

Perancangan Sistem Informasi Manajemen *Work Report* Rumah BUMN Kota Bandung Berbasis Website Menggunakan Metode *Rapid Application Development*

1st Muhammad Aqli Najib
Fakultas Rekayasa Industri
Universitas Telkom Bandung, Indonesia
bokliukra@student.telkomuniversity.ac.id

2nd Amelia Kurniawati
Fakultas Rekayasa Industri
Universitas Telkom Bandung, Indonesia
ameliakurniawati@telkomuniversity.ac.id

3rd Isnaeni Yuli Arini
Fakultas Rekayasa Industri
Universitas Telkom Bandung, Indonesia
isnaeniya@telkomuniversity.ac.id

Abstrak— Usaha Mikro Kecil dan Menengah (UMKM) memainkan peran vital dalam pertumbuhan ekonomi nasional. Rumah Badan Usaha Milik Negara (BUMN) Bandung di bawah binaan BRI, menjadi fasilitator krusial bagi UMKM di Kota Bandung. Namun, kurangnya kesadaran pegawai intern dalam memasukan laporan kerja harian yang harus diisi oleh pegawai intern menyebabkan kurangnya work report yang masuk ke dalam database rumah BUMN, yang dapat mengganggu kegiatan di Rumah BUMN. Tugas akhir ini bertujuan untuk meningkatkan efektivitas pengelolaan work report.

Dalam rangka mendukung pengelolaan work report, perancangan tugas akhir ini berfokus pada pengembangan sistem work report. Metode yang digunakan adalah Rapid Application Development (RAD). Metode ini memiliki empat tahapan yaitu analisis kebutuhan, perencanaan desain, implementasi, dan evaluasi secara berulang untuk memastikan responsivitas terhadap perubahan kebutuhan.

Hasil dari perancangan sistem ini adalah suatu aplikasi pengelolaan yang memiliki fokus pada kegiatan work report di Rumah BUMN. Sistem ini dapat membantu proses pengelolaan work report, serta memudahkan dalam pelacakan KPI dan SOP. Implementasi sistem ini menunjukkan peningkatan efisiensi dan akurasi dalam pengelolaan work report, memberikan solusi yang efektif dalam mengatasi permasalahan pengelolaan di Rumah BUMN.

Dengan adanya rancangan sistem pengelolaan work report, kemampuan dan efisiensi SDM Rumah BUMN dalam mengelola work report dan mencapai KPI meningkat. Sistem yang dirancang tidak hanya membantu dalam pengelolaan tetapi juga memberikan dampak positif terhadap manajemen SDM secara keseluruhan, memastikan bahwa kegiatan di Rumah BUMN dapat berjalan lebih lancar dan terstruktur. Kata kunci— SDM, Rumah BUMN, Work Report, Monitoring, Rapid Application Development (RAD)

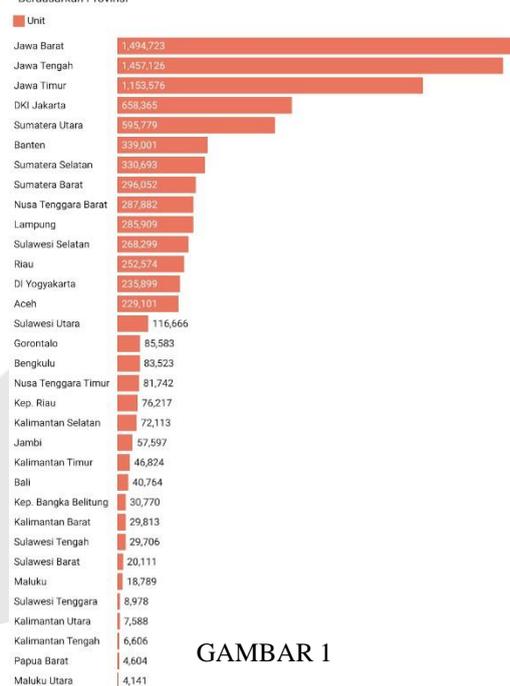
I. PENDAHULUAN

Usaha Mikro Kecil dan Menengah (UMKM) memiliki peran yang besar terhadap pertumbuhan perekonomian nasional (Purwana, Rahmi, & Aditya, 2017). UMKM merupakan salah satu faktor yang penting dalam pertumbuhan perekonomian daerah maupun perekonomian suatu negara. Pada tahun 2022, jumlah UMKM yang tersebar di seluruh

provinsi Indonesia menunjukkan perbedaan total unit UMKM di masing-masing provinsi, dengan total UMKM di beberapa daerah mencapai jutaan unit, seperti yang terlihat pada Gambar 1. (ukmindonesia.id)

Jumlah UMKM di Indonesia Sepanjang 2022

*Berdasarkan Provinsi



GAMBAR 1

Berdasarkan Gambar 1 dapat diketahui bahwa provinsi Jawa Barat memiliki jumlah UMKM paling banyak dari provinsi lain yang ada di Indonesia dengan jumlah 1.494.723 Juta UMKM. Hal ini tentunya menunjukkan bahwa Provinsi Jawa Barat memiliki potensi dan tantangan yang lebih kompleks dalam pengembangan UMKM. Oleh karena itu, pemerintah pun secara serius memberikan dukungan lebih kepada sektor UMKM. Salah satu dukungan tersebut diberikan oleh Kementerian Badan Usaha Milik Negara (BUMN) melalui

program Rumah BUMN. Rumah BUMN berfungsi sebagai *agent of development* dengan fokus pada peningkatan kualitas UMKM (Rumah BUMN, 2022). Pada tahun 2022, sudah terdapat 248 unit Rumah BUMN yang berdiri dan aktif di Indonesia. Salah satunya yaitu Rumah BUMN Bandung yang berada di bawah binaan Bank Rakyat Indonesia (BRI). Rumah BUMN Bandung menjadi salah satu sarana bagi UMKM di Kota Bandung untuk mendapatkan binaan. Binaan yang diberikan oleh Rumah BUMN Bandung yaitu berupa dukungan dan bimbingan pada peningkatan kompetensi, akses pasar, hingga permodalan. Akses kompetensi merupakan salah satu fokus utama pada kegiatan-kegiatan yang ada di Rumah BUMN. Akses kompetensi ini bertujuan untuk meningkatkan kompetensi dari UMKM yang bergabung dengan Rumah BUMN agar dapat berkembang menjadi lebih maju. Adapun beberapa jenis kegiatan dari akses kompetensi ini seperti pelatihan, pendampingan, dan konsultasi. Hingga saat ini, Rumah BUMN hanya memiliki 2 pegawai tetap yang terdiri dari koordinator dan *staff* admin. Oleh karena padatnya kegiatan yang harus dilakukan, Rumah BUMN Bandung membuka kesempatan kegiatan *internship* untuk para mahasiswa. Adapun divisi yang dibuka untuk melakukan kegiatan *internship* yaitu divisi admin, *content creator*, *event* dan *partnership* dan *SME relation*. Pekerjaan dari setiap divisi tersebut adalah sebagai berikut.

TABEL 1

Posisi	Pekerjaan
Admin	Menambahkan Jumlah Pelatihan di Link UMKM
	Menambahkan Berita & Foto Video di website Rumah BUMN
Content Creator	Membuat <i>Campaign</i>
	Membuat Konten Video
Event and Partnership	Mengadakan Pelatihan
SME Relation	Referral Simpanan UMKM
	Referral Pinjaman UMKM
	Referral Qris UMKM

Tentunya, dalam pelaksanaan pekerjaan oleh setiap divisi, Rumah BUMN Bandung sudah menetapkan *Key Performance Indicator* (KPI) yang perlu dicapai oleh setiap divisinya. KPI ini berfungsi sebagai alat ukur kinerja dan memberikan panduan yang jelas tentang apa yang diharapkan oleh Rumah BUMN dari setiap anggota divisi. KPI yang diberikan oleh Rumah BUMN Bandung ini berisi jumlah pekerjaan yang harus mereka capai dan disesuaikan dengan jumlah hari kerja pada setiap bulannya. KPI yang ditetapkan oleh Rumah BUMN Bandung untuk setiap divisi yang ada adalah sebagai berikut.

TABEL 2

KPI	Bulan	Divisi			
		Admin	Content Creator	Event and Partnership	SME Relation
	Januari	63	63	42	147

Februari	60	60	40	140
Maret	63	63	42	147
April	42	42	28	98
Mei	66	66	44	154
Juni	60	60	40	140
Juli	60	60	40	140
Agustus	66	66	44	154
September	60	60	40	140

Berdasarkan Tabel 2, terlihat bahwa setiap divisi di Rumah BUMN Bandung memiliki target KPI bulanan yang beragam. Divisi Admin dan *Content Creator* memiliki target KPI yang relatif konsisten dengan sedikit variasi. Pada bulan Januari dan Maret, target KPI untuk kedua divisi ini mencapai 63, sedangkan pada bulan Februari dan Agustus, target ditetapkan sedikit lebih rendah, yaitu 60. Penurunan drastis terjadi pada bulan April dengan target KPI turun menjadi 42, yang dikarenakan penyesuaian terhadap hari kerja pada bulan tersebut.

Sementara itu, divisi *Event and Partnership* memiliki target KPI yang lebih rendah dibandingkan dengan divisi lainnya. Pada bulan Januari, target KPI divisi ini adalah 42, dan mengalami sedikit *fluktuasi* di bulan-bulan berikutnya. Penurunan signifikan juga terlihat pada bulan April dengan target hanya 28, namun kembali naik menjadi 44 pada bulan Mei. Divisi *SME Relation* memiliki target KPI yang cukup tinggi dibandingkan divisi lainnya. Pada bulan Januari, target mencapai 147, dan sedikit menurun pada bulan Februari menjadi 140. Namun, target kembali stabil pada kisaran 147 hingga 154 di bulan-bulan berikutnya, menunjukkan bahwa divisi ini memiliki beban kerja yang lebih tinggi dan target yang lebih ambisius. Secara keseluruhan, tabel KPI ini menunjukkan bahwa setiap divisi di Rumah BUMN Bandung memiliki target yang disesuaikan dengan jumlah hari kerja dan beban kerja bulanan. Penurunan target KPI pada bulan April di semua divisi disebabkan oleh hari kerja yang lebih. Adapun jumlah mahasiswa *internship* beserta pegawai tetap Rumah BUMN Bandung pada bulan Januari sampai September tahun 2022 yaitu sebagai berikut.

TABEL 3

Jumlah Pekerja Bulan Januari-Maret		Jumlah Pekerja Bulan April-Juni		Jumlah Pekerja Bulan Juli-Agustus	
Koordinator	1	Koordinator	1	Koordinator	1
Admin	3	Admin	3	Admin	3
Content Creator	3	Content Creator	4	Content Creator	4
Event and Partnership	3	Event and Partnership	4	Event and Partnership	5
SME Relation	3	SME Relation	3	SME Relation	4
Total	13	Total	15	Total	17

Tentunya pelaksanaan tugas yang dilakukan oleh mahasiswa *internship* ini memiliki jangka waktu yang relatif singkat

yaitu berkisar antara tiga sampai enam bulan. Selain waktu pelaksanaan *internship* yang terbatas, kehadiran peserta *internship* juga memiliki peran penting dalam pencapaian KPI yang telah ditentukan. Berdasarkan wawancara dengan Koordinator Rumah BUMN Bandung, pendataan kehadiran dari para pekerja dilakukan melalui pengisian *google form*. Berikut merupakan rekap kehadiran dari para pekerja pada periode bulan Januari-september pada tahun 2022 yang direkap oleh Koordinator Rumah BUMN Bandung.

TABEL 4

Bulan	Total Hari kerja per bulan	Jumlah Pekerja	Kehadiran			
			Jumlah Hari Kerja Keseluruhan	Ha dir	Izin/ Sakit	% Kehadiran
Januari	21	15	315	312	3	99%
Februari	20	15	300	292	8	97%
Maret	21	15	315	309	6	98%
April	14	13	182	178	4	98%
Mei	22	13	286	281	5	98%
Juni	20	13	260	251	9	97%
Juli	20	17	340	335	5	99%
Agustus	22	17	374	365	9	98%
September	20	17	340	334	6	98%

Berdasarkan rekap data kehadiran pada Tabel 4, setiap pekerja yang sudah melakukan setiap satu pekerjaan, wajib melakukan pengisian *work report* pada *logbook* yang tersedia. Berikut merupakan data pengiriman *work report* melalui *logbook* yang dimiliki oleh Koordinator Rumah BUMN Bandung.

TABEL 5

Bulan	Pengiriman Work Report			
	Admin	Content Creator	Event and Partnership	SME Relation
Januari	189	126	84	294
Februari	120	60	40	140
Maret	189	63	42	147
April	84	126	112	98
Mei	198	132	88	308
Juni	120	60	40	140
Juli	120	240	120	280
Agustus	132	132	88	154
September	120	120	120	420

Setelah itu, didapatkan juga pencapaian kerja para pekerja pada periode bulan Januari-September pada tahun 2022 sebagai berikut.

TABEL 6

Bulan	Divisi			
	Admin	Content Creator	Event and Partnership	SME Relation
Januari	63	42	42	42
Februari	40	20	20	20
Maret	63	21	21	21
April	28	42	56	14
Mei	66	44	44	44
Juni	40	20	20	20
Juli	40	80	60	40
Agustus	44	44	44	22
September	40	40	60	60

Berdasarkan data yang diberikan, terlihat bahwa pencapaian kerja dari masing-masing divisi sering kali belum mencapai KPI yang telah ditentukan. Seperti pada bulan Januari, nilai KPI untuk divisi *Content Creator* adalah 63, tetapi divisi *Content Creator* hanya berhasil mencapai 42. Namun, setelah dilakukan observasi dan wawancara lebih lanjut kepada pekerja Rumah BUMN, didapatkan bahwa data yang direkap oleh Koordinator tidak sinkron dengan data hasil rekap kehadiran dan *work report* melalui *logbook* yang sudah mereka kerjakan. Berikut merupakan ketidaksinkronan dari rekap kehadiran, pengiriman *work report* dan juga ketercapaian kerja dengan kondisi aktual di Rumah BUMN Bandung pada bulan Januari sampai September pada tahun 2022.

TABEL 7

Bulan	Total Hari kerja per bulan	Jumlah Pekerja	Kehadiran			
			Jumlah Hari Kerja Keseluruhan	Ha dir	Izin/ Sakit	% Kehadiran
Januari	21	15	315	312	3	99%
Februari	20	15	300	294	6	98%
Maret	21	15	315	309	6	98%
April	14	13	182	178	4	98%
Mei	22	13	286	281	5	98%
Juni	20	13	260	251	9	97%
Juli	20	17	340	337	3	99%

Agustus	22	17	374	366	8	98%
September	20	17	340	336	4	99%

TABEL 8

Bulan	Pengiriman Work Report			
	Admin	Content Creator	Event and Partnership	SME Relation
Januari	189	126	84	294
Februari	126	126	42	147
Maret	189	63	42	147
April	126	189	168	147
Mei	189	126	84	294
Juni	126	63	42	147
Juli	189	252	168	294
Agustus	126	126	84	294
September	189	189	126	441

TABEL 9

Bulan	Divisi			
	Admin	Content Creator	Event and Partnership	SME Relation
January	63	42	42	42
February	40	40	20	20
Maret	63	21	21	21
April	28	42	56	14
Mei	66	44	44	44
Juni	40	20	20	20
Juli	60	80	80	40
Agustus	44	44	44	44
September	60	60	60	60

Perbedaan yang dapat terlihat yaitu presentase kehadiran pekerja pada bulan Februari yang dimiliki oleh Koordinator adalah 97%. Sedangkan berdasarkan perhitungan ulang dari hasil *google form* presentase kehadiran pada bulan Februari adalah 98%. Selanjutnya pada pengumpulan *work report* yang direkap oleh Koordinator untuk divisi Admin pada bulan Februari adalah 120. Sedangkan berdasarkan hasil rekap ulang pada bulan Februari divisi Admin mengumpulkan sebanyak 126 *work report*. Pada hasil pencapaian kerja pun terjadi ketidaksinkronan yang sangat jelas. Perbedaan rekap data yang dimiliki ini dikarenakan data yang tidak terintegrasi dan kurangnya akses peserta internship terhadap *Standard Operating Procedure* (SOP) pada Rumah BUMN menyebabkan pencapaian target tidak tercapai. Penggunaan berbagai *platform* seperti *Google Forms* dan *Excel* menyulitkan akses dan koordinasi data.

Selain itu, tidak adanya sistem pengawasan *real-time* terhadap pekerjaan peserta *internship* membuat ketidaksesuaian dalam laporan tidak segera terlihat. Maka dari itu dibutuhkan suatu sistem yang dapat mengatasi permasalahan tersebut untuk memudahkan pihak Rumah BumN maupun *internship* untuk mempermudah mengakses semua data yang ada dalam Rumah BumN. Sistem manajemen *work report* berbasis *website* bertujuan untuk membantu mengorganisir data di Rumah BUMN Bandung, memudahkan akses meskipun terjadi pergantian *outsourse*. Selain itu, peserta *internship* dapat lebih memahami *jobdesc* yang diberikan ke divisi masing-masing, sehingga pekerjaan dapat diselesaikan sesuai target.

II. KAJIAN TEORI

Berikut merupakan kajian teori yang digunakan dalam penelitian ini

A. Pengertian Manajemen Kinerja

Manajemen kinerja adalah proses berkelanjutan yang melibatkan perencanaan, pemantauan, dan evaluasi kinerja karyawan (Budiarto, 2017). Proses ini melibatkan komunikasi yang terbuka dan berkesinambungan, penciptaan visi bersama, serta pendekatan strategis yang terintegrasi yang berfungsi sebagai pendorong utama dalam mencapai tujuan organisasi. Gambar 2 menunjukkan siklus manajemen kinerja.



GAMBAR 2

Berikut adalah penjelasan dari siklus sistem manajemen kinerja yang tergambar pada Gambar 2:

- Tahap Perencanaan (Planning):** Tahap ini merupakan usaha kolaboratif antara manajer dan karyawan, meliputi:
 - Peninjauan deskripsi pekerjaan karyawan untuk memastikan kesesuaian dengan pekerjaan yang dilakukan.
 - Penilaian keterkaitan antara deskripsi pekerjaan karyawan dengan rencana kerja, tujuan, dan strategi perusahaan.
 - Pembuatan rencana kerja yang mendetail tentang tugas-tugas yang harus dilakukan, hasil yang diharapkan, dan standar atau langkah untuk evaluasi kinerja.
 - Pentuan 3-5 area kinerja utama yang menjadi target selama satu tahun.
 - Identifikasi tujuan pelatihan untuk mengembangkan keterampilan dan kompetensi karyawan sesuai dengan kebutuhan pekerjaan mereka.

f. Pengidentifikasi tujuan pengembangan karir yang akan diintegrasikan dalam perencanaan karir jangka panjang karyawan.

2. **Tahap Pemantauan:** Proses ini tidak melibatkan pengawasan secara detail terhadap cara karyawan menjalankan tugas-tugasnya, melainkan fokus pada hasil yang dicapai serta perilaku individu dan dinamika tim yang berdampak pada lingkungan kerja.
3. **Tahap Penilaian (Review):** Tahap ini merupakan kesempatan untuk meninjau dan meringkas kinerja karyawan selama periode tersebut, yang diwujudkan melalui pertemuan penilaian kinerja atau appraisal meeting.

B. *Management Information System (MIS)*

Manajemen Sistem Informasi (MIS) adalah sistem yang menggabungkan teknologi informasi dengan kegiatan bisnis untuk membantu perusahaan mencapai tujuan strategisnya. Menekankan bahwa integrasi antara teknologi informasi dan proses bisnis sangat penting untuk meningkatkan efisiensi dan efektivitas organisasi. MIS melibatkan komponen seperti perangkat keras, perangkat lunak, data, prosedur, dan sumber daya manusia yang bekerja sama untuk mengumpulkan, menyimpan, memproses, dan mendistribusikan informasi. Fungsi utama MIS meliputi peningkatan efisiensi melalui otomatisasi proses bisnis, penyediaan data akurat untuk pengambilan keputusan yang lebih baik, dan membantu perusahaan merespons perubahan pasar serta kebutuhan pelanggan guna memperoleh keunggulan kompetitif (Laudon, *Management Information System Managing the Digital Firm*, 2020).

C. *Rapid Application Development (RAD)*

Rapid Application Development adalah metodologi pengembangan perangkat lunak yang berfokus pada pengembangan cepat dan iteratif, dengan keterlibatan aktif pengguna. Ini memungkinkan tim pengembang untuk menghasilkan perangkat lunak yang memenuhi kebutuhan pengguna dengan lebih cepat dan efisien. Meskipun memiliki beberapa kekurangan, seperti kebutuhan akan keterlibatan pengguna yang tinggi dan mungkin tidak cocok untuk proyek skala besar, RAD tetap menjadi pilihan yang efektif untuk banyak proyek pengembangan perangkat lunak yang membutuhkan fleksibilitas dan kecepatan. Pendekatan ini memastikan bahwa solusi yang dihasilkan tidak hanya cepat dan tepat sasaran, tetapi juga sesuai dengan kebutuhan dan tujuan organisasi (McConell, 2010). Setiap fase dalam Rapid Application Development (RAD) menekankan aspek tertentu dari pengembangan sistem. Berikut adalah penjelasan mengenai fase-fase utama dalam RAD (Sasmito, 2020):

1. **Perencanaan Kebutuhan (Requirement Planning):** Pada tahap ini, pengguna dan analis bertemu untuk mengidentifikasi tujuan sistem dan kebutuhan informasi yang diperlukan untuk mencapai tujuan tersebut. Keterlibatan dari kedua belah pihak dianggap sebagai aspek krusial pada tahap ini.

2. **Proses Desain Sistem:** Pada fase ini, keterlibatan aktif dari pengguna menjadi penentu dalam mencapai tujuan, karena proses ini melibatkan perancangan dan perbaikan berdasarkan umpan balik antara pengguna dan analis. Pengguna memiliki kemampuan untuk memberikan komentar langsung terkait ketidaksesuaian desain, dan desain sistem dibangun dengan merujuk pada dokumentasi kebutuhan pengguna yang telah dibuat pada tahap sebelumnya. Hasil dari tahap ini mencakup spesifikasi perangkat lunak, termasuk struktur sistem secara keseluruhan, struktur data, dan elemen-elemen lainnya.

3. **Implementasi (Implementation):** Pada tahap ini, terlibatlah para pengembang program yang merancang program berdasarkan desain yang telah disetujui oleh pengguna dan analis. Sebelum program diterapkan di organisasi, dilakukan uji coba untuk memastikan apakah ada kesalahan atau tidak. Pada tahap ini, pengguna juga memberikan umpan balik terhadap sistem yang telah dibuat dan memberikan persetujuan terkait implementasi tersebut.

D. *Laravel Framework*

Laravel merupakan framework PHP yang sangat berguna baik bagi pemula maupun programmer berpengalaman. *Framework* ini membantu mempercepat pengembangan dan peluncuran aplikasi web dengan metode PHP berorientasi objek yang canggih. Sintaksisnya yang mudah dipahami dan fungsionalitas modernnya menarik bagi pengembang yang ingin membuat aplikasi yang andal. Penggunaan *framework* mempermudah pengembangan karena menyediakan berbagai modul dan koneksi yang diperlukan. *Web-framework* ini menyediakan fondasi awal untuk pembuatan aplikasi dan memungkinkan pengembang untuk lebih fokus pada aspek-aspek yang lebih menarik. *Laravel* juga menawarkan fitur-fitur kuat seperti lapisan abstraksi database yang ekspresif, dependency injection, dan skalabilitas yang tinggi (Subecz, 2021).

E. *Mysql*

MySQL adalah sistem manajemen basis data relasional yang menggunakan *Structured Query Language (SQL)* untuk mengelola dan mengakses data secara efisien. Data dalam *MySQL* disimpan dalam bentuk tabel yang terdiri dari baris dan kolom, di mana setiap baris merepresentasikan rekaman dan setiap kolom berisi bidang data tertentu. *SQL* memungkinkan permintaan sederhana dan pengambilan data yang cepat, membuat *MySQL* ideal untuk berbagai aplikasi dan sistem yang memerlukan pengelolaan data terstruktur dan cepat (Nixon, 2021).

F. *Unified Modelling Language*

UML (Unified Modeling Language) adalah suatu bahasa standar yang umumnya digunakan dalam industri untuk merinci persyaratan, melakukan analisis, dan merancang sistem. Selain itu, *UML* juga digunakan untuk

menjelaskan struktur dan arsitektur dalam pemrograman berorientasi objek (Putra & Andriani, 2019). UML dapat ditampilkan dalam bentuk grafik dan bagan beserta grafik dan bagan yang menunjukkan kondisi dunia nyata saat ini. Berikut merupakan beberapa model dari UML

a. *Use Case Diagram*

Use Case Diagram adalah metodologi yang digunakan dalam analisis sistem untuk mengidentifikasi, mengklarifikasi, dan persyaratan sistem (Grechanik, McKinley, & Perry, 2007).

b. *Sequence Diagram*

Sequence Diagram merupakan salah satu representasi perilaku yang sering digunakan dalam UML. Diagram ini memvisualisasikan interaksi dan pertukaran pesan antar objek sepanjang waktu. Diagram urutan digunakan untuk mengilustrasikan bagaimana pesan-pesan dikirim di antara objek atau instance untuk mengeksekusi suatu tugas. Secara sederhana, Diagram urutan membantu kita memahami cara objek berkomunikasi dan berbagi informasi dalam suatu proses atau skenario tertentu (Al-Fedaghi, 2021).

c. *Activity Diagram*

Activity Diagram adalah merupakan salah satu dari lima diagram UML yang sering digunakan dalam industri. Prioritas tinggi pada penggunaan Diagram Kelas, yang menempati peringkat pertama sebagai diagram UML yang paling umum, menjadi alasan mengapa kami memilih untuk menekankan kedua model ini (Reggio, Leotta, Ricca, & Clerissi, 2013).

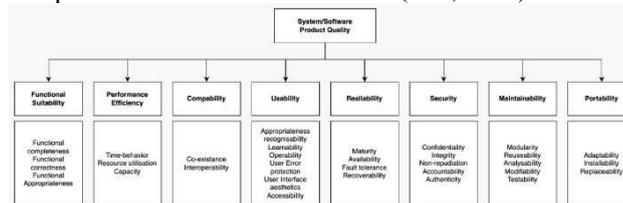
G. *Black Box Testing*

Pengujian kotak hitam, yang juga dikenal sebagai pengujian fungsional, merupakan suatu metode pengujian di mana penilaian dilakukan berdasarkan pada spesifikasi yang telah ditentukan oleh klien (Jampani, Talasu, & Manjula, 2016). Dalam proses ini, penguji sistem tidak memiliki akses ke kode program internal dari sistem yang sedang diuji. Oleh karena itu, pengujian kotak hitam difokuskan pada fungsi eksternal sistem tanpa memperhatikan detail implementasi internal.

H. *User Acceptance Testing*

User Acceptance Testing (UAT) adalah tahap pengujian dalam metode V-Model yang melibatkan pengguna akhir atau pemangku kepentingan yang relevan untuk menguji sistem perangkat lunak. Tujuan UAT adalah memastikan bahwa sistem memenuhi kebutuhan bisnis dan persyaratan pengguna. Pengujian ini dilakukan dalam lingkungan dan skenario penggunaan nyata sebelum sistem diterima dan diimplementasikan. UAT biasanya dilakukan sebelum peluncuran fitur baru dalam aplikasi untuk memastikan bahwa semua aspek sistem telah diuji dan siap digunakan dalam operasi sehari-hari (Anjasmara & Arif, 2023). Dalam tugas akhir ini, UAT diterapkan dengan standar ISO 25010. Standar ini mencakup delapan karakteristik yang mengacu pada sifat dinamis dan statis dari sistem komputer. Karakteristik dan sub-karakteristik

tersebut memberikan istilah yang konsisten untuk mendefinisikan, mengukur, dan mengevaluasi kualitas sistem dan perangkat lunak (Pratama & Mutiara, 2021). Hal ini dapat dicapai dengan mendefinisikan karakteristik kualitas yang diperlukan dan diinginkan yang terkait dengan tujuan dan sasaran pemangku kepentingan terhadap sistem tersebut, berikut merupakan gambaran delapan indikator dalam ISO 25010 (ISO, 2011).



GAMBAR 3

Dalam pengembangan aplikasi, prioritas utama adalah memenuhi kebutuhan dan harapan pengguna. Oleh karena itu, dengan fokus pada tiga parameter utama yaitu *Functional Suitability*, *Performance Efficiency*, dan *Usability*, dapat secara tepat menargetkan kebutuhan utama pengguna. Berikut penjelasan lebih lanjut mengenai ketiga parameter ini:

1. *Functional Suitability*: Memastikan aplikasi menawarkan semua fungsi yang dibutuhkan pengguna (*Functional Completeness*). Hal ini penting untuk memberikan solusi lengkap bagi kebutuhan pengguna, sehingga meningkatkan kepuasan dan kepercayaan pengguna terhadap aplikasi.
2. *Performance Efficiency* : Memastikan aplikasi memiliki waktu respons yang cepat (*Time-behaviour*). Pengguna menginginkan aplikasi yang responsif dan cepat untuk meningkatkan efisiensi dalam menyelesaikan tugas, mengurangi frustrasi, dan memberikan pengalaman yang lebih baik.
3. *Usability* : Memastikan aplikasi mudah digunakan dan dipahami (*Learnability*). Dengan membuat aplikasi yang mudah dipelajari, pengguna baru dapat dengan cepat memahami cara menggunakan aplikasi, mengurangi waktu pembelajaran, dan meningkatkan adopsi aplikasi.

I. *Software Development Life Cycle (SDCL)*

Siklus Hidup Pengembangan Sistem (SDLC) atau Siklus Hidup Sistem adalah proses dalam rekayasa sistem dan perangkat lunak yang melibatkan pembuatan dan perubahan sistem, serta model dan metodologi yang digunakan untuk mengembangkan sistem tersebut. Konsep SDLC ini biasanya mengacu pada pengembangan sistem komputer atau sistem informasi. Dalam rekayasa perangkat lunak, SDLC menjadi dasar bagi berbagai metodologi pengembangan perangkat lunak. Metodologi ini membentuk kerangka kerja untuk merencanakan dan mengendalikan proses pengembangan sistem informasi, yang pada dasarnya adalah proses pengembangan perangkat lunak. Terdapat tiga jenis metode siklus hidup sistem yang banyak digunakan: siklus hidup sistem tradisional, siklus hidup menggunakan prototyping, dan

siklus hidup sistem berorientasi objek. Konsep SDLC juga mendasari berbagai model pengembangan perangkat lunak, seperti model *waterfall*, *prototyping*, *iteratif*, *spiral*, *rapid application development* (RAD), dan lainnya (Kurniyanti & Murdiani, 2022). Tahapan-tahapan dalam SDLC secara keseluruhan, sebagaimana dijelaskan dalam buku (Rosa, 2022), adalah sebagai berikut:

1. Inisiasi (*initiation*): Tahap ini biasanya dimulai dengan penyusunan proposal proyek perangkat lunak.
2. Pengembangan konsep sistem (*system concept development*): Menentukan lingkup konsep yang mencakup dokumen lingkup sistem, analisis manfaat dan biaya, rencana manajemen, serta pembelajaran kemudahan sistem.
3. Perencanaan (*planning*): Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya, serta menyediakan dasar untuk memperoleh sumber daya yang diperlukan guna mencapai solusi.
4. Analisis kebutuhan (*requirement analysis*): Menganalisis kebutuhan pengguna sistem perangkat lunak, mengidentifikasi kebutuhan tersebut, dan menyusun dokumen kebutuhan fungsional.
5. Desain (*design*): Mengubah kebutuhan detail menjadi kebutuhan yang lengkap, dengan dokumen desain sistem yang berfokus pada cara memenuhi fungsi-fungsi yang diperlukan.
6. Pengembangan (*development*): Mengonversi desain menjadi sistem informasi yang lengkap, mencakup bagaimana memperoleh dan menginstalasi lingkungan sistem yang diperlukan.
7. Implementasi (*implementation*): Termasuk persiapan untuk implementasi, penerapan perangkat lunak dalam lingkungan produksi (pengguna), serta penanganan dan penyelesaian masalah yang teridentifikasi.

III. METODE

Metode yang digunakan pada penelitian ini yaitu metode *Rapid Application Development* (RAD). Dimana metode ini dapat membantu untuk mengurangi waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional, khususnya antara tahap perancangan dan implementasi suatu sistem informasi. Adapun tahapan yang dilakukan pada penelitian ini yaitu sebagai berikut

1. Tahap Pendahuluan

Tahap pendahuluan merupakan penulisan latar belakang masalah dari suatu problem owner yang terjadi pada sistem. Dalam penulisan latar belakang masalah dibuat dengan data-data yang merujuk pada masalah yang terjadi pada sistem eksisting dan problem owner tersebut. Setelah dilakukannya latar belakang dilanjutkan pembuatan perumusan masalah yang didapat dari permasalahan yang ada di latar belakang. Selanjutnya membuat tujuan dari

permasalahan yang terjadi yang merujuk pada solusi yang akan diberikan di tugas akhir ini yaitu untuk meningkatkan kualitas program pelatihan.

2. Tahap Pengembangan Sistem

Tahap pengembangan sistem merupakan tahap yang berkaitan dengan pengembangan sistem yang berkaitan dengan perencanaan dan pengumpulan kebutuhan yang diperlukan untuk mengembangkan aplikasi, tahap ini ini juga berkaitan dengan perancangan desain sistem aplikasi. Tahap ini terdiri dari dua bagian yaitu *requirement planning* dan *prototype*, dua bagian tersebut berisi sebagai berikut:

- Requirement Planning

Tahap ini berisi analisis stakeholder yang menentukan pihak user yang terlibat di dalam sistem dan juga menentukan peran setiap stakeholder yang ada. Kemudian melakukan pengumpulan data terkait kebutuhan dari stakeholder. Pengumpulan data menggunakan cara interview dengan Rumah BUMN yang merupakan *problem owner*. Selanjutnya dilakukan identifikasi dari hasil pengumpulan data tersebut yang menjadi acuan fitur-fitur pada desain sistem.

- Prototype

Tahap ini berisi perancangan desain sistem dan pembuatan desain mockup. Dalam pembuatan sistem ini diawali dengan pembuatan basis data yang dirancang melalui *entity relationship diagram* yang bertujuan untuk mengetahui relasi antar objek dalam basis data. Selanjutnya melakukan desain sistem menggunakan *Unified Modeling Language* (UML), seperti pembuatan *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*. Kemudian, dilanjutkan dengan pembuatan mockup dari sistem agar dapat mempresentasikan bagaimana sistem yang akan dibuat.

3. Tahap Analisis dan Hasil

Pada tahap ini berisi implementasi dari hasil tahap sebelumnya yaitu, pembuatan aplikasi. Dari hasil desain sistem *Unified Model Language* (UML) dan desain mockup yang sudah dibuat. Kemudian, dilanjutkan dengan pengujian untuk mengeksplorasi berbagai skenario agar dapat mengetahui sistem yang dibuat sudah sesuai dan tidak ada bug yang terjadi. Setelah itu dilanjutkan dengan dengan pengujian *User Acceptance Test* di mana pengguna akhir akan memastikan bahwa sistem sudah memenuhi persyaratan bisnis dan memenuhi ekspektasi dari pengguna akhir, dan jika belum memenuhi ekspektasi dan persyaratan yang ada maka tahap ini akan kembali pada tahap perancangan *Unified Model Language* (UML). Dalam tahap ini akan dilakukan analisis

kekurangan dalam sistem dan akan di revisi sesuai dengan sistem yang dijanjikan. Dan jika sistem sudah sesuai dengan persyaratan dan ekspektasi pengguna akhir maka akan dilanjutkan dengan tahap selanjutnya yaitu evaluasi hasil rancangan yang dimana dalam evaluasi ini akan berfokus pada evaluasi hasil rancangan yang sudah dibuat kemudian di bandingkan dengan kondisi eksisting sistem apakah usulan sistem sudah lebih baik dan dapat meningkatkan kualitas dari program pelatihan daripada sistem yang sudah ada. Setelah itu dilakukan analisis kelebihan yang ada pada sistem yang sudah dibuat dan kekurangan ada pada sistem yang sudah dibuat.

4. Kesimpulan dan Saran

Dalam tahap akhir ini akan dilakukan evaluasi keseluruhan dari perancangan sistem yang sudah dibuat untuk menilai apakah perancangan sudah dibuat sesuai dengan tujuan. Dan juga diberikan saran untuk pengembangan sistem selanjutnya.

IV. HASIL DAN PEMBAHASAN

A. Proses Perancangan

1. Identifikasi Stakeholder

• Koordinator

Sebagai problem owner yang memiliki tanggung jawab sebagai penanggung jawab yaitu sebagai bagian yang akan mengakses, consulted yaitu sebagai pihak yang dapat diskusi mengenai kesesuaian sistem. Dan juga memiliki tanggung jawab untuk monitoring work report yang sudah di input.

• Admin

Sebagai problem user yang memiliki kewajiban dalam melakukan kehadiran setiap kedatangan inputing dokumen, dan menginput hasil work report.

• Divisi Event, Design Content

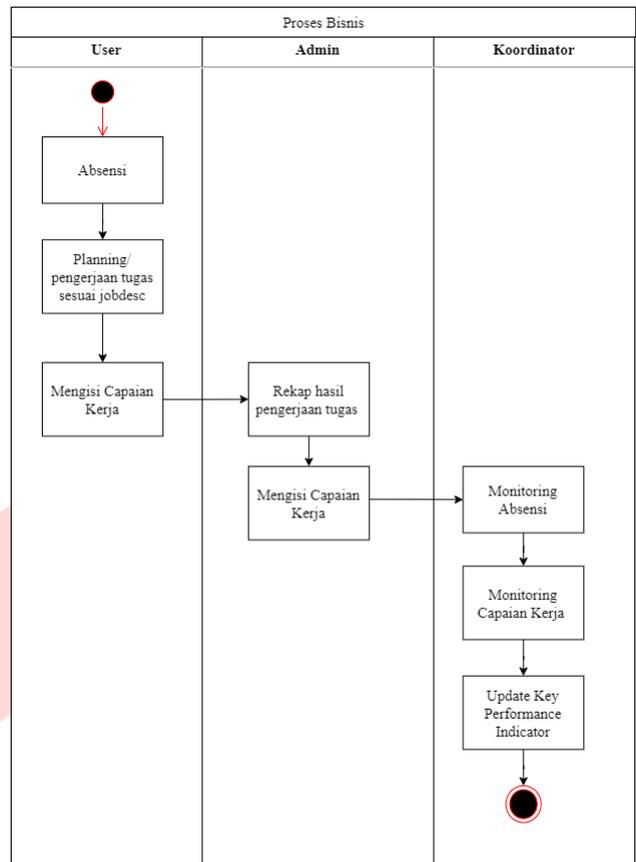
Sebagai problem user yang memiliki kewajiban dalam melakukan inputing dokumen, dan menginput hasil work report.

TABEL 4

No	Stakeholder	Pihak Terlibat
1	Problem Owner	Koordinator Rumah BUMN Bandung.
2	Problem User	Koordinator Rumah BUMN Bandung, Admin, Divisi Design Content social media, Divisi Event.

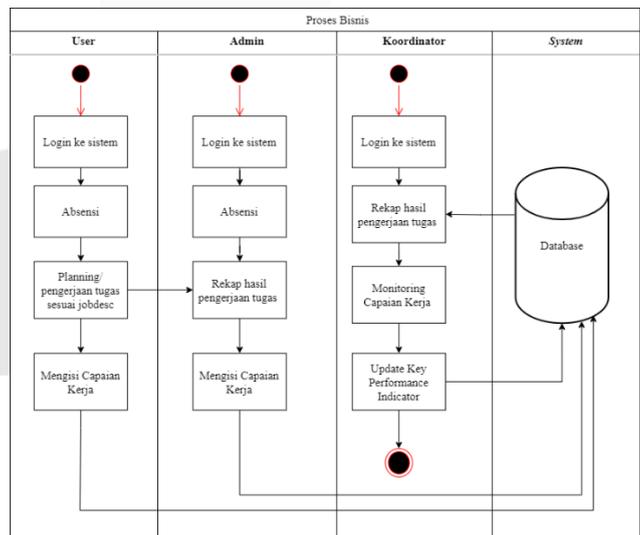
2. Analisis Proses Bisnis

Analisis proses bisnis dalam penulisan tugas akhir ini berdasarkan Proses Bisnis usulan, dikarenakan kondisi eksisting pada bagian website Rumah BUMN tidak memiliki proses bisnis, sehingga penulis menganalisis berdasarkan analisis stakeholder yang ada pada Rumah BUMN Bandung.



GAMBAR 4

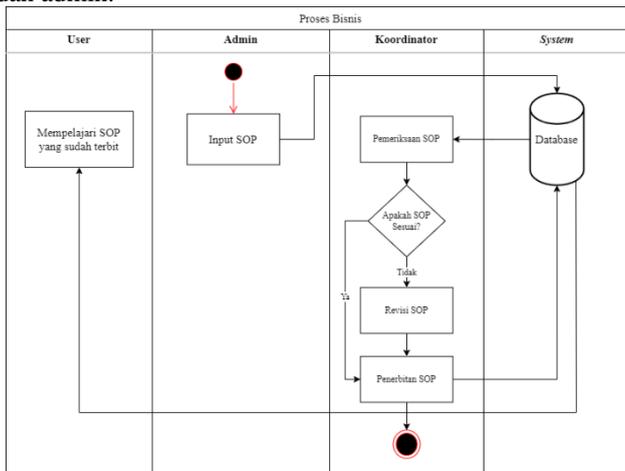
Gambar 4 merupakan proses bisnis eksisting yang ada di Rumah BUMN Bandung yang melibatkan 3 stakeholder yaitu User yang berisi divisi event, divisi design content yang kedua yaitu admin, dan ketiga yaitu koordinator.



GAMBAR 5

Gambar 5 diatas merupakan Proses Bisnis Usulan, proses ini melibatkan beberapa stakeholder. pertama, user (UMKM) yang berisi divisi event, design content yang di proses awalnya akan melakukan kehadiran kemudian mengerjakan tugas sesuai divisi dan mengisi capaian kerja yang akan masuk ke dalam database, dan di monitoring oleh koordinator. Kemudian admin memiliki tugas melakukan kehadiran dan rekap hasil pengerjaan tugas yang dilakukan

oleh divisi lainnya dan juga mengisi capaian kerja yang sudah dilakukan. Dilanjutkan oleh tugas dari koordinator yang berfungsi sebagai orang yang *monitoring* pekerjaan dari user dan admin.



GAMBAR 6

Gambar 6 merupakan proses bisnis usulan dari dokumentasi SOP, proses ini melibatkan beberapa *stakeholder*. Pertama yaitu dari sisi admin melakukan inputasi dokumen SOP kemudian data dokumen sop masuk ke dalam database. Kedua yaitu koordinator melakukan pemeriksaan dokumen sop, jika dokumen sop sudah sesuai maka akan dilakukan penerbitan sop kemudian akan masuk ke dalam database, dan jika belum sesuai maka akan dilakukan revisi dan dilanjutkan ke penerbitan dokumen sop. Ketiga yaitu *user* yang akan mempelajari dokumen sop yang sudah diterbitkan.

3. Identifikasi Kebutuhan Pengguna

Identifikasi kebutuhan pengguna dilakukan agar dapat memberikan deskripsi keseluruhan mengenai tujuan dari sistem yang ingin dibangun oleh *problem owner* dan *problem user*. Deskripsi dari keseluruhan sistem yaitu spesifikasi sistem, *User Interface* dan *User Experience* dari sistem yang dibutuhkan oleh *problem owner*. *Problem owner* membutuhkan sistem yang dapat mengawasi pekerjaan dari semua divisi agar dapat target dari pekerjaan dapat tercapai. Tabel 10 berikut merupakan hasil dari identifikasi kebutuhan yang menyesuaikan dengan kebutuhan pengguna berdasarkan hasil wawancara oleh *problem owner* yaitu pak Supriatna dari Rumah BUMN Bandung.

TABEL 10

No	Kebutuhan Pengguna
1	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat melakukan proses Login untuk <i>problem user</i> .
2	Sistem Informasi Manajemen berbasis <i>web</i> yang memiliki tampilan yang sederhana.
3	Sistem Informasi Manajemen berbasis <i>web</i> yang memiliki <i>User Experience</i> yang mudah digunakan dan mudah diakses oleh <i>problem user</i> .
4	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat diakses dengan laptop dan komputer.
5	Sistem Informasi Manajemen berbasis <i>web</i> yang memiliki hak akses untuk <i>problem owner</i> dan <i>user</i> .
6	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menginput kehadiran.

7	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menampilkan data kehadiran per-hari atau perbulan.
8	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menginput dokumen.
9	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menghapus dokumen.
10	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menginput target dan capaian kerja
11	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menampilkan target dan capaian kerja
12	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menginput hasil <i>work report</i>
13	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menampilkan keseluruhan hasil <i>work report</i>
14	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menginput KPI, target, posisi
15	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menampilkan KPI, target, posisi
16	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menghitung persentase pencapaian dari KPI
17	Sistem Informasi Manajemen berbasis <i>web</i> yang dapat menampilkan persentase pencapaian dari KPI

Berdasarkan identifikasi pengguna, sistem yang dirancang memiliki empat kelompok *User* pengguna yaitu pihak koordinator atau *problem owner* dari Rumah BUMN Bandung, kemudian Admin, Divisi *Design Content social media*, Divisi *Event*. Dalam empat kelompok *User* memiliki hak akses yang berbeda pada tiap kelompoknya. Hak akses ini dirancang berdasarkan aktivitas yang dilakukan oleh Tabel 11 berikut merupakan hak akses setiap kelompok.

TABEL 11

<i>User</i>	Hak Akses
Koordinator	<ol style="list-style-type: none"> 1. Dapat melakukan <i>login</i>, <i>logout</i> 2. Dapat melihat data kehadiran 3. Dapat melihat, menghapus dokumen SOP 4. Dapat melihat, menambah data KPI 5. Dapat melihat keseluruhan hasil kinerja harian
Admin	<ol style="list-style-type: none"> 1. Dapat melakukan kehadiran 2. Dapat melihat, menambahkan data dokumen SOP 3. Dapat menambahkan data hasil kinerja harian
<i>Intern</i>	<ol style="list-style-type: none"> 1. Dapat melakukan kehadiran 2. Dapat menambahkan data hasil kinerja harian 3. Dapat melihat dokumen SOP

Ada pun data tambahan berupa *Key Performance Indicator* yang berisi target dari capaian kegiatan pada RB Bandung dan persentase total dari pencapaian kegiatan yang sudah dilakukan. Kemudian data hasil *work report* dari setiap divisi dan perorangannya di RB Bandung. Penjelasan terkait data sekunder terlampir pada Tabel 12.

TABEL 12

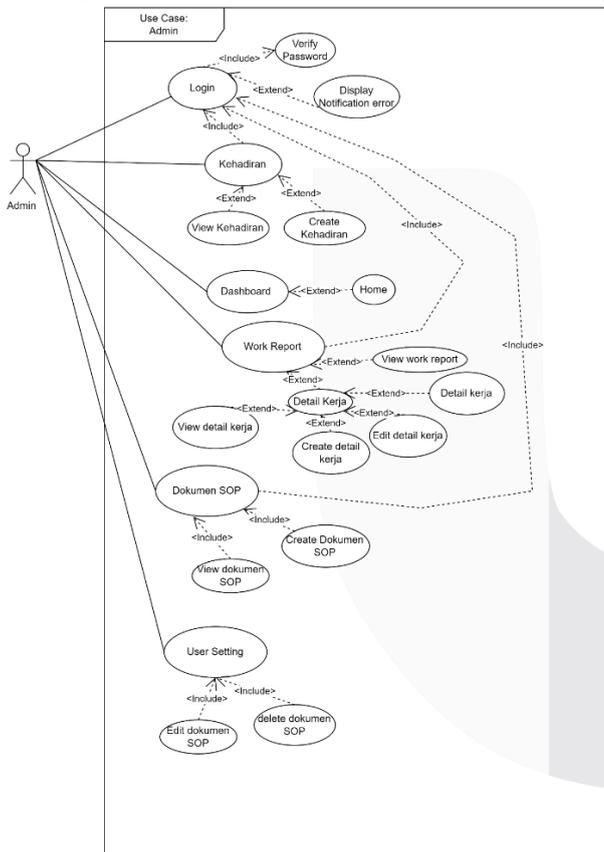
Data Sekunder	Deskripsi
---------------	-----------

Data Performance Indicator	Key	Data ini berisikan tentang target dari capaian kegiatan yang ada pada RB Bandung dan Persentase total dari pencapaian kegiatan yang sudah dilakukan.
Data Hasil Work report		Data ini berisikan tentang hasil work report tiap individu berdasarkan divisinya.

Sistem manajemen tersebut membutuhkan perangkat lunak agar dapat menjalankan sistem yang akan dikelola.

4. Pemodelan Usecase Diagram

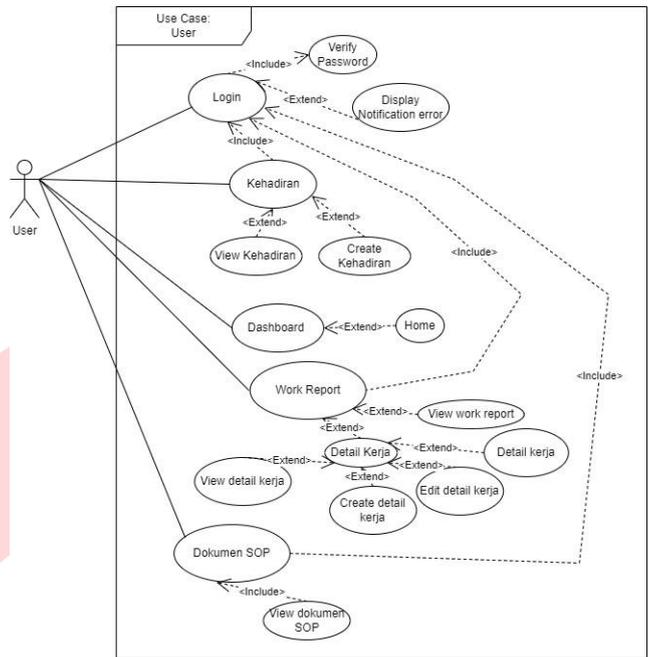
Use Case diagram adalah sebuah diagram UML yang menggambarkan interaksi antara aktor-aktor dengan sistem. Diagram ini memungkinkan peninjauan sistem dari sudut pandang eksternal, memberikan gambaran yang jelas tentang bagaimana pengguna berinteraksi dengan sistem. Dalam konteks aktor Koordinator, diagram ini menggambarkan peran dan fungsi yang dimiliki Koordinator dalam sistem. Aktor-aktor dalam pemodelan use case ini dibagi menjadi tiga yaitu Admin, User, Koordinator. berikut merupakan usecase pada aktor Admin.



GAMBAR 7

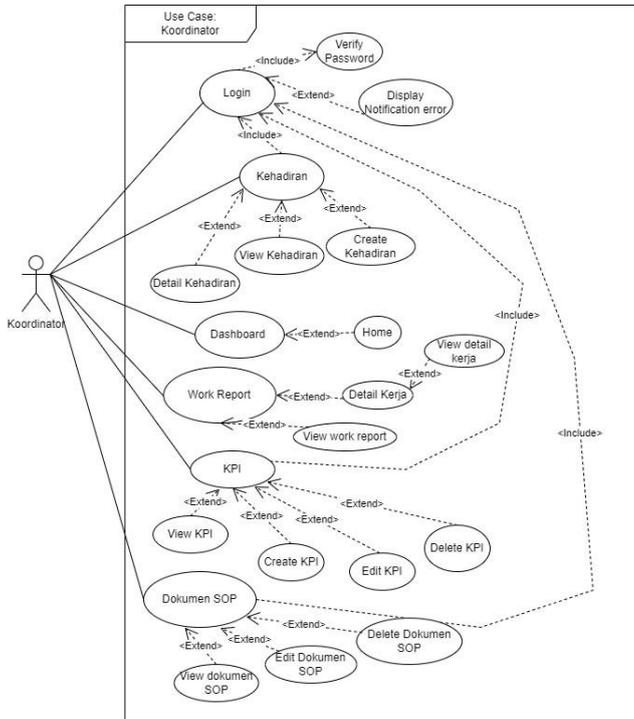
Gambar 7 merupakan usecase diagram pada aktor Admin. Admin diberikan hak akses untuk melakukan login dan jika melakukan kesalahan pada proses login maka akan diberikan notifikasi error. Dan jika sudah berhasil melakukan login, admin dapat hak akses menu kehadiran di dalam menu kehadiran admin terdapat view kehadiran, dan create kehadiran agar dapat melakukan kehadiran kehadiran. Kemudian terdapat juga menu work report Admin yang di dalamnya berisikan view work report dan halaman detail

kerja yang berisi detail kerja, create detail kerja, edit detail kerja dan delete detail kerja. Selanjutnya admin juga dapat hak akses untuk memasuki menu dokumen SOP dan melakukan create dokumen SOP.



GAMBAR 18

Gambar 8 merupakan usecase diagram pada aktor user. User diberikan hak akses untuk melakukan login dan jika melakukan kesalahan pada proses login maka akan diberikan notifikasi error. Dan jika sudah berhasil melakukan login, admin dapat hak akses menu kehadiran di dalam menu kehadiran admin terdapat view kehadiran, dan create kehadiran agar dapat melakukan kehadiran kehadiran. Kemudian terdapat juga menu work report admin yang di dalamnya berisikan view work report dan halaman detail kerja yang berisi detail kerja, create detail kerja, edit detail kerja dan delete detail kerja. Selanjutnya admin juga dapat hak akses untuk memasuki menu dokumen SOP dan melakukan download dokumen SOP.

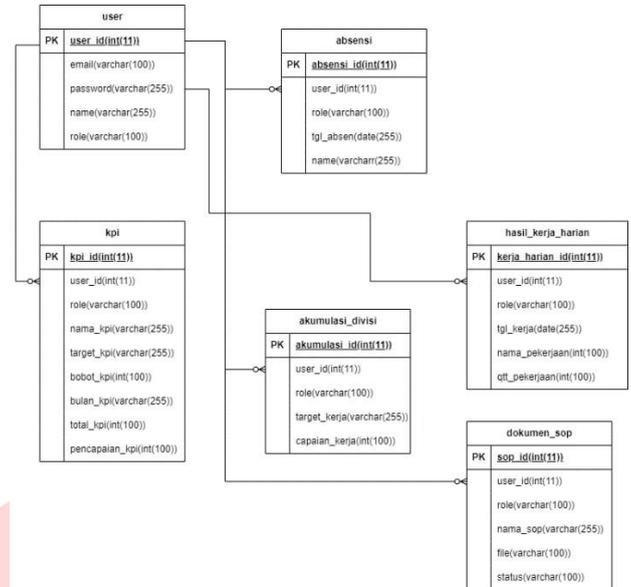


GAMBAR 9

Gambar 9 merupakan usecase diagram pada aktor koordinator. Koordinator diberikan hak akses untuk melakukan *login* dan jika melakukan kesalahan pada proses login maka akan diberikan notifikasi *error*. Dan jika sudah memasuki sistem maka memiliki akses menu dashboard yang berisi tentang *view KPI*, *Create KPI*, *Edit KPI*, dan *Delete KPI*. Kemudian terdapat juga menu Kehadiran yang berisikan data kehadiran yang masuk. Menu selanjutnya adalah dokumen SOP dimana koordinator dapat memeriksa dokumen SOP, *edit* dokumen SOP, dan dapat melakukan *delete* dokumen sop. Menu selanjutnya terdapat halaman *work report* dimana koordinator dapat hak akses untuk melihat keseluruhan data *work report* dari setiap individu, agar dapat menilai performa dari setiap individu.

5. Pemodelan Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah sebuah representasi visual yang digunakan untuk memodelkan rancangan database yang diperlukan oleh sebuah sistem informasi manajemen. Diagram ini menunjukkan hubungan dan relasi antara entitas-entitas yang terlibat dalam sistem informasi yang di rancang dalam sistem ini yang terlihat pada Gambar 14.



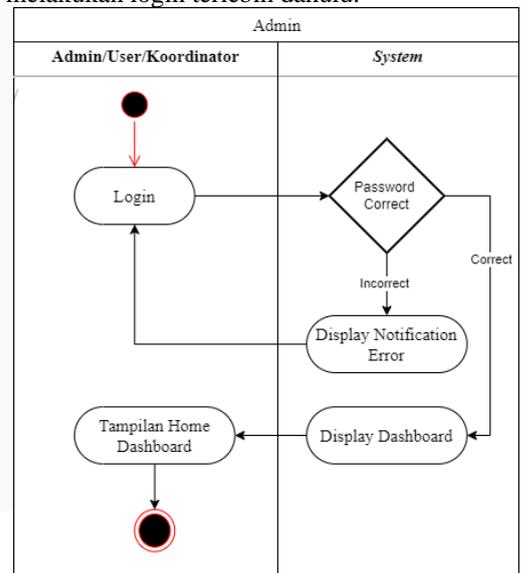
GAMBAR 10

6. Pemodelan Activity Diagram

Activity diagram menggambarkan aliran aktivitas yang dapat dilakukan oleh user pada sistem yang dirancang. Pada activity diagram ini menjelaskan masuk ke dalam sistem dan 6 submenu yang terdapat dalam sistem work report ini yaitu:

a. Login

Merupakan langkah awal dari penggunaan sistem, koordinator, user, dan admin wajib untuk melakukan login terlebih dahulu.



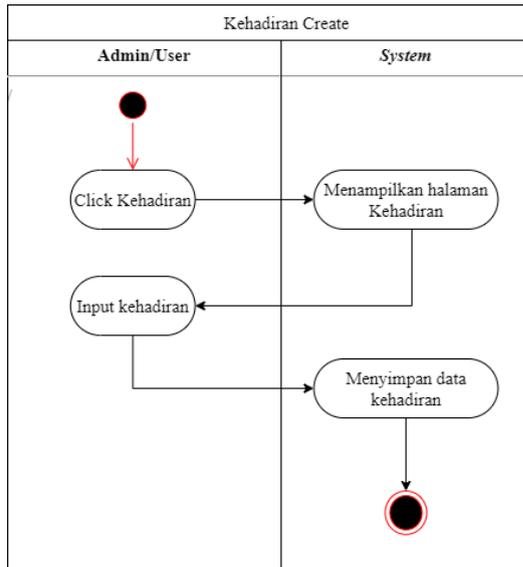
GAMBAR 11

Gambar 11 merupakan alur aktivitas login untuk *user*, koordinator, dan admin. Aktivitas login ini dimulai dari mengakses *website*, kemudian masuk ke halaman *login* dan mengisi email dan password kemudian submit. Jika *password* salah maka akan muncul notifikasi *error* dan jika *password* benar maka akan dilanjutkan ke penampilan dari *dashboard* utama.

b. Kehadiran

Merupakan alur aktivitas setelah sub menu untuk *user*, koordinator, dan admin yang berfungsi

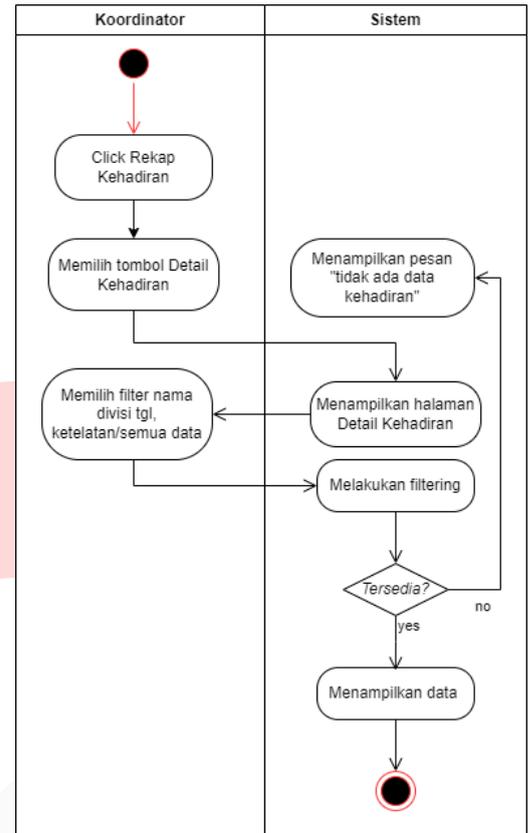
untuk melakukan kehadiran harian dan pemantauan kehadiran.



GAMBAR 12

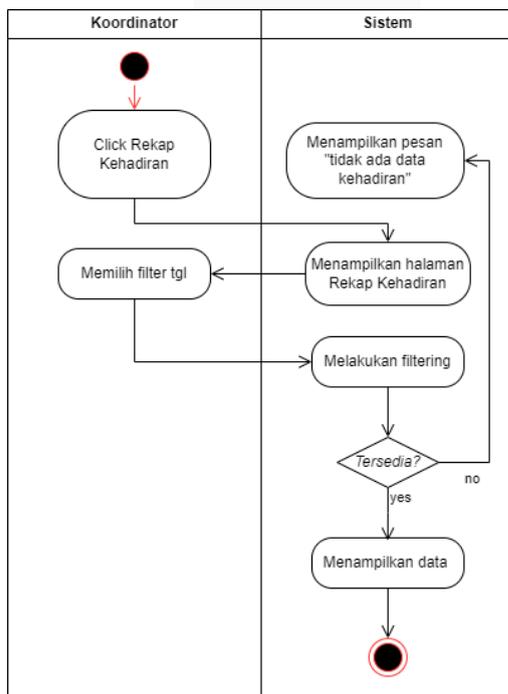
Gambar 12 merupakan alur aktivitas kehadiran untuk *user*, admin. Aktivitas kehadiran ini dimulai dari melakukan klik halaman kehadiran pada *sub menu*. Kemudian sistem akan menampilkan halaman kehadiran. Dan *user* atau admin akan melakukan *input* kehadiran.

dilakukan pemeriksaan. Agar dapat melihat performa dari setiap divisi.



GAMBAR 14

Gambar 14 merupakan alur aktivitas *detail* kehadiran untuk koordinator. Aktivitas kehadiran ini dimulai dari koordinator memilih *sub menu* kehadiran dan menekan tombol *detail* kehadiran. Kemudian sistem akan menampilkan halaman *detail* kehadiran yang dimana koordinator dapat melihat data kehadiran per individu yang sudah ada di *database* dan dapat melakukan filtering untuk dilakukan pemeriksaan. Agar dapat melihat performa dari setiap individu.

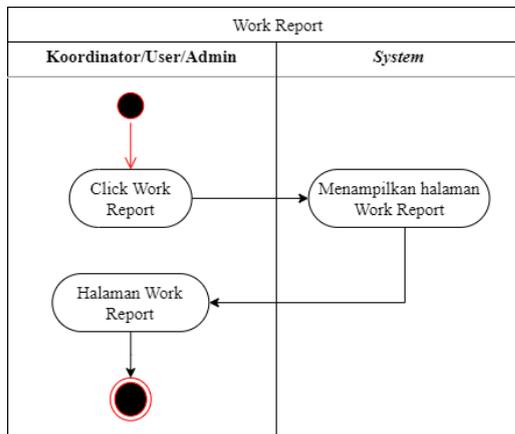


GAMBAR 13

Gambar 13 merupakan alur aktivitas kehadiran untuk koordinator. Aktivitas kehadiran ini dimulai dari koordinator memilih *sub menu* kehadiran. Kemudian sistem akan menampilkan halaman kehadiran yang dimana koordinator dapat melihat data kehadiran yang sudah ada di *database* dan dapat melakukan filtering untuk

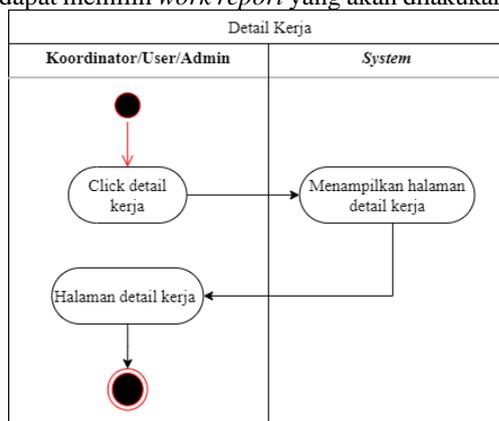
c. *Work Report*

Merupakan sub menu untuk *user*, koordinator, dan admin yang berfungsi untuk melakukan penambahan, pengubahan, penghapusan data *work report* dan pemantauan *work report*.



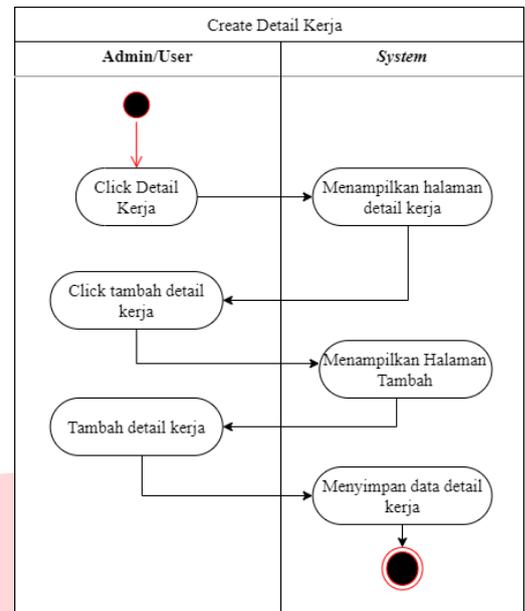
GAMBAR 15

Gambar 15 merupakan alur aktivitas *work report* untuk koordinator, user dan admin. Aktivitas *work report* ini dimulai dari pengguna memilih sub menu *work report*. Kemudian sistem akan menampilkan halaman *work report* yang dimana user dapat melihat data *work report* yang sudah ada di *database*. Page ini berfungsi koordinator melakukan pemeriksaan agar dapat melihat performa dari setiap divisi dan untuk user agar dapat memilih *work report* yang akan dilakukan.



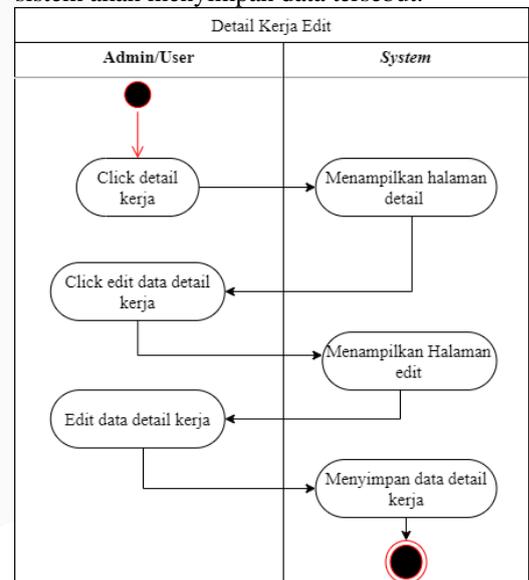
GAMBAR 16

Gambar 16 merupakan alur aktivitas *work report* untuk koordinator, user dan admin. Aktivitas *work report* ini dimulai dari pengguna memilih sub menu *work report* dan memilih tombol aksi detail kerja. Kemudian sistem akan menampilkan halaman *detail* kerja yang dimana user dapat melihat data *detail* kerja yang sudah ada di *database*. Page ini berfungsi koordinator melakukan pemeriksaan agar dapat melihat performa dari setiap individu dan untuk user agar dapat memilih menambah hasil kerja beserta *detail* pekerjaannya.



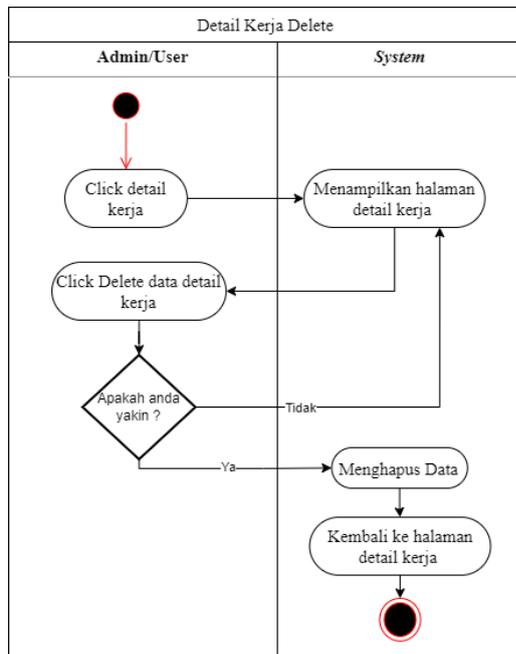
GAMBAR 17

Gambar 17 merupakan alur aktivitas *detail* kerja *create* untuk *user*, admin. Alur ini dimulai dari admin atau user memilih *sub menu* *work report* kemudian oleh sistem ditampilkan halaman *detail* kerja. Dan admin atau *user* memilih tambah data. kemudian sistem menampilkan halaman tambah, setelah itu user menambahkan data detail kerja dan memilih submit kemudian sistem akan menyimpan data tersebut.



GAMBAR 18

Gambar 18 merupakan alur aktivitas *detail* kerja *edit* untuk *user*, admin. Alur ini dimulai dari admin atau user memilih *sub menu* *detail* kerja kemudian oleh sistem ditampilkan halaman *detail* kerja. Dan admin atau *user* memilih mengubah data. kemudian sistem menampilkan halaman edit, setelah itu *user* mengubah data *detail* kerja dan memilih *submit* kemudian sistem akan menyimpan data tersebut.

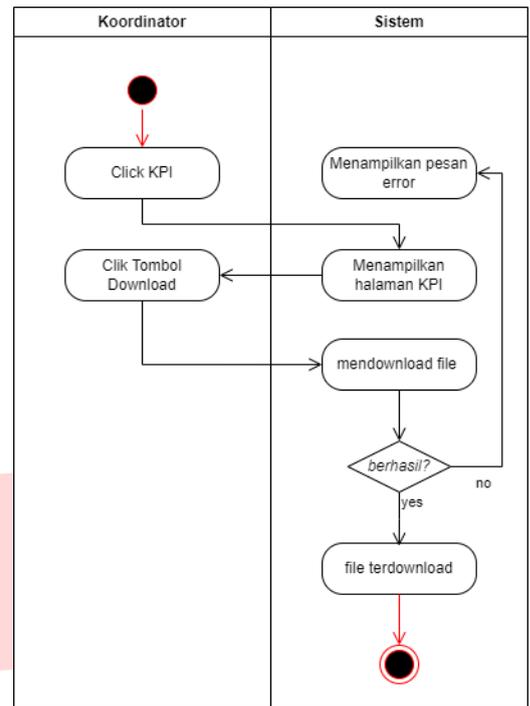


GAMBAR 19

Gambar 19 merupakan alur aktivitas *detail Kerja delete* untuk *user*, *admin*. Alur ini dimulai dari *admin* atau *user* memilih *sub menu detail Kerja* kemudian oleh sistem ditampilkan halaman *detail Kerja*. Dan *admin* atau *user* memilih hapus data. kemudian sistem menampilkan halaman hapus data, setelah itu *user* menghapus data *detail Kerja* dan sistem akan menampilkan notifikasi yakin atau tidak. Jika memilih tidak maka akan kembali ke halaman *detail Kerja* dan jika memilih ya maka sistem akan menghapus data tersebut.

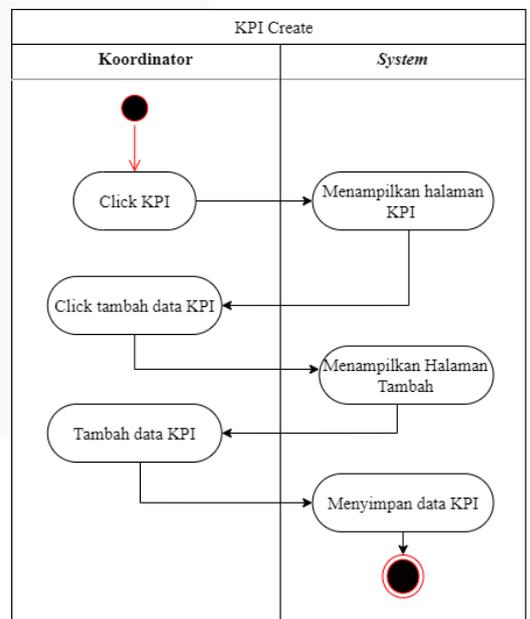
d. KPI

Merupakan sub menu untuk koordinator yang berfungsi untuk menampilkan data, *download file*, penambahan, pengubahan, penghapusan data KPI.



GAMBAR 20

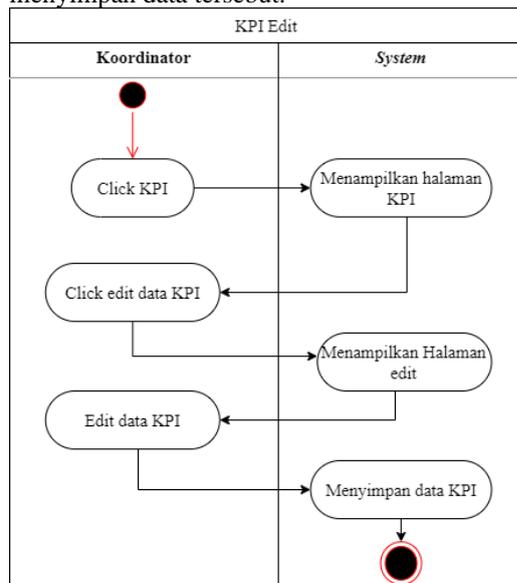
Gambar 20 merupakan alur aktivitas KPI untuk koordinator. Aktivitas KPI ini dimulai dari koordinator memilih sub menu KPI. Kemudian sistem akan menampilkan halaman KPI yang dimana koordinator dapat melihat data KPI yang sudah ada di *database* untuk dilakukan pemeriksaan dan juga melakukan *downloading file* untuk mendapatkan *file excelnya*.



GAMBAR 21

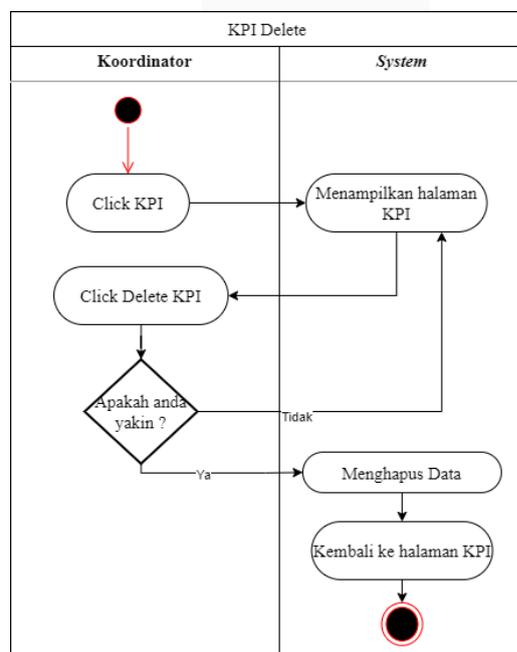
Gambar 21 merupakan alur aktivitas KPI untuk koordinator. Alur ini dimulai dari koordinator memilih *sub menu KPI* kemudian oleh sistem ditampilkan halaman KPI. Dan koordinator memilih tambah data. kemudian sistem menampilkan halaman tambah, setelah itu koordinator menambahkan data KPI dan

memilih submit kemudian sistem akan menyimpan data tersebut.



GAMBAR 22

Gambar 22 merupakan alur aktivitas KPI *edit* untuk koordinator. Alur ini dimulai dari koordinator memilih *sub menu* KPI kemudian oleh sistem ditampilkan halaman KPI. Dan koordinator memilih mengubah data. kemudian sistem menampilkan halaman *edit*, setelah itu user mengubah data KPI dan memilih *submit* kemudian sistem akan menyimpan data tersebut.



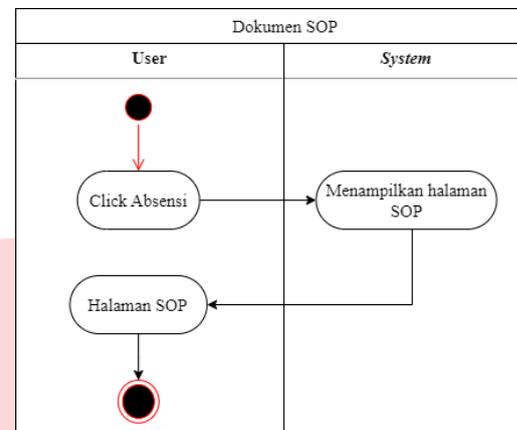
GAMBAR 23

Gambar 23 merupakan alur aktivitas KPI *delete* untuk koordinator. Alur ini dimulai dari koordinator memilih *sub menu* KPI kemudian oleh sistem ditampilkan halaman KPI. Dan koordinator memilih hapus data. kemudian sistem menampilkan halaman hapus data, setelah itu koordinator menghapus data KPI dan sistem akan menampilkan notifikasi yakin atau tidak.

Jika memilih tidak maka akan kembali ke halaman KPI dan jika memilih ya maka sistem akan menghapus data tersebut.

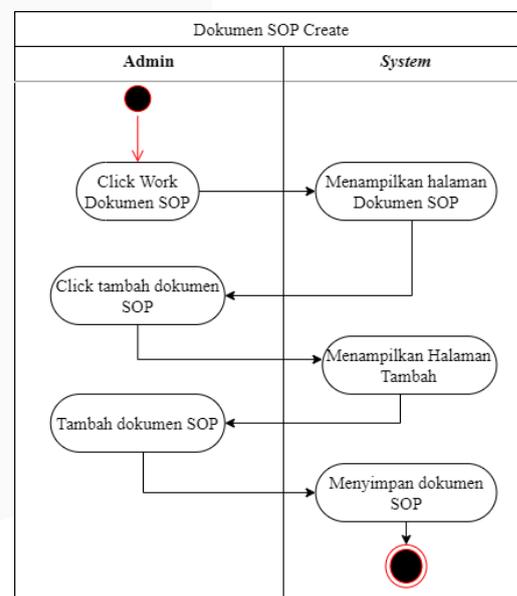
e. Dokumen SOP

Merupakan *sub menu* untuk koordinator yang berfungsi untuk melakukan penambahan, perubahan, penghapusan data dokumen SOP.



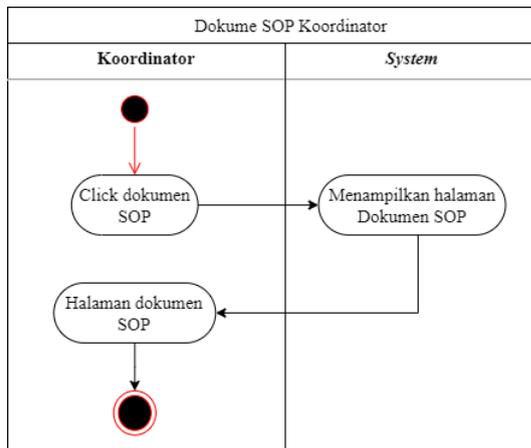
GAMBAR 24

Gambar 24 merupakan alur aktivitas dokumen sop untuk *user*. Aktivitas dokumen sop ini dimulai dari user memilih *sub menu* dokumen sop. Kemudian sistem akan menampilkan halaman dokumen sop yang dimana *user* dapat melihat data dokumen sop yang sudah ada di database untuk dilakukan.



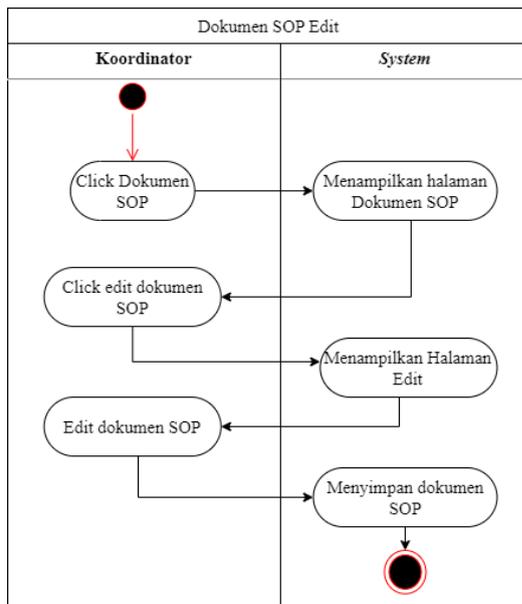
GAMBAR 25

Gambar 25 merupakan alur aktivitas dokumen SOP untuk admin. Alur ini dimulai dari admin memilih *sub menu* dokumen SOP kemudian oleh sistem ditampilkan halaman dokumen SOP. Dan admin memilih tambah data. kemudian sistem menampilkan halaman tambah, setelah itu admin menambahkan data SOP dan memilih submit kemudian sistem akan menyimpan data tersebut.



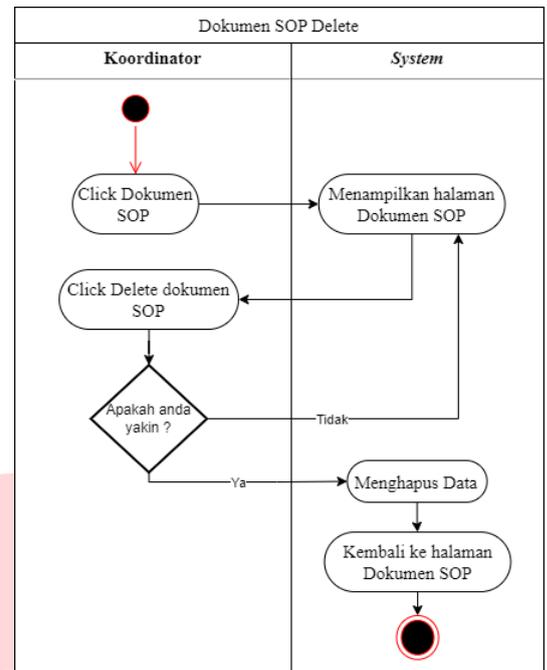
GAMBAR 26

Gambar 26 merupakan alur aktivitas dokumen SOP untuk koordinator. Aktivitas dokumen SOP ini dimulai dari koordinator memilih sub menu dokumen SOP. Kemudian sistem akan menampilkan halaman dokumen SOP yang dimana koordinator dapat melihat data dokumen SOP yang sudah ada di database untuk dilakukan pemeriksaan dokumen SOP. Agar dapat melihat apakah dokumen sudah sesuai.



GAMBAR 27

Gambar 27 merupakan alur aktivitas dokumen SOP edit untuk koordinator. Alur ini dimulai dari koordinator memilih sub menu dokumen SOP kemudian oleh sistem ditampilkan halaman dokumen SOP. Dan koordinator memilih mengubah data. kemudian sistem menampilkan halaman edit, setelah itu user mengubah data dokumen SOP dan memilih submit kemudian sistem akan menyimpan data tersebut.

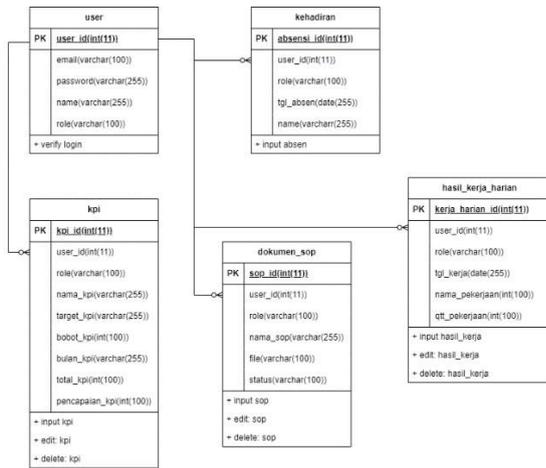


GAMBAR 28

Gambar 28 merupakan alur aktivitas dokumen SOP delete untuk koordinator. Alur ini dimulai dari koordinator memilih sub menu dokumen SOP kemudian oleh sistem ditampilkan halaman dokumen SOP. Dan koordinator memilih hapus data. kemudian sistem menampilkan halaman hapus data, setelah itu koordinator menghapus data dokumen SOP dan sistem akan menampilkan notifikasi yakin atau tidak. Jika memilih tidak maka akan kembali ke halaman dokumen SOP dan jika memilih ya maka sistem akan menghapus data tersebut.

7. Class Diagram

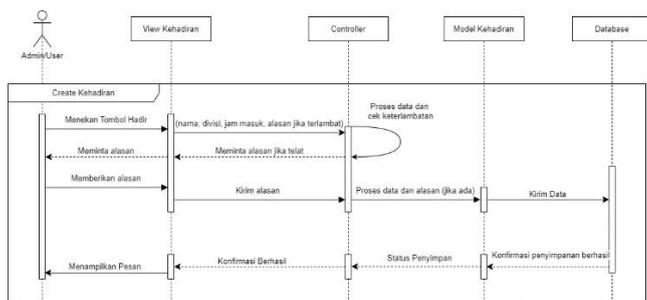
Class diagram adalah jenis diagram yang digunakan dalam pemodelan objek berorientasi kelas untuk menggambarkan struktur statis dari suatu sistem perangkat lunak. Diagram ini menunjukkan kelas-kelas dalam sistem beserta atribut-atributnya, metode-metodenya, dan hubungan antar kelas seperti pewarisan, asosiasi, dan agregasi. Class diagram membantu dalam memvisualisasikan desain sistem secara keseluruhan, memperjelas hubungan antara kelas-kelas, dan memberikan pandangan yang jelas tentang bagaimana objek-objek dalam sistem berinteraksi satu sama lain. Berikut merupakan gambar class diagram untuk sistem work report.



GAMBAR 29

8. Diagram Sequence

Diagram sequence secara umum menunjukkan urutan interaksi antara komponen dalam sebuah sistem. Proses dimulai ketika *aktor* mengirim pesan ke komponen pertama, yang kemudian berinteraksi dengan komponen lainnya, mengirim pesan dan memproses data sesuai urutan waktu. Setelah tugas selesai, komponen terakhir mengembalikan hasil atau status kepada *aktor*. Diagram ini memvisualisasikan alur komunikasi dan aliran data dalam sistem, memastikan setiap langkah dilakukan dengan benar dan efisien.



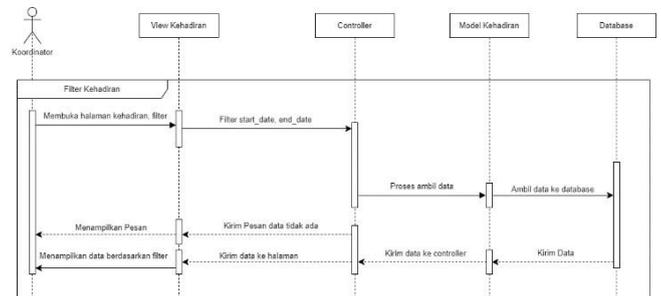
GAMBAR 30

Pada diagram *sequence* di gambar 30, proses dimulai ketika Admin atau User menekan tombol Hadir di halaman View Kehadiran. Proses terbagi menjadi dua langkah utama:

- Menambahkan Kehadiran**, Admin/User menekan tombol Hadir, mengirimkan data seperti nama, divisi, dan jam masuk ke *Controller*. *Controller* memeriksa keterlambatan, dan jika ada, meminta alasan dari pengguna melalui *View Kehadiran*. Pengguna memberikan alasan, yang kemudian dikirim kembali ke *Controller*. *Controller* memproses dan mengirimkan data ke *Model Kehadiran* untuk disimpan di *Database*. Setelah penyimpanan berhasil, konfirmasi dikirim balik ke *Controller* dan kemudian ke *View Kehadiran*, yang menampilkan pesan konfirmasi ke pengguna.
- Konfirmasi dan Penyimpanan**, *Model Kehadiran* menyimpan data kehadiran ke *Database* dan mengirimkan konfirmasi penyimpanan berhasil ke *Controller*. *Controller* mengirimkan konfirmasi keberhasilan kembali ke *View Kehadiran*, yang

kemudian menampilkan pesan konfirmasi kepada Admin/User.

Dengan dua langkah utama ini, sistem memastikan bahwa setiap data kehadiran dicatat dengan benar dan memberikan umpan balik langsung kepada pengguna.

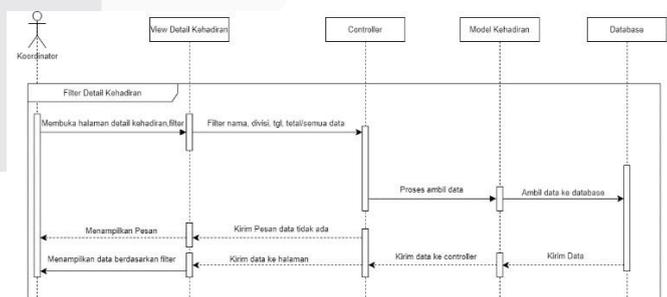


GAMBAR 31

Pada diagram *sequence* di gambar 31 untuk memfilter data kehadiran, proses dimulai ketika Koordinator membuka halaman *View Kehadiran*. Proses ini terdiri dari dua langkah utama:

- Meminta Data Kehadiran Berdasarkan Filter**, Koordinator memilih tanggal awal (*start_date*) dan tanggal akhir (*end_date*) di *View Kehadiran*. *View Kehadiran* mengirim permintaan filter ini ke *Controller*, yang memproses permintaan dan meneruskannya ke *Model Kehadiran* untuk mengambil data dari *Database*. *Model Kehadiran* mengakses *Database* dan mengembalikan data ke *Controller*.
- Menampilkan Data atau Pesan**, *Controller* mengirimkan data yang difilter kembali ke *View Kehadiran*. Jika data ada, *View Kehadiran* menampilkan data yang sesuai dengan filter kepada Koordinator. Jika tidak ada data yang sesuai, *View Kehadiran* menampilkan pesan bahwa data tidak ditemukan.

Dengan dua langkah ini, sistem memungkinkan pengguna untuk memfilter dan melihat data kehadiran berdasarkan rentang tanggal yang dipilih, serta memberikan *feedback* mengenai keberadaan data yang sesuai dengan kriteria filter.



GAMBAR 32

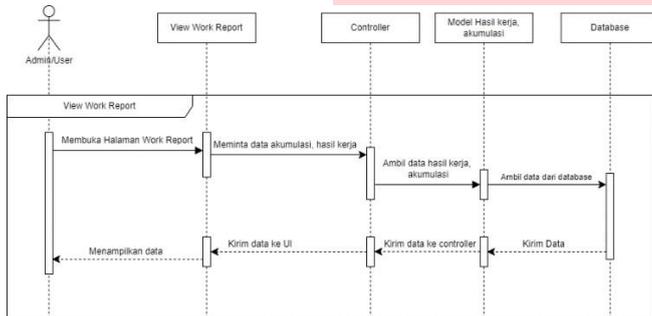
Pada diagram *sequence* di gambar 32 untuk memfilter *detail* kehadiran, proses dimulai ketika Koordinator membuka halaman *View Detail Kehadiran*. Proses ini dibagi menjadi dua langkah utama:

- Meminta Data Detail Kehadiran Berdasarkan Filter**: Koordinator memilih filter seperti nama,

divisi, tanggal, dan *detail* lainnya di *View Detail Kehadiran*. *View* mengirim permintaan ini ke *Controller*, yang memproses dan meneruskannya ke *Model Kehadiran*. *Model* mengambil data dari *Database* sesuai dengan filter yang diberikan dan mengirimkan data kembali ke *Controller*.

2. **Menampilkan Data atau Pesan:** *Controller* mengirim data yang difilter ke *View Detail Kehadiran*. Jika data ada, *View* menampilkan data yang sesuai kepada Koordinator. Jika tidak ada data yang cocok dengan filter, *View* menampilkan pesan bahwa data tidak ditemukan.

Dengan dua langkah ini, sistem memungkinkan Koordinator untuk memfilter dan melihat *detail* kehadiran berdasarkan kriteria yang dipilih, serta memberikan *feedback* langsung mengenai hasil pencarian data.

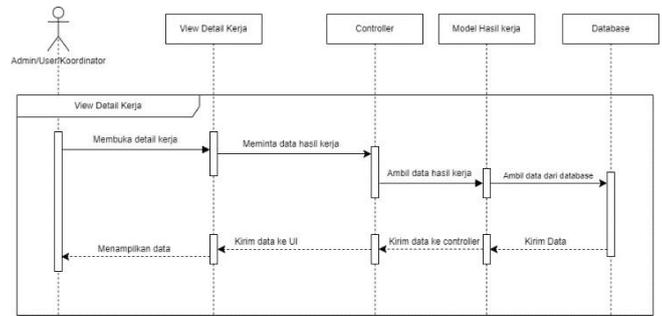


GAMBAR 33

Pada diagram *sequence* di gambar 33 untuk halaman *work report*, proses dimulai dengan Admin atau User membuka halaman *View Work report*. Proses ini dibagi menjadi dua langkah utama:

- Menampilkan Data Work report,** Admin/User/Koordinator membuka halaman kerja, yang meminta data *work report* dari *Controller*. *Controller* mengambil data *work report* melalui Model Hasil Kerja dan akumulasi dari *Database*. Setelah data diperoleh, Model Hasil Kerja mengirimkan data ke *Controller*, yang kemudian mengirim data ke *View Work report*.
- Menampilkan Data kepada Pengguna,** *View Work report* menampilkan data *work report* yang diperoleh kepada Admin/User/Koordinator, sehingga pengguna dapat melihat informasi *work report* secara *real-time*.

Dengan dua langkah ini, sistem memastikan bahwa data *work report* diambil dari *Database* dan ditampilkan kepada pengguna secara efektif.

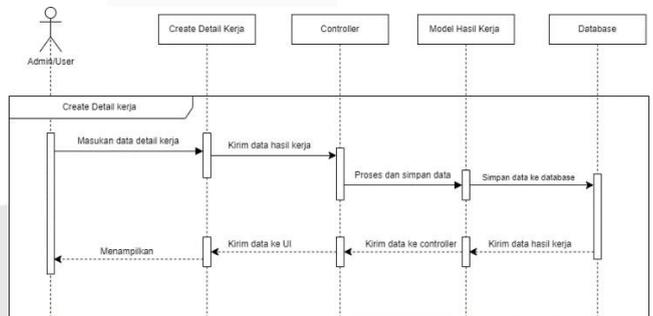


GAMBAR 34

Pada diagram *sequence* di gambar 34 untuk halaman *Detail Kerja*, proses dimulai dengan Admin atau User membuka halaman *View Detail Kerja*. Proses ini dibagi menjadi dua langkah utama:

- Menampilkan Data Detail Kerja,** Admin/User/Koordinator membuka halaman kerja, yang meminta data *work report* dari *Controller*. *Controller* mengambil data *work report* melalui Model Hasil Kerja dari *Database*. Setelah data diperoleh, Model Hasil Kerja mengirimkan data ke *Controller*, yang kemudian mengirim data ke *View Detail Kerja*.
- Menampilkan Data kepada Pengguna,** *View Detail Kerja* menampilkan data *work report* yang diperoleh kepada Admin/User, sehingga pengguna dapat melihat informasi *Detail Kerja* secara *real-time*.

Dengan dua langkah ini, sistem memastikan bahwa data *Detail Kerja* diambil dari *Database* dan ditampilkan kepada pengguna secara efektif.



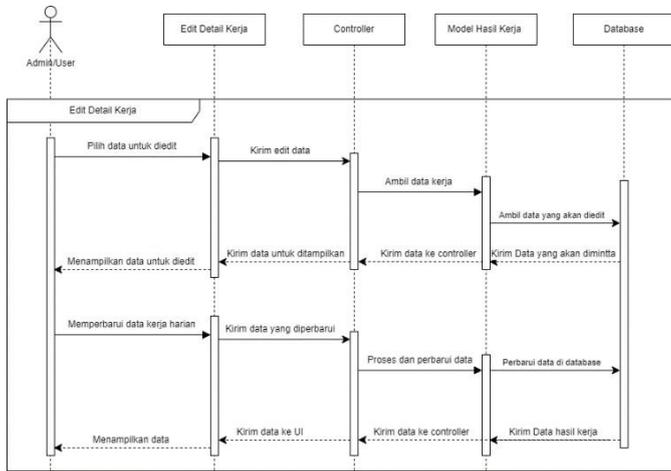
GAMBAR 35

Pada diagram *sequence* di gambar 35 untuk pembuatan data *detail* kerja, proses dimulai ketika Admin atau User memasukkan data *detail* kerja di halaman *View Detail kerja*. Proses ini dibagi menjadi dua langkah utama:

- Menambahkan Data Detail kerja,** Admin/User memasukkan data *detail* kerja yang kemudian dikirim oleh *View* ke *Controller*. *Controller* memproses data dan meneruskannya ke Model Hasil Kerja untuk disimpan ke *Database*. Setelah data berhasil disimpan, *Database* mengirimkan konfirmasi penyimpanan ke Model Hasil Kerja, yang kemudian mengembalikan status penyimpanan ke *Controller*.

- b) **Konfirmasi dan Tampilan, Controller** mengirimkan status keberhasilan penyimpanan kembali ke *View Detail* kerja, yang kemudian menampilkan pesan konfirmasi kepada Admin/User bahwa data *detail* kerja telah berhasil ditambahkan.

Dengan dua langkah ini, sistem memastikan bahwa data *detail* kerja yang dimasukkan oleh pengguna dicatat dan dikonfirmasi secara efisien, memberikan umpan balik *real-time* kepada pengguna.

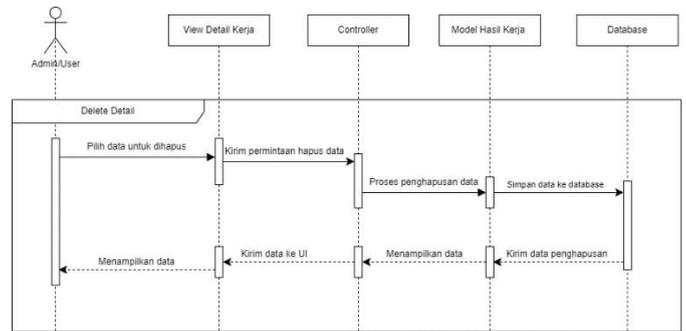


GAMBAR 36

Pada diagram sequence di gambar 36 untuk mengedit data *detail* kerja, proses dimulai ketika Admin atau User memilih data untuk diedit di halaman *View Detail* kerja. Proses ini terdiri dari dua langkah utama:

- Memilih dan Menampilkan Data untuk Diedit,** Admin/User memilih data *detail* kerja yang akan diedit di *View*. *View* mengirim permintaan edit data ke *Controller*, yang kemudian mengambil data dari *Model Hasil Kerja*. *Model* meminta data dari *Database* dan mengirimkan data yang akan diedit kembali ke *Controller*, yang lalu mengirimkannya ke *View* untuk ditampilkan dan diedit oleh pengguna.
- Memperbarui dan Menyimpan Data yang Diedit,** Setelah pengguna memperbarui data di *View*, *View* mengirimkan data yang diperbarui ke *Controller*. *Controller* memproses dan memperbarui data melalui *Model Hasil Kerja* di *Database*. Setelah pembaruan berhasil, *Database* mengirim konfirmasi ke *Model*, yang kemudian mengirimkan status keberhasilan ke *Controller*. *Controller* akhirnya mengirimkan status pembaruan kembali ke *View*, yang menampilkan pesan konfirmasi kepada Admin/User bahwa data telah berhasil diperbarui.

Dengan dua langkah ini, sistem memastikan bahwa data *detail* kerja yang dipilih oleh pengguna dapat diedit dan disimpan kembali dengan benar, serta memberikan umpan balik langsung kepada pengguna mengenai status pembaruan.

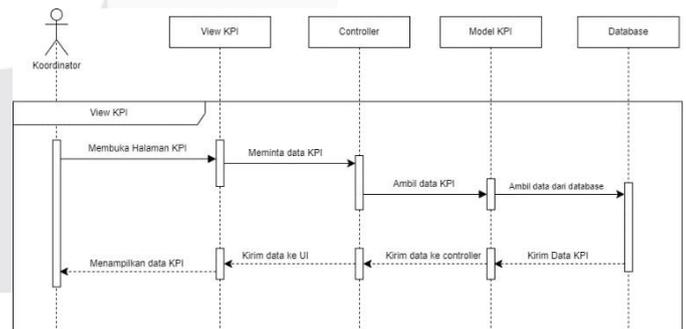


GAMBAR 37

Pada diagram sequence di gambar 37 untuk menghapus data *detail* kerja, proses dimulai ketika Admin atau User memilih data yang akan dihapus pada halaman *View Detail* kerja. Proses ini dibagi menjadi dua langkah utama:

- Memilih dan Mengirim Permintaan Penghapusan Data,** Admin/User memilih data untuk dihapus di *View*, yang kemudian mengirimkan permintaan penghapusan data ke *Controller*. *Controller* memproses permintaan ini dan meneruskannya ke *Model Hasil Kerja*. *Model* menginstruksikan *Database* untuk menghapus data yang dipilih.
- Mengonfirmasi dan Menampilkan Penghapusan,** Setelah data dihapus, *Database* mengirimkan konfirmasi penghapusan kembali ke *Model Hasil Kerja*. *Model* mengirim status penghapusan ke *Controller*, yang kemudian mengirimkan konfirmasi tersebut ke *View*. *View* akhirnya menampilkan pesan konfirmasi kepada Admin/User bahwa data *detail* kerja telah berhasil dihapus.

Dengan dua langkah ini, sistem memastikan bahwa data *detail* kerja dapat dipilih, diproses untuk penghapusan, dan pengguna mendapatkan umpan balik langsung mengenai status penghapusan data.



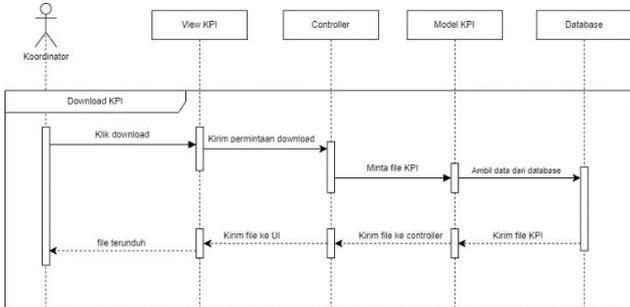
GAMBAR 38

Pada diagram sequence di gambar 38 untuk menampilkan data KPI, proses dimulai ketika Koordinator membuka halaman *View KPI*. Proses ini terdiri dari dua langkah utama:

- Meminta dan Mengambil Data KPI,** Koordinator membuka halaman KPI di *View*, yang kemudian meminta data KPI dari *Controller*. *Controller* mengambil data KPI melalui *Model KPI* dari *Database*. Setelah data diperoleh, *Model KPI* mengirimkannya kembali ke *Controller*.

- b) **Menampilkan Data KPI**, *Controller* meneruskan data yang diperoleh ke *View KPI*. *View KPI* kemudian menampilkan data KPI kepada Koordinator.

Dengan dua langkah ini, sistem memastikan bahwa data KPI diambil dari *Database* dan ditampilkan kepada pengguna secara efisien dan real-time.

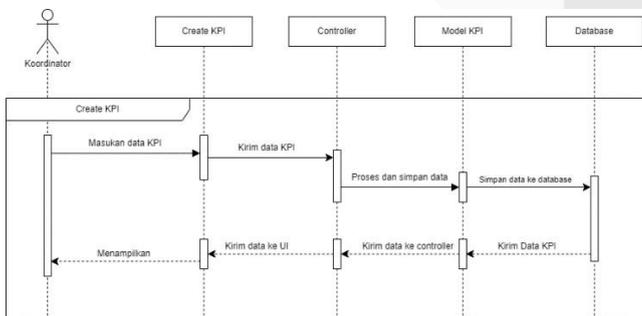


GAMBAR 39

Pada diagram sequence di gambar 39 untuk mengunduh file KPI, proses dimulai ketika Koordinator mengklik tombol "Download" di halaman *View KPI*. Proses ini terdiri dari dua langkah utama:

1. **Mengirim Permintaan Pengunduhan**, Koordinator mengklik tombol "Download", dan *View* mengirimkan permintaan pengunduhan ke *Controller*. *Controller* memproses permintaan ini dengan meminta file KPI dari *Model KPI*. *Model* kemudian mengambil file dari *Database* atau sistem penyimpanan data.
2. **Mengirim dan Menyediakan File untuk Diunduh**, Setelah file diambil, *Model KPI* mengirimkan file kembali ke *Controller*. *Controller* meneruskan file tersebut ke *View KPI*, yang kemudian menyediakan file untuk diunduh oleh Koordinator, sehingga file KPI terunduh ke perangkat pengguna.

Dengan dua langkah ini, sistem memastikan bahwa file KPI diambil dari *Database* dan disediakan untuk diunduh oleh pengguna secara efisien, memberikan *feedback* yang jelas dan cepat.

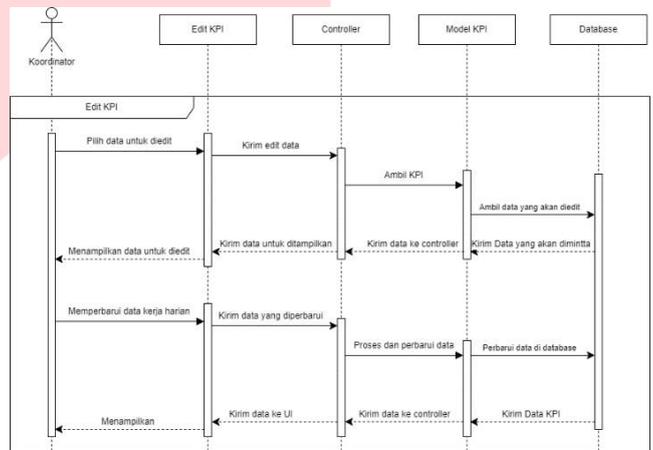


GAMBAR 40

Pada diagram sequence di gambar 40 untuk pembuatan data KPI, proses dimulai ketika Koordinator memasukkan data KPI pada halaman *View KPI*. Proses ini terbagi menjadi dua langkah utama:

- a) **Memasukkan dan Menyimpan Data KPI**, Koordinator memasukkan data KPI di *View*, yang kemudian mengirimkan data tersebut ke *Controller*. *Controller* memproses dan meneruskannya ke *Model KPI* untuk disimpan ke *Database*. Setelah penyimpanan berhasil, *Database* mengirimkan konfirmasi penyimpanan kembali ke *Model KPI*, yang mengirim status tersebut ke *Controller*.
- b) **Konfirmasi dan Tampilan**, *Controller* mengirimkan status penyimpanan kembali ke *View KPI*. *View KPI* kemudian menampilkan pesan konfirmasi kepada Koordinator bahwa data KPI telah berhasil disimpan.

Dengan dua langkah ini, sistem memastikan bahwa data KPI yang dimasukkan oleh pengguna dicatat di *Database* dan pengguna menerima umpan balik langsung mengenai status penyimpanan.

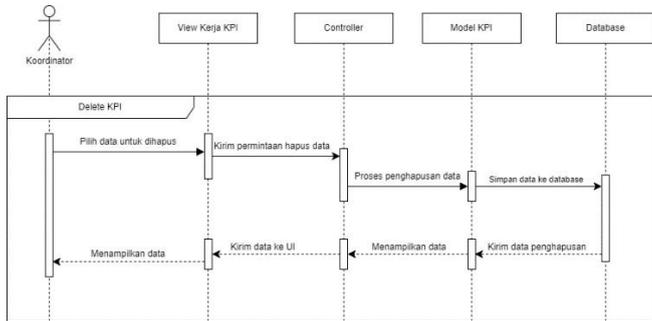


GAMBAR 41

Pada diagram sequence di gambar 41 untuk mengedit data KPI, proses dimulai ketika Koordinator memilih data KPI yang akan diedit di halaman *View KPI*. Proses ini terbagi menjadi dua langkah utama:

- a) **Memilih dan Menampilkan Data untuk Diedit**, Koordinator memilih data yang akan diedit pada *View*, yang kemudian mengirim permintaan edit ke *Controller*. *Controller* mengambil data KPI dari *Model KPI*, yang mengakses *Database* untuk mendapatkan data yang akan diedit. Setelah data diperoleh, *Model* mengirimkannya kembali ke *Controller*, yang lalu mengirim data tersebut ke *View* untuk ditampilkan dan diedit oleh Koordinator.
- b) **Memperbarui dan Menyimpan Data**, Setelah Koordinator memperbarui data KPI di *View*, data yang diperbarui dikirimkan kembali ke *Controller*. *Controller* memproses data tersebut dan meneruskannya ke *Model KPI* untuk memperbarui data di *Database*. Setelah pembaruan berhasil, *Database* mengirim konfirmasi ke *Model*, yang kemudian mengirimkan status pembaruan ke *Controller*. *Controller* mengirimkan status tersebut kembali ke *View*, yang menampilkan pesan konfirmasi bahwa data KPI telah berhasil diperbarui.

Dengan dua langkah ini, sistem memastikan bahwa data KPI yang dipilih oleh pengguna dapat diedit dan disimpan kembali dengan benar, serta memberikan *feedback* langsung mengenai status pembaruan kepada pengguna.

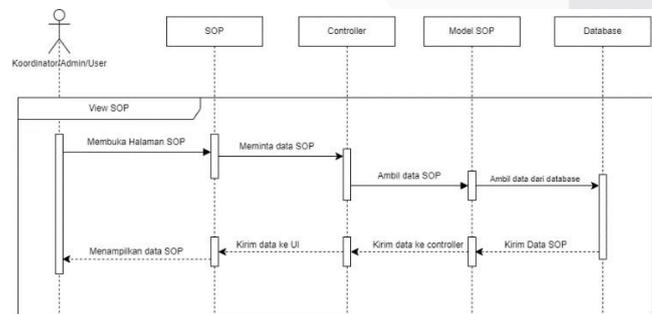


GAMBAR 42

Pada diagram sequence di gambar 42 untuk menghapus data KPI, proses dimulai ketika Koordinator memilih data yang akan dihapus di halaman *View Kerja KPI*. Proses ini terbagi menjadi dua langkah utama:

- a) **Mengirim Permintaan Penghapusan**, Koordinator memilih data KPI yang akan dihapus melalui *View*. *View* mengirimkan permintaan penghapusan ke *Controller*. *Controller* memproses permintaan ini dan meneruskannya ke *Model KPI*, yang kemudian menginstruksikan *Database* untuk menghapus data tersebut.
- b) **Konfirmasi dan Tampilan Penghapusan**, Setelah data berhasil dihapus, *Database* mengirimkan konfirmasi penghapusan kembali ke *Model KPI*. *Model* mengirim status penghapusan ini ke *Controller*, yang kemudian mengirimkannya kembali ke *View Kerja KPI*. *View* akhirnya menampilkan pesan konfirmasi kepada Koordinator bahwa data KPI telah berhasil dihapus.

Dengan dua langkah ini, sistem memastikan bahwa data KPI dapat dipilih dan dihapus dengan benar dari *Database*, serta memberikan *feedback* langsung kepada pengguna mengenai status penghapusan data.



GAMBAR 43

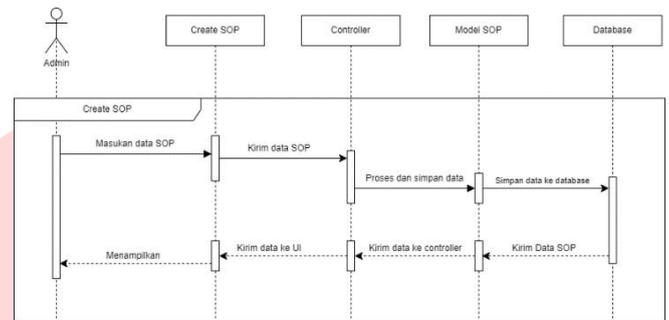
Pada diagram sequence di gambar 43 untuk menampilkan data SOP, proses dimulai ketika Koordinator, Admin, atau User membuka halaman *View SOP*. Proses ini terdiri dari dua langkah utama:

- a) **Meminta dan Mengambil Data SOP**, Koordinator/Admin/User membuka halaman SOP

di *View*, yang meminta data SOP dari *Controller*. *Controller* mengambil data melalui *Model SOP* dari *Database*. Setelah data diambil, *Model SOP* mengirimkannya kembali ke *Controller*.

- b) **Menampilkan Data SOP**, *Controller* mengirim data yang diperoleh ke *View SOP*. *View SOP* kemudian menampilkan data SOP kepada Koordinator, Admin, atau User.

Dengan dua langkah ini, sistem memastikan bahwa data SOP diambil dari *Database* dan ditampilkan kepada pengguna secara efisien dan *real-time*.

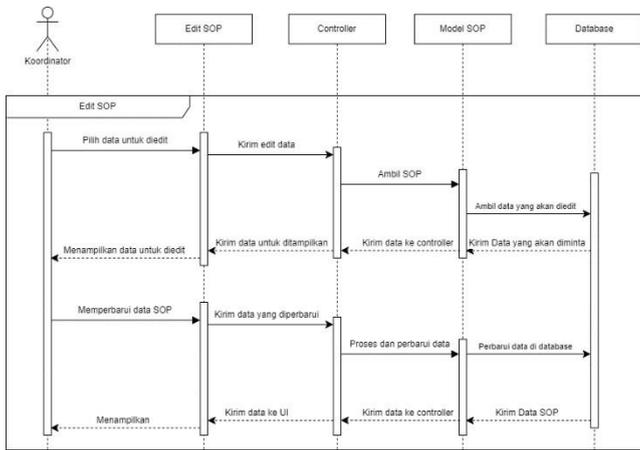


GAMBAR 44

Pada diagram sequence di gambar 44 untuk pembuatan data SOP, proses dimulai ketika Admin memasukkan data SOP di halaman *View SOP*. Proses ini terdiri dari dua langkah utama:

- a) **Memasukkan dan Menyimpan Data SOP**, Admin memasukkan data SOP di *View*, yang kemudian mengirimkan data tersebut ke *Controller*. *Controller* memproses data dan meneruskannya ke *Model SOP* untuk disimpan ke *Database*. Setelah data berhasil disimpan, *Database* mengirimkan konfirmasi penyimpanan ke *Model SOP*, yang kemudian mengirim status tersebut ke *Controller*.
- b) **Konfirmasi dan Tampilan**, *Controller* mengirimkan status penyimpanan kembali ke *View SOP*. *View SOP* kemudian menampilkan pesan konfirmasi kepada Admin bahwa data SOP telah berhasil disimpan.

Dengan dua langkah ini, sistem memastikan bahwa data SOP yang dimasukkan oleh Admin dicatat di *Database* dan memberikan *feedback* langsung mengenai status penyimpanan kepada pengguna.

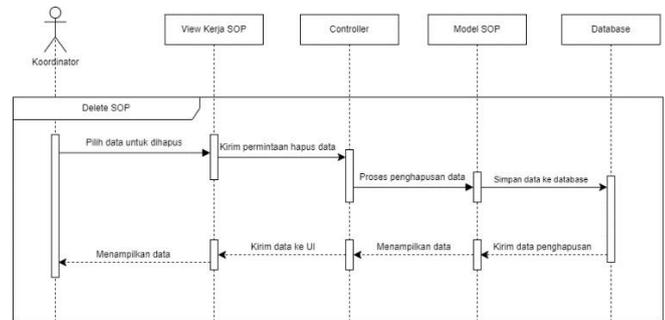


GAMBAR 45

Pada diagram sequence di gambar 45 untuk mengedit data SOP, proses dimulai ketika Koordinator memilih data SOP yang akan diedit di halaman *View SOP*. Proses ini dibagi menjadi dua langkah utama:

- Memilih dan Menampilkan Data untuk Diedit**, Koordinator memilih data yang akan diedit di *View*, yang kemudian mengirimkan permintaan edit ke *Controller*. *Controller* mengambil data dari *Model SOP*, yang mengakses *Database* untuk mendapatkan data yang akan diedit. Setelah data diperoleh, *Model* mengirimkannya kembali ke *Controller*, yang kemudian mengirim data tersebut ke *View* untuk ditampilkan dan diedit oleh Koordinator.
- Memperbarui dan Menyimpan Data**, Setelah Koordinator memperbarui data SOP di *View*, data yang diperbarui dikirimkan kembali ke *Controller*. *Controller* memproses data tersebut dan meneruskannya ke *Model SOP* untuk memperbarui data di *Database*. Setelah pembaruan berhasil, *Database* mengirim konfirmasi ke *Model*, yang kemudian mengirimkan status pembaruan ke *Controller*. *Controller* mengirimkan status tersebut kembali ke *View*, yang menampilkan pesan konfirmasi bahwa data SOP telah berhasil diperbarui.

Dengan dua langkah ini, sistem memastikan bahwa data SOP yang dipilih oleh pengguna dapat diedit dan disimpan kembali dengan benar, serta memberikan *feedback* langsung mengenai status pembaruan kepada pengguna.



GAMBAR 46

Pada diagram sequence di gambar 46 untuk menghapus data SOP, proses dimulai ketika Koordinator memilih data yang akan dihapus di halaman *View Kerja SOP*. Proses ini terdiri dari dua langkah utama:

- Mengirim Permintaan Penghapusan**, Koordinator memilih data yang akan dihapus di *View*, yang kemudian mengirimkan permintaan penghapusan data ke *Controller*. *Controller* memproses permintaan ini dan meneruskannya ke *Model SOP*, yang kemudian menginstruksikan *Database* untuk menghapus data tersebut.
- Konfirmasi dan Tampilan Penghapusan**, Setelah data dihapus, *Database* mengirimkan konfirmasi penghapusan kembali ke *Model SOP*. *Model* mengirim status penghapusan ini ke *Controller*, yang kemudian mengirimkannya kembali ke *View Kerja SOP*. *View* menampilkan pesan konfirmasi kepada Koordinator bahwa data SOP telah berhasil dihapus.

Dengan dua langkah ini, sistem memastikan bahwa data SOP dapat dipilih dan dihapus dengan benar dari *Database*, serta memberikan *feedback* langsung kepada pengguna mengenai status penghapusan data.

B. Hasil Rancangan

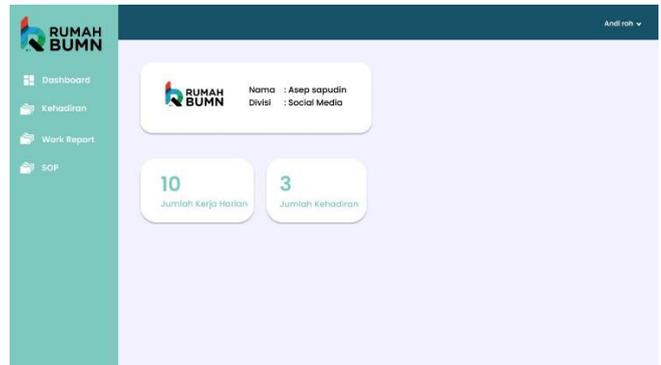
1. Pemodelan *Mock-up*

Pemodelan *mock-up* adalah proses pembuatan model visual yang bertujuan untuk menunjukkan bagaimana produk atau proyek akan terlihat dan berfungsi saat selesai. *Mock-up* sering digunakan dalam desain produk, aplikasi perangkat lunak, dan situs *website* untuk memberikan gambaran yang jelas kepada tim pengembang dan klien tentang tampilan akhir produk. Pemodelan ini mencakup elemen-elemen seperti tata letak, warna, jenis huruf, dan fungsi interaktif dasar, tetapi tidak melibatkan implementasi kode atau *detail* teknis yang lebih mendalam. Dengan menggunakan *mock-up*, tim dapat mengidentifikasi masalah desain lebih awal dalam proses pengembangan dan melakukan perubahan yang diperlukan sebelum memulai produksi atau pengembangan yang lebih lanjut.



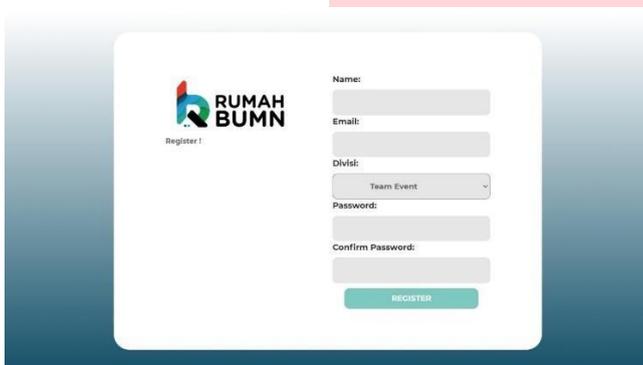
GAMBAR 47

Gambar 47 merupakan tampilan mockup halaman *login*, terdapat logo rumah BUMN dan kalimat *welcoming*. Dan juga terdapat form yang terdiri dari email, password, dan tombol masuk.



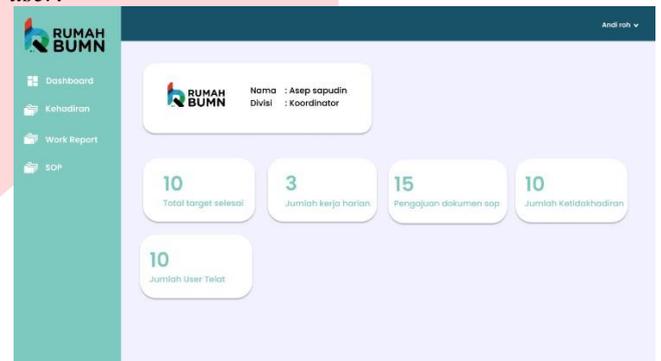
GAMBAR 50

Gambar 50 merupakan tampilan dari halaman *dashboard* untuk *intern*. Yang terdiri dari *sidebar* yang terdapat beberapa *menu* dan logo. Kemudian terdapat nama dari *user*, dan divisi *user*.



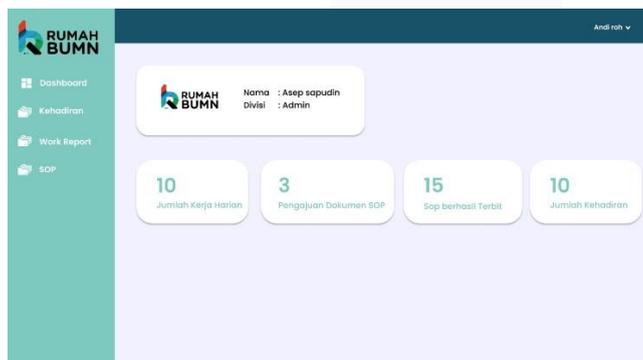
GAMBAR 48

Gambar 48 merupakan tampilan *mockup* halaman *register*, terdapat logo rumah BUMN dan kalimat *welcoming*. Dan juga terdapat form yang terdiri dari email, password, dan tombol masuk.



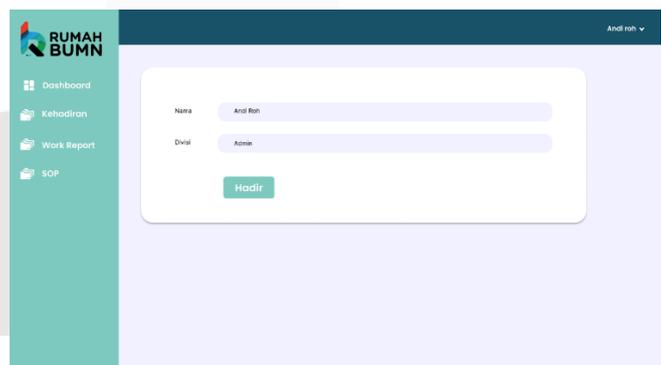
GAMBAR 51

Gambar 51 merupakan tampilan dari halaman *dashboard* untuk koordinator. Yang terdiri dari *sidebar* yang terdapat beberapa *menu* dan logo. Kemudian terdapat nama dari *user*, divisi *user*, kemudian terdapat card yang berfungsi untuk memberi informasi jumlah kehadiran, jumlah SOP, dan jumlah *work report*.



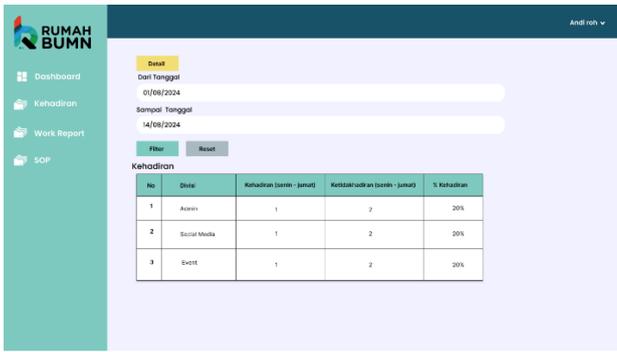
GAMBAR 49

Gambar 49 merupakan tampilan dari halaman *dashboard* untuk admin. Yang terdiri dari *sidebar* yang terdapat beberapa *menu* dan logo. Kemudian terdapat nama dari pengguna, dan divisi *user*.



GAMBAR 52

Gambar 52 merupakan tampilan dari halaman kehadiran untuk *intern* dan admin. Yang terdiri dari *sidebar* yang terdapat beberapa *menu* dan logo. Kemudian terdapat nama dari *user*, dan divisi *user*, dan tombol untuk melakukan kehadiran.



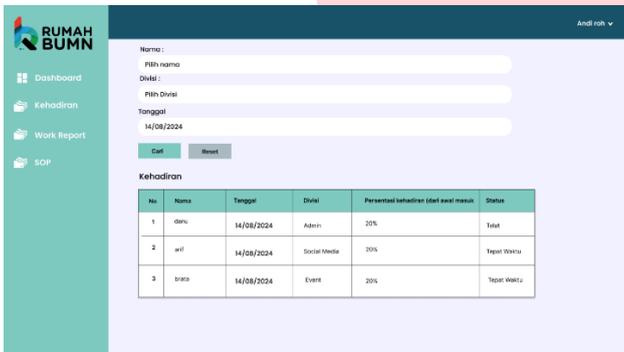
GAMBAR 53

Gambar 53 merupakan tampilan dari halaman kehadiran untuk koordinator. Yang terdiri dari *sidebar* yang terdapat beberapa menu dan logo. Kemudian terdapat filter dan tabel yang berisi divisi, kehadiran, ketidakhadiran, % kehadiran.



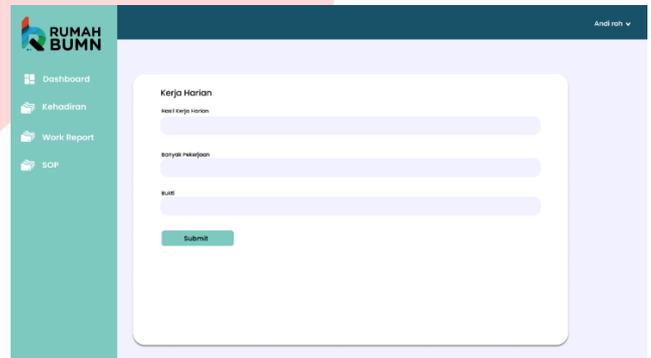
GAMBAR 56

Gambar 56 merupakan tampilan dari halaman *work report* untuk koordinator. Yang terdiri dari *sidebar* yang terdapat beberapa menu dan logo. Kemudian terdapat tabel akumulasi *team* yang berisi nama target, divisi, tenggat waktu, target tercapai, persentase capaian.



GAMBAR 54

Gambar 54 merupakan tampilan dari halaman *detail* kehadiran untuk koordinator. Yang terdiri dari *sidebar* yang terdapat beberapa menu dan logo. Kemudian terdapat filter dan tabel yang berisi divisi, kehadiran, ketidakhadiran, % kehadiran, Status.



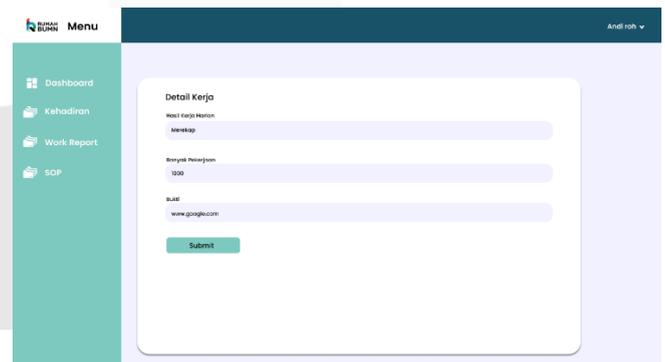
GAMBAR 57

Gambar 57 merupakan halaman mockup tambah data dari tabel hasil *work report* Yang terdiri dari *sidebar* yang terdapat beberapa menu dan logo. Dan juga berisi *form* hasil *work report*, banyak pekerjaan, bukti, dan tombol *submit*.



GAMBAR 55

Gambar 55 merupakan tampilan dari halaman *work report* untuk *intern* dan admin. Yang terdiri dari *sidebar* yang terdapat beberapa menu dan logo. Kemudian terdapat tabel akumulasi *team* yang berisi nama target, divisi, tenggat waktu, target tercapai, persentase capaian.



GAMBAR 58

Gambar 58 merupakan halaman mockup *edit* data dari tabel hasil *work report* Yang terdiri dari *sidebar* yang terdapat beberapa menu dan logo. Dan juga berisi *form* hasil *work report*, banyak pekerjaan, bukti, dan tombol *submit*.



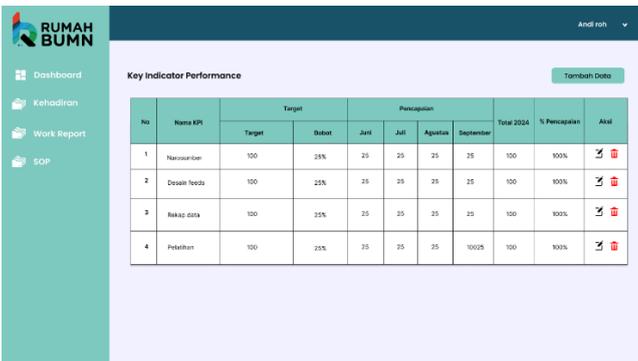
GAMBAR 59

Gambar 59 merupakan tampilan dari halaman dokumen SOP untuk koordinator. Yang terdiri dari *sidebar* yang terdapat beberapa menu dan logo. Kemudian terdapat tabel dokumen SOP yang berisi nama SOP, divisi, status, dan aksi. Untuk perbedaan koordinator dengan admin dan *intern* adalah untuk di admin terdapat penambahan tombol untuk tambah SOP dan tidak adanya tombol terbitkan dan hapus, kemudian untuk *user* biasa kolom aksi hanya terdapat tombol download saja dan tidak bisa menambahkan SOP.



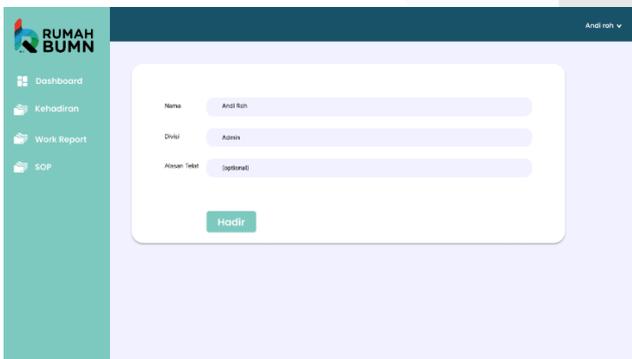
GAMBAR 62

Gambar 62 melakukan penambahan tombol detail kerja jika agar dapat melihat detail kerja.



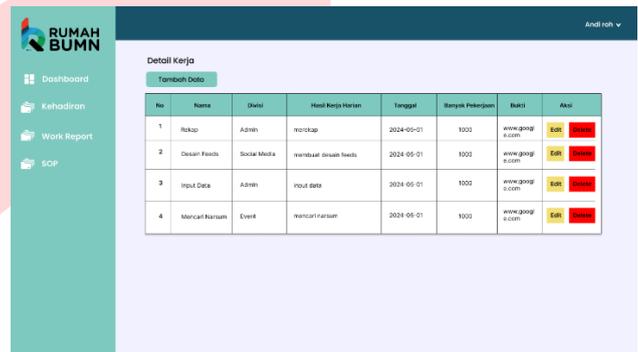
GAMBAR 60

Gambar 60 merupakan tampilan dari halaman *key indicator performance* untuk koordinator. Yang terdiri dari *sidebar* yang terdapat beberapa menu dan logo. Kemudian terdapat tabel *key performance indicator* yang berisi nama KPI, target, bobot, target perbulan, total tahunan, dan persen pencapaian.



GAMBAR 61

Gambar 61 melakukan penambahan form alasan jika *intern/admin* terlambat untuk melakukan kehadiran.



GAMBAR 63

Gambar 63 melakukan penambahan halaman detail kerja untuk *user* admin dan intern yang terdiri dari *sidebar* yang terdapat beberapa menu dan logo. Kemudian terdapat tabel hasil *detail* kerja yang berisi nama, divisi, hasil kerja, tanggal, banyak pekerjaan, bukti, aksi yang berisi tombol untuk melakukan *editing* dan *delete*. Dan untuk koordinator terdapat perbedaan sedikit yaitu tidak adanya tombol edit, dan delete.

V. KESIMPULAN

Dalam tugas akhir yang berjudul PERANCANGAN SISTEM INFORMASI MANAJEMEN *WORK REPORT* RUMAH BUMN KOTA BANDUNG BERBASIS WEBSITE MENGGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT*, telah berhasil dirancang sebuah sistem informasi yang mengintegrasikan berbagai fungsi manajemen dan pelaporan kinerja karyawan. Sistem ini dibagi menjadi tiga peran utama yang mencakup Koordinator, Admin, dan *User*, dengan masing-masing peran memiliki akses dan tanggung jawab yang jelas. Koordinator memiliki kemampuan untuk mengawasi kehadiran dan mengelola kinerja *work report* dari Admin dan *User*, serta memiliki wewenang untuk menambahkan, mengesahkan, dan mengedit KPI serta dokumen SOP yang telah diinput oleh Admin. Di sisi lain, Admin bertugas untuk menginput kehadiran, melakukan input, pengeditan, dan penghapusan laporan kerja, serta bertanggung jawab atas input data dokumen SOP. *User*, yang mungkin terbagi dalam divisi seperti *Event* dan *Design Content* untuk media sosial, fokus pada pengelolaan kehadiran dan pembuatan, pengeditan, serta penghapusan laporan kerja sesuai dengan tugas dan fungsi divisi mereka. Sistem ini dilengkapi dengan fitur-fitur

yang mendukung efisiensi kerja dan transparansi laporan, termasuk dashboard khusus untuk masing-masing peran yang memudahkan pengawasan dan evaluasi kinerja secara real-time. Pada fase pengujian, sistem telah menunjukkan performa yang stabil dan mampu menjalankan semua fungsi sesuai dengan spesifikasi yang ditetapkan. Hasil pengujian menegaskan bahwa sistem ini efektif dalam memfasilitasi pengelolaan *work report* dan dapat menjadi alat bantu yang efisien untuk meningkatkan produktivitas serta kualitas pengelolaan sumber daya manusia di RUMAH BUMN Kota Bandung.

REFERENSI

- Al-Fedaghi, S. (2021). UML Sequence Diagram: An Alternative Model. *International Journal of Advanced Computer Science and Applications*, Vol. 12, No. 5.
- Budiarto. (2017). *Key Performance Indicator*. Depok: Huta .
- Dennis, Wixom , & Tegarden . (2015). *Systems Analysis And Design: An Object-Oriented Approach With UML*. Hoboken: Wiley.
- Grechanik, M., McKinley, K. S., & Perry, D. E. (2007). Recovering Use-Case-Diagram-To-Source-Code Traceability Links. *Proceedings of the 6th joint meeting of the European Software Engineering*. Dubrovnik : ACM SIGSOFT.
- Hambling, B., & Goethem, P. v. (2013). *USER ACCEPTANCE TESTING A STEP-BY-STEP GUIDE*. Swindon: BCS Learning & Development .
- Hariyanto, S. (2016). SISTEM INFORMASI MANAJEMEN . 80-85.
- Hasrul , & Siregar, L. H. (2016). PENERAPAN TEKNIK KRIPTOGRAFI PADA DATABASE MENGGUNAKAN. Vol.2 Nol.2.
- Hermitha, C., Jaison, J., Yadav, J., & Fiona, J. R. (2019). Analysis on Web Frameworks. *Journal of Physics: Conference Series*, 1362.
- ISO. (2011). Systems and software engineering system Systems and software quality Requirement and Evaluation (SQuRE) System and software quality models. *ISO/IEC 25010:2011*.
- Jampani, Talasu, & Manjula. (2016). Survey of software testing techniques. *International Journal for Research in Applied Science & Engineering Technology*, 4(IV), 24–29.
- Jirava, P. (2004). *System Development Life Cycle*.
- Kenneth, K. (2010). *Analisis dan Perancangan Sistem*. Indeks.
- Laudon, K. C., & Laudon, J. P. (2014). *Management Information Systems Managing The Digital Firm* . London: Pearson Education Limited .
- Nursam, N. (2017). Manajemen Kinerja. *Journal of Islamic Education Management*, 170.
- Pandey, V., AvinashBairwa, & Bhattacharya, S. (2013). Application of the Pareto Principle in. *International Journal of Engineering and Technology (IJET)*, 2649-2654. .
- Pricillia, T., & Zulfachmi. (2021). Survey Paper: Perbandingan Metode Pengembangan . *Bangkit Indonesia*.
- Purwana, D., Rahmi, & Aditya, S. (2017). Pemanfaatan Digital Marketing Bagi Usaha Mikro, Kecil, dan Menengah (UMKM) di Kealurahan Malaka Sari, Duren Sawit. *Pemberdayaan Masyarakat Madani (JPMM)*, 4.
- Putra, D. T., & Andriani, R. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem. *Jurnal TEKNOIF*, Vol. 7 No. 1.
- Reggio, G., Leotta, M., Ricca, F., & Clerissi, D. (2013). What are the used UML diagrams? *CEUR Workshop*, vol. 1078, pp. 3–12.
- Rumah BUMN. (2022, Oktober 29). *rumah bumn*. Retrieved from rumah-bumn.id: <https://rumah-bumn.id/about>
- Supranto, J. (2000). *Statistik Teori dan Aplikasi*. Jakarta: Erlangga.
- Wahyuningrum, d. (2014). Perancangan WEB e-Commerce dengan Metode Rapid Application Development (RAD) untuk Produk Unggulan Desa. *Seminar Teknologi Informasi dan Komunikasi* .