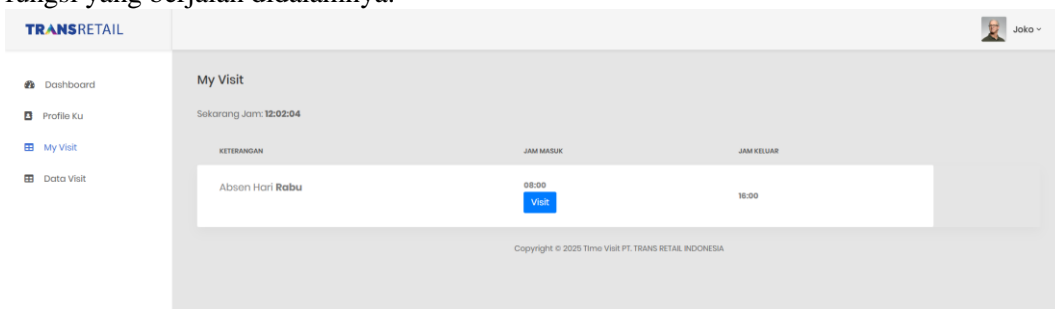
Gambar 6 Tampilan Jadwal

Tampilan Program pada gambar 7 tampilan profile ku beserta fungsi fungsi yang berjalan didalamnya.



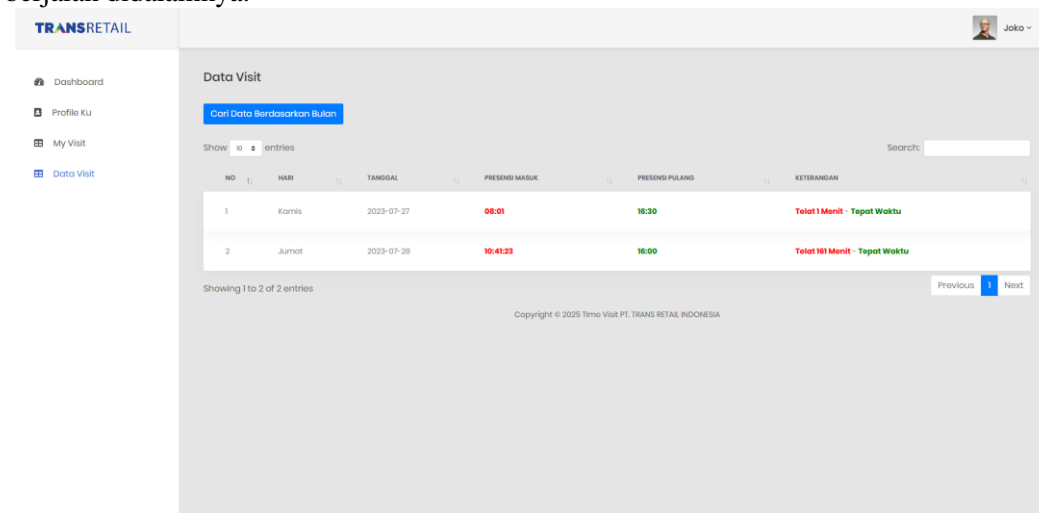
**Gambar 7 Tampilan profile ku**

Tampilan Program pada gambar 8 tampilan program Laporan My Visit beserta fungsi fungsi yang berjalan didalamnya.



**Gambar 8 Tampilan Absenku**

Tampilan Program pada gambar 9 tampilan Data visit ku beserta fungsi fungsi yang berjalan didalamnya.




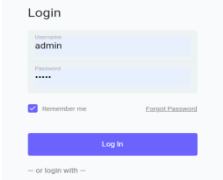

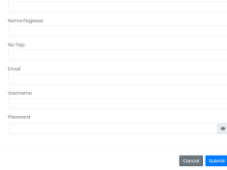

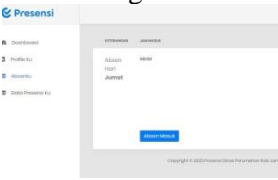

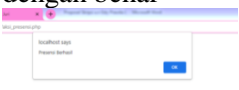
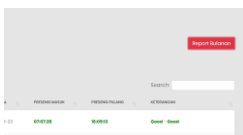
**Gambar 9 Tampilan Data Visit**


### 3.2 Pengujian Sistem



Pengujian yang digunakan adalah menggunakan blackbox. Adapun detailnya ditunjukkan pada Tabel 4 berikut ini:

Tabel 4 Tabel Pengujian

Fungsi yang diuji	Kondisi	Output yang diharapkan	Output yang dihasilkan	Sistem
Login Aplikasi 	Nama pengguna dan kata sandi benar 	Sukses masuk aplikasi	Sukses masuk aplikasi	ok
	Nama pengguna dan kata sandi salah maupun kosong 	Gagal masuk aplikasi	Gagal masuk aplikasi	ok
Tambah data Visitor 	Form diisi dengan benar 	Sukses tambah data	Sukses tambah data	ok
	Form diisi beberapa atau kosong 	Gagal Tambah data	Gagal Tambah data	ok
Ubah data Visitor 	Form absensi diisi dengan benar 	Sukses ubah data	Sukses ubah data	ok
Cetak data Siswa 	Data Visitor yang akan di cetak	Sukses cetak data Visitor	Sukses cetak data Visitor	ok

Menambah user 	Form Isi data username dan data diri	Sukses menambah user	Sukses menambah user	ok
--	--------------------------------------	----------------------	----------------------	----

Tabel 4 adalah tampilan dari tampilan Pengujian Blackbox yang melakukan pengecekan terhadap fungsi fungsi yang dilakukan aplikasi.

#### 4 PENUTUP

Berdasarkan penjelasan dan hasil dari bab-bab sebelumnya dapat disimpulkan sebagai berikut:

- Dalam Implementasi Time Visit Monitoring Pada PT Trans Retail Indonesia, proses pencarian dan penyajian informasi belum dapat dilakukan secara mudah dan cepat, tingkat kesalahan dalam proses pengolahan data masih sering terjadi dan dalam pembuatan Laporan memerlukan waktu yang lama.
- Di dalam sistem yang diusulkan telah menghasilkan suatu aplikasi program dengan kelebihan-kelebihannya sebagai berikut :
  - Memiliki fasilitas untuk menambah (*Entry*), memperbaiki (*Edit*), dan menghapus (*Delete*), data laporan yang tersimpan dalam data base.
  - Penyajian informasi dan Laporan dapat dengan cepat diperoleh sehingga memudahkan sistem informasi.

#### 5 DAFTAR PUSTAKA

- Adnan, F. (2020). Kunjungan Lapangan (Field Trip) Sebagai Metode Pembelajaran. *Annual Conference on Islamic Education and Thought (ACIET)*, 1(1), 94–104. <http://pkm.uika-bogor.ac.id/index.php/aciet/article/view/610>
- Alhabasi, M. T., & Haryono, W. (2023). Perancangan Sistem Informasi E-Book Kunjungan Kerja Di Victoria Busana Berbasis Website Menggunakan Metode Waterfall. *Journal Information & Computer*, 1(1), 70–81. <https://doi.org/10.32493/jicomisc.v1i1.26813>
- Ananda Muhamad Tri Utama. (2022). PERBANDINGAN MODEL WATERFALL DENGAN PROTOTYPE PADA PENGEMBANGAN SYSTEM INFORMASI BERBASIS WEBSITE. 9(08), 356–363.
- Badan Pengembangan dan Pembinaan Bahasa, K. P. dan K. R. I. (2016). Implementasi. <https://kbbi.kemdikbud.go.id/entri/Implementasi>
- Destriana, R., Husain, S. M., Handayani, N., & Siswanto, A. T. P. (2022). Diagram UML Dalam Membuat Aplikasi Android Firebase" Studi Kasus Aplikasi Bank Sampah". <https://books.google.com/books?hl=en&lr=&id=6bM-EQAAQBAJ&oi=fnd&pg=PP1&dq=diagram+uml+dalam+membuat+aplikasi+android&ots=Lqt0dVIZvq&sig=M3k3QbJrJmhQC0imgSxTDYSmoHo>
- Grafiani, C. P. (2021). Seni Manajemen Waktu: Rahasia Bagaimana Orang-Orang Sukses Mengatur Waktu Mereka. [https://books.google.com/books?hl=en&lr=&id=5ec\\_EAAAQBAJ&oi=fnd&pg=PR3&dq=pengertian+waktu+kunjungan&ots=UnxIvs6\\_39&sig=ieQtN97oojfKHkwwf4ypBolpeIw](https://books.google.com/books?hl=en&lr=&id=5ec_EAAAQBAJ&oi=fnd&pg=PR3&dq=pengertian+waktu+kunjungan&ots=UnxIvs6_39&sig=ieQtN97oojfKHkwwf4ypBolpeIw)
- Irawan, D., & Anshory, I. (2020). Implementasi Esp 32 Cam Dan Sensor Infrared Untuk Monitoring Pengunjung Dilokasi Wisata. *Seminar Nasional Fortei7-5 Forum*

- Pendidikan Tinggi Teknik Elektro Indonesia Regional VII, 95–99. <https://journal.fortei7.org/index.php/sinarFe7/article/view/369>
- Kusuma Dewi, E. H., Pratama, I. S., Putera, A. S., & Carudin, C. (2022). Black Box Testing pada Aplikasi Pencatatan Peminjaman Buku Menggunakan Boundary Value Analysis. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 6(3), 315. <https://doi.org/10.30998/string.v6i3.11958>
- Minasari, S., Azam, M., & Rahayu, S. R. (2022). Pengembangan Aplikasi Monitoring Penyakit Hipertensi dan Diabetes Mellitus Terintegrasi. *Jurnal Manajemen Kesehatan Indonesia*, 10(1), 28–42. <https://doi.org/10.14710/jmki.10.1.2022.28-42>
- Pane, S. F., Zamzam, M., & Fadillah, M. D. (2020). Membangun Aplikasi Peminjaman Jurnal Menggunakan Aplikasi Oracle Apex Online. [books.google.com. https://books.google.com/books?hl=en&lr=&id=sdPXDwAAQBAJ&oi=fnd&pg=PR16&dq=membangun+aplikasi+peminjaman+jurnal+menggunakan+aplikasi+oracle+apex+online&ots=23GdNnr31J&sig=E8YIdCk3I4D00-LU8VYzF\\_k2U0U](https://books.google.com/books?hl=en&lr=&id=sdPXDwAAQBAJ&oi=fnd&pg=PR16&dq=membangun+aplikasi+peminjaman+jurnal+menggunakan+aplikasi+oracle+apex+online&ots=23GdNnr31J&sig=E8YIdCk3I4D00-LU8VYzF_k2U0U)
- Santoso, H. (2023). Mencegah Pembiayaan Bermasalah Dengan Monitoring Dan Pengawasan Prespektif Vaithzal Rivai. *Jurnal Ekonomi Dan Bisnis Syariahe- Issn: 2809-3224*, 2(September), 11–39.