

**PERANCANGAN *DASHBOARD* INTERAKTIF UNTUK APLIKASI
PEMANTAUAN KEDISIPLINAN KARYAWAN DI PT XYZ**

***DESIGNING AN INTERACTIVE DASHBOARD FOR EMPLOYEE DISCIPLINE
MONITORING APPLICATION AT PT XYZ***

Gregorius Farrel Yoewono¹, Tri Sutrisno², Irvan Lewenusa³

Sistem Informasi, Fakultas Teknologi Informasi, Universitas Tarumanagara^{1,2}

Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara³

E-mail: gregorius.825210048@stu.untar.ac.id¹, tris@fti.untar.ac.id²,

irvanl@fti.untar.ac.id³

ABSTRACT

In the ever-growing digital era, PT XYZ faces challenges in monitoring employee discipline in real-time. The problem that arises is the limitations of the company's management system that does not have adequate data visualization tools, so that employee discipline information is difficult to access quickly by various departments. This research aims to design an interactive dashboard using Grafana to help visualize employee discipline data. The development method used is Solo Scrum, which includes the stages of requirement identification, backlog creation, sprint, and deployment. The results showed that the designed dashboard can display real-time data related to employee attendance and punctuality, and reduce the potential for human error in data processing. In conclusion, this dashboard makes it easier for PT XYZ to monitor employee discipline and support strategic decision making.

Keywords: *Dashboard, Employee Discipline, Solo Scrum, Grafana.*

ABSTRAK

Dalam era digital yang terus berkembang, PT XYZ menghadapi tantangan dalam memantau kedisiplinan karyawan secara *real-time*. Permasalahan yang muncul adalah keterbatasan sistem manajemen perusahaan yang tidak memiliki alat visualisasi data yang memadai, sehingga informasi disiplin karyawan sulit diakses secara cepat oleh berbagai departemen. Penelitian ini bertujuan merancang *dashboard* interaktif menggunakan Grafana untuk membantu memvisualisasikan data kedisiplinan karyawan. Metode pengembangan yang digunakan adalah *Solo Scrum*, yang meliputi tahap identifikasi kebutuhan, pembuatan *backlog*, *sprint*, dan *deployment*. Hasil penelitian menunjukkan bahwa *dashboard* yang dirancang dapat menampilkan data *real-time* terkait kehadiran dan ketepatan waktu karyawan, serta mengurangi potensi kesalahan manusia dalam pengolahan data. Kesimpulannya, *dashboard* ini memudahkan PT XYZ dalam memantau kedisiplinan karyawan dan mendukung pengambilan keputusan strategis.

Kata kunci: *Dashboard, Kedisiplinan Karyawan, Solo Scrum, Grafana.*

PENDAHULUAN

Dalam era digital yang berkembang pesat, perusahaan di berbagai industri harus terus meningkatkan efisiensi dan produktivitas mereka untuk tetap kompetitif. Salah satu cara untuk mencapai hal ini adalah melalui penggunaan teknologi informasi, terutama dalam pemantauan dan penilaian kedisiplinan karyawan. Teknologi informasi memungkinkan pengumpulan, analisis, dan visualisasi data secara *real-time*, memfasilitasi pengambilan keputusan yang lebih cepat dan akurat. Dalam hal ini, *dashboard* merupakan salah satu alat yang efektif untuk menampilkan tampilan visual dari informasi dalam satu layar [1]. Penelitian ini melibatkan PT XYZ, yang merupakan suatu perusahaan yang telah menggunakan aplikasi manajemen dengan berbagai fitur untuk mendukung operasional sehari-hari. Aplikasi ini dapat mengelola akses menggunakan kartu, memantau pengunjung, serta mencatat riwayat kejadian seperti absensi karyawan. Namun, aplikasi ini sendiri tidak memiliki alat visualisasi data, dan tidak ada administrasi penggunaan yang jelas sehingga tidak ada fleksibilitas untuk digunakan di departemen lain. Permasalahan yang timbul adalah penyampaian informasi yang tidak *real-time* dan menambah beban operator sehingga dapat terjadi *human error*, ini terjadi karena hanya satu perangkat dengan satu operator yang dapat melihat data secara *real-time*, sedangkan dibutuhkan waktu untuk memproses dan meneruskannya ke departemen lain.

Menyadari keterbatasan yang ada pada aplikasi manajemen saat ini, diperlukan suatu solusi yang mampu mengatasi masalah penyampaian informasi secara *real-time* dan

mengurangi potensi kesalahan manusia. Untuk mengatasi keterbatasan ini, maka *dashboard* interaktif akan diintegrasikan ke dalam sistem manajemen yang sudah ada. *Dashboard* ini tidak hanya akan memungkinkan visualisasi data yang lebih jelas dan mudah dipahami oleh berbagai departemen, tetapi juga memungkinkan akses data secara langsung tanpa perlu melalui banyak tahapan manual. Penelitian ini bertujuan untuk merancang *dashboard* yang mampu memvisualisasikan data yang tersimpan dalam *database* aplikasi manajemen perusahaan, baik data *real-time* atau data historis, sehingga dapat membantu manajemen dalam memantau kedisiplinan karyawan dan membuat keputusan yang lebih strategis. Teknologi yang akan digunakan untuk perancangan *dashboard* ini menggunakan Grafana untuk pengembangan *dashboard*. Dalam penelitian yang berjudul “*Data visualization and monitoring with Grafana and Prometheus*” (2021) oleh T Leppänen, disebutkan bahwa pemakaian Grafana memiliki banyak keuntungan, seperti dukungan banyak sumber data, terdapat *Query Editor* untuk semua sumber data tersebut, panel dapat menggabungkan sumber data berbeda, kemampuan *import* dan *export dashboard*, *render* panel sebagai gambar PNG, dan komunitas yang menawarkan banyak dokumentasi, panduan, dan video tentang perangkat lunak mereka dan penggunaannya [2].

METODE

Metode Perancangan

Metode yang digunakan untuk perancangan *dashboard* ini menggunakan metode *Solo Scrum*, *Solo Scrum* adalah adaptasi dari *Scrum* dan *PSP (Personal Software Process)* untuk pengembangan perangkat lunak secara

mandiri. Alur kerja dimulai dengan pembuatan *backlog* produk, yang berisi semua tugas yang diperlukan untuk mengimplementasikan fitur-fitur yang diinginkan oleh pengguna. Dari kelompok tugas ini, beberapa dipilih untuk dikembangkan selama *sprint*. Proses *sprint* ini bersifat iteratif hingga tanggal penyelesaian proyek tiba, atau hingga semua tugas selesai. Selama *sprint*, perlu ada aktivitas manajemen yang bertujuan untuk merencanakan dan memantau pengembangan proyek. Setelah menyelesaikan tugas-tugas, proyek tersebut diserahkan, yang akan ditinjau oleh kelompok evaluasi dan, pada akhirnya, pertemuan orientasi akan diadakan dengan orang-orang yang akan menggunakan produk yang dikembangkan [3]. Tahapan pada perancangannya adalah:

1. *Definition of Requirements*

Dilakukan identifikasi kebutuhan *user* dan menyusun *backlog*. Identifikasi kebutuhan ini dimulai dari wawancara dengan user perusahaan untuk mendapatkan *KPI* yang akan dibutuhkan, lalu akan dilakukan penggambaran umum fitur dan jenis tampilan dari *dashboard*. Kemudian diurutkan sesuai dengan kepentingan fitur. Selain mengidentifikasi kebutuhan dan penyusunan *backlog*. Grafana OSS dipilih sebagai alat visualisasi data. Grafana OSS adalah perangkat lunak *open source* untuk visualisasi dan analitik data. Ini memungkinkan pengguna untuk membuat, memvisualisasikan, memberi peringatan, dan menjelajahi metrik, *log*, dan jejak data pengguna, di mana

pun mereka disimpan. Grafana menyediakan alat untuk mengubah basis data deret waktu menjadi grafik dan visualisasi yang mendalam [4]. Selain itu Grafana juga dipilih karena berbagai fiturnya yang cocok dengan permasalahan yang dihadapi, yaitu Grafana dapat digunakan dari berbagai perangkat dan memiliki administrasi pengguna yang jelas.

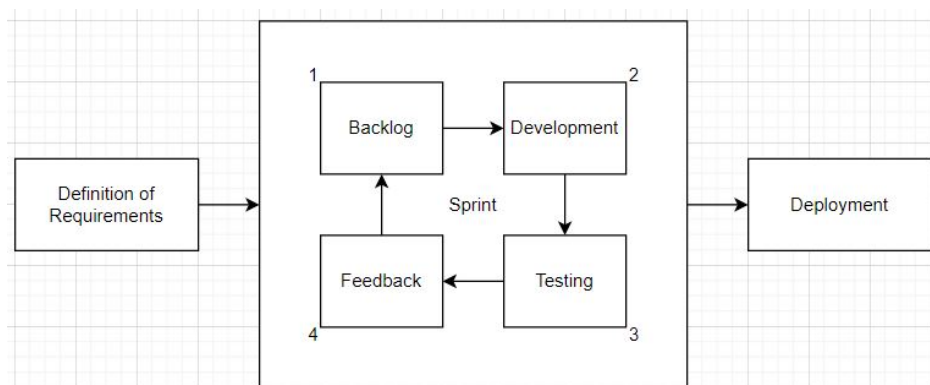
2. *Sprint*

Pada tahap ini, produk-produk *backlog* dikerjakan. *Sprint* dipecah menjadi beberapa tugas yang lebih kecil dan dapat diselesaikan dalam satu *sprint*. Kemudian *sprint* dikerjakan satu per satu. Pada akhir dari setiap *sprint* akan dilakukan *user acceptance test* dari hasil *sprint*, jika ada kendala maka akan dijadwalkan *sprint* baru. Sebagian besar dari tahap ini adalah pengerjaan panel untuk visualisasi data, setiap panel menggunakan bahasa pemrograman *Transact-SQL (T-SQL)*, yang merupakan bagian dari bahasa *SQL*. Perbedaannya dengan bahasa lain adalah *SQL* dapat menangani banyak baris dari satu atau lebih tabel dengan satu *statement* [5].

3. *Deployment*

Setelah seluruh *sprint* dan *testing* oleh penulis telah dilakukan, lalu *user acceptance test* sudah lolos. Penulis meluncurkan produk akhirnya ke pengguna secara sekaligus dan siap digunakan oleh pengguna.

Tahapan dalam Metode *Solo Scrum* dapat dilihat pada **Gambar 1**.



Gambar 1. Tahapan Penelitian Metode *Solo Scrum*

Metode Pengumpulan Data

Metode pengumpulan data yang digunakan adalah pengumpulan data sekunder. Data sekunder, atau “data yang ada”, adalah data yang dikumpulkan oleh orang lain untuk digunakan dalam menjawab pertanyaan penelitian, ini adalah praktik yang umum dan diterima dalam dunia penelitian dan akademis [6]. Data sekunder dapat berupa survey berskala besar atau data yang dikumpulkan sebagai bagian dari penelitian pribadi [7]. Perlu diingat bahwa semua data sekunder merupakan data primer pada suatu waktu [6]. Data didapat dari perusahaan dan diekstraksi ke dalam *SQL Server Management Studio 19*.

Metode Uji KPI

Key Performance Indicator (KPI) adalah indikator yang fokus pada aspek-aspek kinerja organisasi yang paling krusial untuk keberhasilan organisasi saat ini dan di masa depan [8]. Metode pengujian *KPI* yang digunakan adalah metode *SMART Criteria*. *KPI* yang didapatkan dari hasil wawancara dengan *user* PT XYZ akan kembali disesuaikan dengan kriteria *SMART*, *SMART* seringkali digunakan untuk menyediakan referensi yang mendeterminasikan kualitas dari metrik, *SMART* terdiri dari *Specific*,

Measurable, *Attainable*, *Realistic*, dan *Timely*. Kriterianya dapat dijelaskan sebagai berikut: [9].

Specific : Jelas dan fokus untuk mencegah misinterpretasi.

Measurable : Bisa dikuantifikasikan dan dibandingkan dengan data lain

Attainable : Dapat diraih, masuk akal, dan kredibel dalam kondisi yang diharapkan

Realistic : Sesuai dengan batasan organisasi dan *cost-effective*

Timely : Dapat diraih dalam waktu yang diberikan

HASIL DAN PEMBAHASAN

Definition of Requirements

Pada tahap ini, dilakukan identifikasi seluruh persyaratan dan data yang akan digunakan pada perancangan *dashboard*. *User* dari perusahaan terlibat dalam melakukan wawancara untuk merancang *KPI* dan pengumpulan data. Persyaratan ini digunakan sebagai landasan dasar pembentukan panel visualisasi data, panel ini akan disusun kedalam *backlog*, yang akan dikerjakan dalam tahap berikutnya. Berdasarkan hasil wawancara, *KPI* diidentifikasi dan diuji menggunakan *SMART Criteria*. Hasilnya ada pada **Tabel 1**

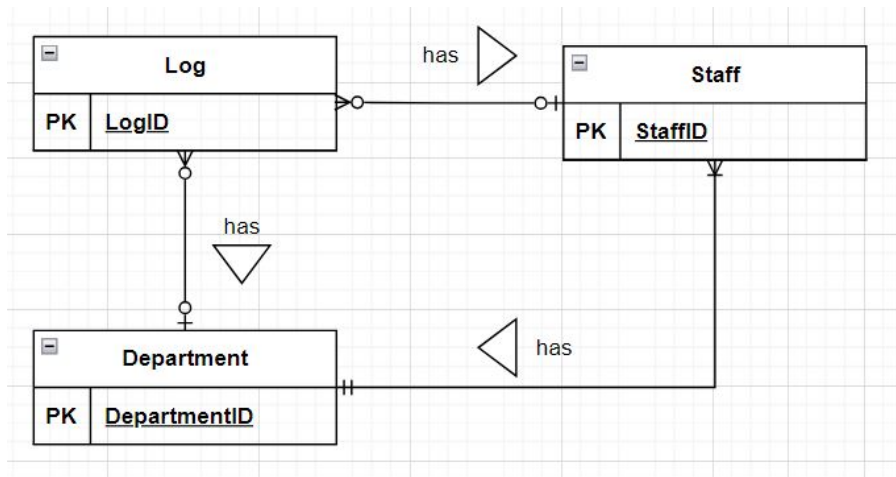
Tabel 1. Tabel KPI

NO	Key Performance Indicator	SMART Criteria
1	Tingkat Kehadiran	<p><i>Specific:</i> Presentase kehadiran karyawan yang hadir di tempat kerja</p> <p><i>Measurable:</i> Mengukur kehadiran setiap periode waktu tertentu</p> <p><i>Attainable:</i> Target 90% kehadiran per bulan</p> <p><i>Realistic:</i> Mempertimbangkan kebijakan yang ada, target ini sesuai dengan kapasitas perusahaan</p> <p><i>Timely:</i> Data akan diukur dan dilaporkan sesuai dengan periode waktu yang ditentukan</p>
2	Tingkat Ketepatan Waktu	<p><i>Specific:</i> Persentase karyawan yang tiba tepat waktu di kantor.</p> <p><i>Measurable:</i> Mengukur jumlah karyawan yang datang tepat waktu dibandingkan dengan jumlah karyawan yang telat</p> <p><i>Attainable:</i> Target 90% karyawan datang tepat waktu.</p> <p><i>Realistic:</i> Dengan melakukan identifikasi permasalahan, maka solusi dapat diberikan sesuai kebutuhan masing-masing karyawan</p> <p><i>Timely:</i> Data akan diukur dan dilaporkan sesuai dengan periode waktu yang ditentukan</p>
3	Waktu Rata-rata Masuk	<p><i>Specific:</i> Rata-rata waktu masuk karyawan ke tempat kerja.</p> <p><i>Measurable:</i> Menghitung waktu masuk karyawan dalam jam.</p> <p><i>Attainable:</i> Target waktu rata-rata masuk tidak lebih dari 09:00</p> <p><i>Realistic:</i> Mengingat kebiasaan saat ini, target ini dapat dicapai.</p> <p><i>Timely:</i> Data akan diukur dan dilaporkan sesuai dengan periode waktu yang ditentukan</p>
4	Waktu Rata-rata Keluar	<p><i>Specific:</i> Rata-rata waktu keluar karyawan dari tempat kerja.</p> <p><i>Measurable:</i> Menghitung waktu keluar karyawan dalam jam.</p> <p><i>Attainable:</i> Target waktu rata-rata keluar tidak lebih dari 18:30.</p> <p><i>Realistic:</i> Dengan peningkatan manajemen waktu, ini adalah target yang realistis.</p> <p><i>Timely:</i> Data akan diukur dan dilaporkan sesuai dengan periode waktu yang ditentukan</p>
5	Jam Puncak Karyawan Masuk	<p><i>Specific:</i> Identifikasi jam sibuk karyawan masuk kerja.</p>

		<p><i>Measurable</i>: Mencatat waktu masuk karyawan selama periode waktu tertentu</p> <p><i>Attainable</i>: Mengurangi kemacetan hingga 20% pada jam sibuk.</p> <p><i>Realistic</i>: Dengan penyesuaian jadwal, target ini dapat dicapai.</p> <p><i>Timely</i>: Data akan diukur dan dilaporkan sesuai dengan periode waktu yang ditentukan</p>
6	Jam Puncak Karyawan Keluar	<p><i>Specific</i>: Identifikasi jam sibuk karyawan keluar dari kerja.</p> <p><i>Measurable</i>: Mencatat waktu keluar karyawan selama periode waktu tertentu</p> <p><i>Attainable</i>: Mengurangi kemacetan hingga 20% pada jam sibuk.</p> <p><i>Realistic</i>: Dengan penyesuaian proses, target ini dapat dicapai</p> <p><i>Timely</i>: Data akan diukur dan dilaporkan sesuai dengan periode waktu yang ditentukan</p>
7	Tingkat Akses Masuk Gagal	<p><i>Specific</i>: Persentase kegagalan akses masuk sistem oleh karyawan.</p> <p><i>Measurable</i>: Menghitung jumlah akses gagal dibandingkan dengan total percobaan akses.</p> <p><i>Attainable</i>: Target kurang dari 2% akses gagal</p> <p><i>Realistic</i>: Dengan perbaikan sistem, target ini dapat dicapai.</p> <p><i>Timely</i>: Data akan diukur dan dilaporkan sesuai dengan periode waktu yang ditentukan</p>

Setelah *KPI* ditentukan, data sekunder digunakan untuk membantu pengembangan *dashboard*. Data ini merupakan data buatan yang digunakan untuk visualisasi struktur database yang akan digunakan. Data divisualisasikan dalam bentuk *conceptual data model* dan *logical data model*. *Conceptual Data Model* adalah representasi dari data

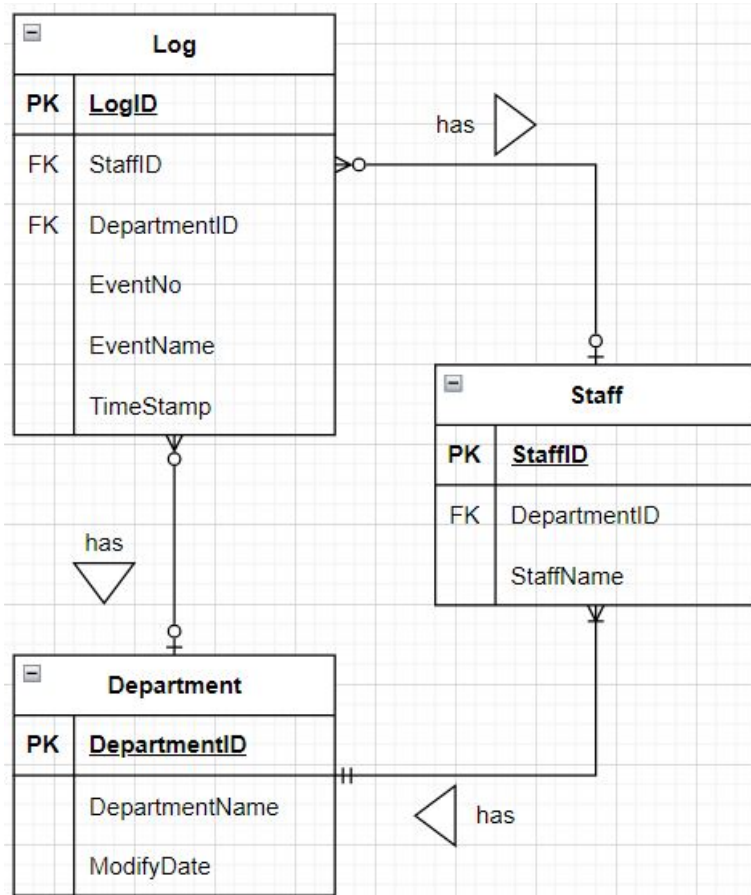
organisasi sedangkan *Logical Data Model* adalah versi basis data dari pola, komponen, dan aplikasi siap pakai [10]. Model data konseptual dapat dilihat pada **Gambar 2** dan tabel identifikasi dapat dilihat pada **Tabel 2**, Model data logis dapat dilihat pada **Gambar 3** dan tabel spesifikasinya dapat dilihat pada **Tabel 3**.



Gambar 2. Model Data Konseptual

Tabel 2. Tabel Identifikasi Entitas

Entity Name	Description	Occurence
Log	Informasi kejadian-kejadian yang dialami aplikasi manajemen	Data muncul saat aplikasi manajemen mengalami kejadian-kejadian
Staff	Informasi pegawai dan pengunjung yang aktif	Data muncul sesuai dengan pegawai dan pengunjung aktif yang ada di PT XYZ
Department	Informasi departemen didalam PT XYZ	Data muncul dari daftar departemen yang ada di PT XYZ



Gambar 3. Model Data Logis

Tabel 3. Tabel Spesifikasi

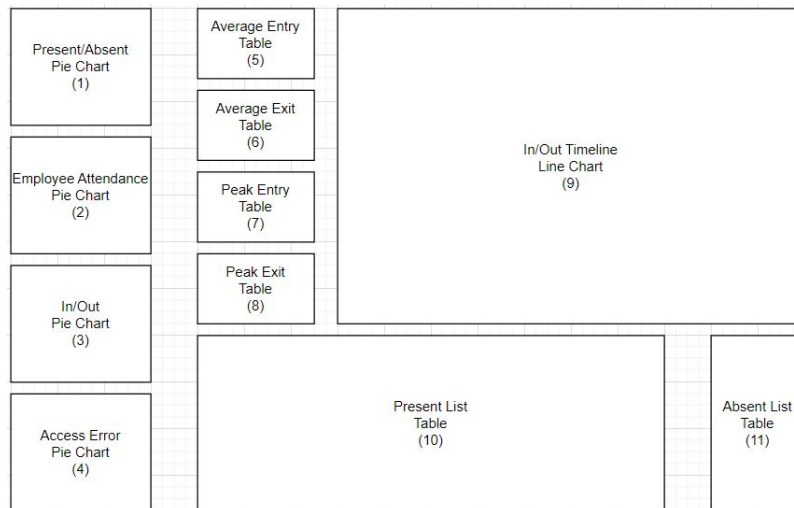
Nama Entitas	Atribut	Deskripsi	Jenis Data dan Ukurannya	Nulls	Multi-Valued	Domains
Log	LogID	Mengidentifikasi setiap kejadian	uniqueidentifier	No	No	No
	EventNo	Kode kejadian	Smallint	No	No	No
	EventName	Deskripsi kejadian	Varchar(40)	No	No	No
	StaffID	ID karyawan atau pengunjung	Nvarchar(20)	Yes	No	No
	DepartmentID	Departemen dari karyawan atau pengunjung	Nvarchar(50)	Yes	No	No
	TimeStamp	Tanggal dan waktu aplikasi mencatat kejadian	Datetime	No	No	No
Staff	StaffID	ID karyawan atau pengunjung	Nvarchar(20)	No	No	No
	StaffName	Nama staf yang ditampilkan	Nvarchar(100)	No	No	No

	DepartmentID	Departemen dari karyawan atau pengunjung	Nvarchar(50)	No	No	No
Department	DepartmentID	Departemen dari karyawan atau pengunjung	Nvarchar(50)	No	No	No
	DepartmentName	Nama Departemen	Nvarchar(50)	No	No	No
	ModifyDate	Tanggal modifikasi informasi departemen	Nvarchar(50)	No	No	No

Sprint

Tahap ini dimulai dengan mengidentifikasi *backlog* apa yang akan dikerjakan, lalu pengerjaan *backlog* dilakukan, setelah itu user perusahaan akan melakukan *user acceptance test* untuk produk dari *backlog* tersebut,

kemudian *user* akan menerima atau memberi *feedback* yang akan dimasukkan kembali ke dalam *backlog*. Rancangan *mockup dashboard* dapat dilihat pada **Gambar 4**, penjelasan dari setiap panel *dashboard* dapat dilihat pada **Tabel 4**.



Gambar 4. Mockup Dashboard

Tabel 4. Tabel Panel Dashboard

No	Nama	Jenis	KPI	Deskripsi
1	<i>Present/Absent</i>	<i>Pie Chart</i>	Tingkat Kehadiran	Persentase karyawan yang hadir dan absen
2	<i>Employee Attendance</i>	<i>Pie Chart</i>	Tingkat Ketepatan Waktu	Persentase karyawan yang tepat waktu dan telat
3	<i>In/Out</i>	<i>Pie Chart</i>		Persentase karyawan yang didalam dan diluar area kerja
4	<i>Access Error</i>	<i>Pie Chart</i>	Tingkat Akses Masuk Gagal	Persentase kejadian akses masuk berhasil dan gagal

5	<i>Average Entry</i>	<i>Table</i>	Waktu Rata-rata Masuk	Waktu rata-rata karyawan melakukan absensi masuk
7	<i>Peak Entry</i>	<i>Table</i>	Jam Puncak Karyawan Masuk	Waktu puncak karyawan masuk area kantor
8	<i>Peak Exit</i>	<i>Table</i>	Jam Puncak Karyawan Keluar	Waktu puncak karyawan keluar area kantor
9	<i>In/Out Timeline</i>	<i>Line Chart</i>	Waktu Rata-rata Masuk, Waktu Rata-rata Keluar, Jam Puncak Karyawan Masuk, Jam Puncak Karyawan Keluar	Perbandingan frekuensi karyawan masuk dan keluar dari waktu ke waktu
10	<i>Present List</i>	<i>Table</i>		Daftar informasi karyawan yang hadir, informasi terdiri dari nama, jam pertama masuk, jam terakhir keluar, status ketepatan waktu absensi, dan jam kerja.
11	<i>Absent List</i>	<i>Table</i>		Daftar nama karyawan yang tidak hadir

Sprint dilakukan dengan jumlah 5 *sprint*, dengan pencapaian sebagai berikut:

Sprint 1:

- Pengerjaan Panel No. 1,2,3, dan 4
- Panel 1, 2, 3, dan 4 (Diterima)

Sprint 2:

- Pengerjaan Panel No. 5,6,7, dan 8
- Panel 5, 6, 7, dan 8 (Diterima)

Sprint 3:

- Pengerjaan Panel No. 9
- Panel 9 (Ditolak dengan alasan persempit pengelompokan data dari per 1 jam menjadi per 30 menit)

Sprint 4:

- Pengerjaan Panel No. 10 dan 11
- Panel 10 (Ditolak dengan alasan menambahkan kolom yang menandakan jam kerja per hari)

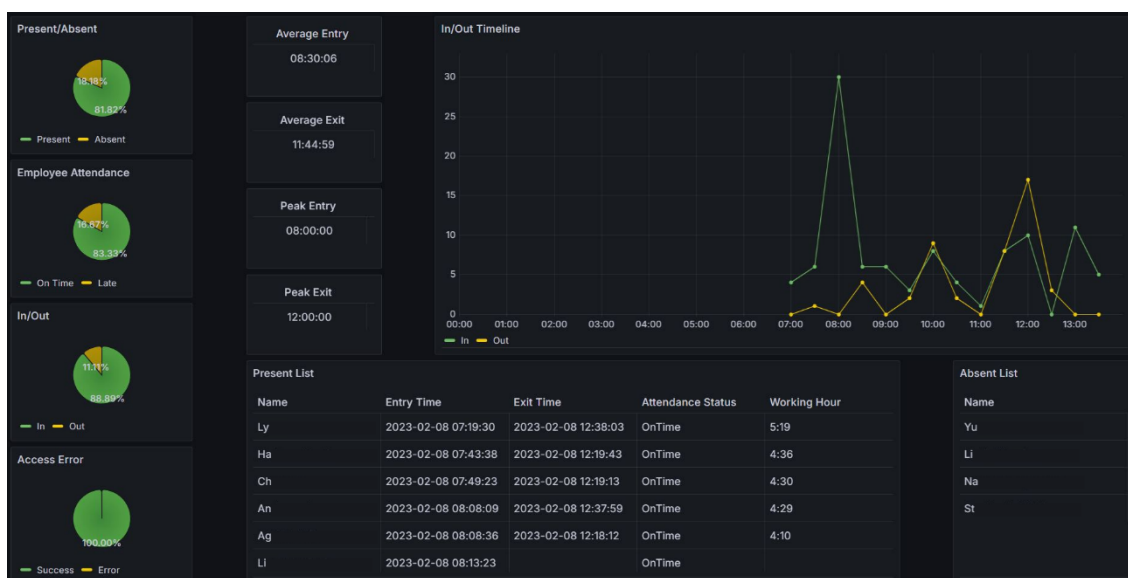
- Panel 11 (Diterima)

Sprint 5:

- Pengerjaan Panel No. 9 dan 10
- Panel 9 dan 10 (Diterima)

Deployment

Setelah tahap *sprint* selesai, maka *dashboard* diluncurkan dalam lingkungan perusahaan, hasil *dashboard* dapat dilihat pada **Gambar 5**, dimana *dashboard* sedang menampilkan data per hari dari jam 00:00:00 hingga 13:30:00 saat gambar *dashboard* diambil. *Dashboard* akan mulai kembali dari 0 saat berganti hari.

Gambar 5. Tampilan *Dashboard*

SIMPULAN DAN SARAN

Penelitian perancangan *dashboard* untuk PT XYZ ini telah berhasil membuat *dashboard* berdasarkan identifikasi *KPI* yang diuji dengan *SMART Criteria*, identifikasi entitas data, dan berbagai masukan dari *user* selama perancangan. Dengan *dashboard* ini, berbagai departemen PT XYZ dapat memantau berbagai data absensi yang berkaitan dengan kedisiplinan karyawannya. Karena perancangan ini menggunakan metode *Solo Scrum*, maka pengembangan ini akan mudah dipahami dan direncanakan oleh pihak lain karena dapat hanya dikerjakan oleh satu orang, perencanaan pengembangan untuk lebih dari satu orang juga akan mudah karena salah satu metode yang diadopsi oleh *Solo Scrum* adalah *Agile*. Perubahan *dashboard* sendiri juga mudah dilakukan karena hanya perlu merubah *query T-SQL* yang ada pada setiap panel, sehingga berbagai evaluasi dan perubahan dapat dilakukan secara mandiri.

Berdasarkan temuan yang diperoleh dalam penelitian ini, terdapat

beberapa saran yang dapat dipertimbangkan untuk penelitian lebih lanjut terkait dengan metode perancangannya. Untuk data yang memerlukan transformasi, maka pengetahuan tentang bahasa pemrograman yang berkaitan dengan basis data harus dimiliki agar dapat menampilkan data yang akurat, visualisasi data juga bergantung pada koneksi antara Grafana dengan sumber data sehingga diperlukan cara untuk memastikan koneksi data yang kuat.

DAFTAR PUSTAKA

- [1] M. Ropianto, A. R. Basar, and D. Hermanto, "Perancangan Dashboard Sebagai Sistem Informasi Di Dinas Perumahan Rakyat, Pemukiman Dan Pertamanan Kota Batam," *JR: Jurnal Responsive Teknik Informatika*, vol. 2, no. 02, pp. 123-133, 2018
- [2] T. Leppänen, "Data visualization and monitoring with Grafana and Prometheus," 2021.

- [3] J. N. Brito, C. Rebelo, and M. A. Brito, "*Scrum solo application in a project with a strong integration component*," 2020.
- [4] Grafana Labs, "Grafana Open Source," *Grafana Documentation*, [Online]. Tersedia: <https://grafana.com/docs/grafana/latest/fundamentals/>. Diakses pada: 31 Oktober 2024.
- [5] D. Petkovic, *Microsoft SQL Server 2019: A Beginner's Guide*, 7th ed. New York: McGraw-Hill, 2020, pp. 1-800. ISBN: 9781260458879.
- [6] D. C. Watkins, *Secondary Data in Mixed Methods Research*, 1st ed. London, UK: SAGE Publications, 2022.
- [7] J. P. Tripathy, "*Secondary data analysis: Ethical issues and challenges*," **Iranian Journal of Public Health**, vol. 42, no. 12, pp. 1478, 2013.
- [8] D. Parmenter, *Key Performance Indicators: Developing, Implementing, and Using Winning KPIs*, 4th ed. Hoboken, NJ, USA: Wiley, 2020.
- [9] Parida and U. Kumar, "*Maintenance performance measurement (MPM): issues and challenges*," *Journal of Quality in Maintenance Engineering*, vol. 12, no. 3, pp. 239-251, 2006.
- [10] J. S. Valacich, J. F. George, and J. A. Hoffer, *Modern Systems Analysis and Design*, 9th ed. Pearson, 2020.