

**Perancangan Program Aplikasi *Timesheet* Berbasis *Website* Pada  
Sinar Maju Bersama Bogor**



**SKRIPSI**

Diajukan untuk memenuhi salah satu syarat kelulusan Strata Satu (S1)

**Vincent Wijaya**

**17200711**

**Program Studi Teknologi Informasi**

**Fakultas Teknik dan Informatika**

**Universitas Bina Sarana Informatika**

**Jakarta**

**2024**

## SURAT PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini:

Nama : Vincent Wijaya  
NIM : 17200711  
Jenjang : Sarjana (S1)  
Program Studi : Teknologi Informasi  
Fakultas : Teknik dan Informatika  
Perguruan Tinggi : Universitas Bina Sarana Informatika

Dengan ini menyatakan bahwa Tugas Akhir yang telah saya buat dengan judul: **“Perancangan Program Aplikasi *Timesheet* Berbasis Website Pada Sinar Maju Bersama Bogor”**, adalah asli (orisinal) atau tidak plagiat (menjiplak) dan belum pernah diterbitkan/dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak manapun juga. Apabila di kemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa Tugas Akhir yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari **Universitas Bina Sarana Informatika** dicabut/dibatalkan.

Dibuat di : Jakarta

Pada tanggal : 01 Juli 2024

Yang menyatakan,



Vincent Wijaya

# **SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

Yang bertanda tangan di bawah ini, saya:

Nama Lengkap : Vincent Wijaya  
NIM : 17200711  
Fakultas : Teknik Informatika  
Program Studi : Teknologi Informasi  
Perguruan Tinggi : Universitas Bina Sarana Informatika Kramat 98

Dengan ini menyatakan bahwa seluruh data, informasi, interpretasi serta pernyataan yang terdapat dalam karya ilmiah Penulis dengan judul **“Perancangan Program Aplikasi Timesheet Berbasis Website Pada Sinar Maju Bersama Bogor”** ini, kecuali yang disebutkan sumbernya adalah hasil pengamatan, penelitian, pengelolaan, serta pemikiran saya.

Penulis menyetujui untuk memberikan izin kepada pihak **Universitas Bina Sarana Informatika** untuk mendokumentasikan karya ilmiah saya tersebut secara internal dan terbatas, serta tidak untuk mengunggah karya ilmiah Penulis pada repository Universitas Bina Sarana Informatika.

Penulis bersedia untuk bertanggung jawab secara pribadi, tanpa melibatkan pihak Universitas Bina Sarana Informatika, atas materi/isi karya ilmiah tersebut, termasuk bertanggung jawab atas dampak atau kerugian yang timbul dalam bentuk akibat tindakan yang berkaitan dengan data, informasi, interpretasi serta pernyataan yang terdapat pada karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya

Dibuat di : Bogor  
Pada tanggal : 01 Juli 2024  
Yang menyatakan,



Vincent Wijaya

## PERSETUJUAN DAN PENGESAHAN SKRIPSI

Skripsi ini diajukan oleh:

Nama : Vincent Wijaya  
NIM : 17200711  
Jenjang : Sarjana (S1)  
Program Studi : Teknologi Informasi  
Fakultas : Teknik dan Informatika  
Perguruan Tinggi : Universitas Bina Sarana Informatika  
Judul Skripsi : Perancangan Program Aplikasi Timesheet Berbasis Website  
Pada Sinar Maju Bersama Bogor

Telah dipertahankan pada periode 2024-1 dihadapan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh Sarjana Komputer (S.Kom) pada Program Sarjana (S1) Program Studi Teknologi Informasi di Universitas Bina Sarana Informatika.

Pembimbing I : Ari Abdilah, M.Kom.

Penguji I : Fintri Indriyani, M.Kom.

Penguji II : Rachmawati Darna Astuti, M.Kom.

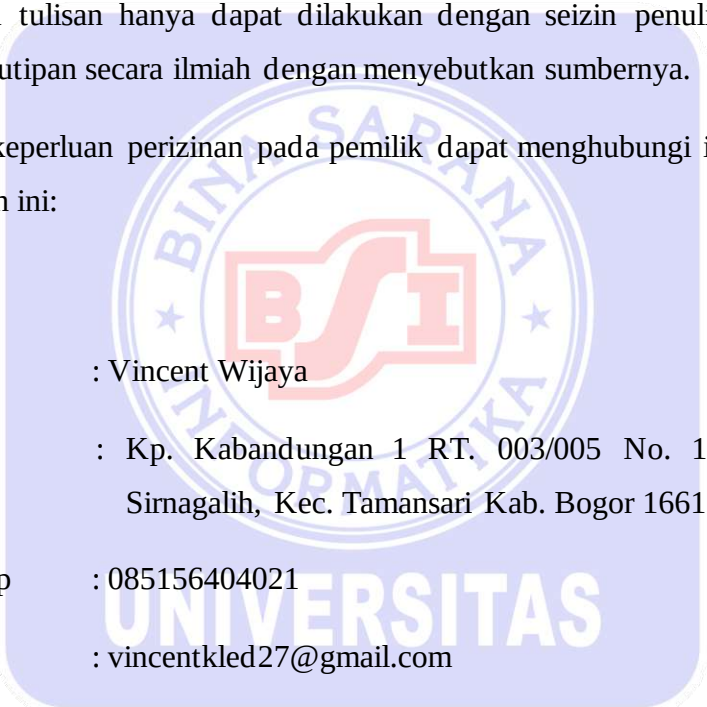


## LEMBAR PANDUAN PENGGUNAAN HAK CIPTA

Skripsi sarjana yang berjudul: “**Perancangan Program Aplikasi *Timesheet* Berbasis Website Pada Sinar Maju Bersama Bogor**” adalah hasil karya tulis asli VINCENT WIJAYA dan bukan hasil terbitan sehingga peredaran karya tulis hanya berlaku dilingkungan akademik saja, serta memiliki hak cipta. Oleh karena itu, dilarang keras untuk menggandakan baik sebagian maupun seluruhnya karya tulis ini, tanpa seizin penulis.

Referensi kepustakaan diperkenankan untuk dicatat tetapi pengutipan atau peringkasan isi tulisan hanya dapat dilakukan dengan seizin penulis dan disertai ketentuan pengutipan secara ilmiah dengan menyebutkan sumbernya.

Untuk keperluan perizinan pada pemilik dapat menghubungi informasi yang tertera di bawah ini:



Nama : Vincent Wijaya

Alamat : Kp. Kabandungan 1 RT. 003/005 No. 18A Kelurahan  
Sirnagalih, Kec. Tamansari Kab. Bogor 16610

No. Telp : 085156404021

E-mail : vincentkled27@gmail.com





## LEMBAR KONSULTASI SKRIPSI

UNIVERSITAS BINA SARANA INFORMATIKA

NIM : 17200711  
Nama Lengkap : Vincent Wijaya  
Dosen Pembimbing : Ari Abdilah, M.Kom  
Judul Skripsi : **Perancangan Program Aplikasi *Timesheet* Berbasis Website Pada Sinar Maju Bersama Bogor**

| No. | Tanggal Bimbingan | Pokok Bahasan                                      | Paraf Dosen Pembimbing I |
|-----|-------------------|----------------------------------------------------|--------------------------|
| 1.  | 08 April 2024     | Bimbingan Perdana                                  |                          |
| 2.  | 15 Mei 2024       | Bimbingan Bab 1                                    |                          |
| 3.  | 22 Mei 2024       | Revisi BAB I dan Bimbingan BAB II                  |                          |
| 4.  | 29 Mei 2024       | Revisi BAB II dan Bimbingan BAB III                |                          |
| 5.  | 4 Juni 2024       | Revisi BAB I, BAB II, BAB III dan Bimbingan BAB IV |                          |
| 6.  | 11 Juni 2024      | Revisi BAB III dan Bimbingan BAB IV                |                          |
| 7.  | 25 Juni 2024      | Revisi BAB IV dan Bimbingan Laporan Skripsi        |                          |
| 8.  | 01 Juli 2024      | Bimbingan Akhir                                    |                          |
| 9.  | 05 Agustus 2024   | Revisi Sidang Skripsi                              |                          |

Catatan untuk Dosen Pembimbing.  
Bimbingan Skripsi

Dimulai pada tanggal : 08 April 2024  
Diakhiri pada tanggal : 05 Agustus 2024  
Jumlah pertemuan bimbingan : 9

Disetujui oleh,  
Dosen pembimbing I

(Ari Abdilah, M.Kom)

## PERSEMBAHAN

*“Sebab TUHAN, Dia sendiri akan berjalan di depanmu, Dia sendiri akan menyertai engkau, Dia tidak akan membiarkan engkau dan tidak akan meninggalkan engkau;*

*Janganlah takut dan janganlah patah hati”.*

*(Ulangan 31)*

Dengan mengucapkan syukur kepada Tuhan Yang Maha Esa, Skripsi saya Vincent Wijaya dipersembahkan untuk:

1. Papa dan Mama (Wijaya Cahyadi dan Evita Debora) saya tercinta yang selalu memberikan Doa terbaik dan menjadi motivasi saya dalam penyelesaian skripsi ini supaya saya bisa lulus dan mendapatkan gelar sarjana, guna menjadi manusia yang lebih baik dimasa mendatang.
2. Om dan Tante (Johan dan Leli) saya yang telah memberikan dukungan dalam masa penyelesaian perkuliahan.
3. Sahabat dan teman kampus yang sudah memberikan Doa, dukungan, dan saran.
4. Teman-teman di kantor PT. Bumi Amarta Teknologi Mandiri yang membantu saya berfikir kritis dalam bootcamp

*Tanpa mereka,*

*aku dan karya ini tak akan pernah ada*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga penulisan skripsi ini dapat diselesaikan dengan baik. Adapun judul dari penulisan Skripsi yang diambil adalah “Perancangan Program Aplikasi *Timesheet* Berbasis *Website* Pada Sinar Maju Bersama Bogor”. Tujuan dari penulisan Skripsi ini yaitu untuk memenuhi salah satu syarat kelulusan Sarjana Universitas Bina Sarana Informatika Kampus Kota Jakarta.

Dalam penyusunan Skripsi ini, penulis menyadari bahwa tanpa adanya bimbingan dan dukungan dari semua pihak, penulisan Skripsi ini tidak akan dapat terselesaikan dengan baik dan tepat pada waktunya. Oleh karena itu, secara khusus penulis mengucapkan terimakasih yang tulus kepada:

1. Bapak Rektor Universitas Bina Sarana Informatika
2. Dekan Fakultas Teknik dan Informatika.
3. Ketua Program Studi Teknologi Informasi Universitas Bina Sarana Informatika.
4. Bapak Ari Abdilah, M.Kom selaku Dosen Pembimbingan Skripsi.
5. Dosen / staff / karyawan Universitas Bina Sarana Informatika Kota Jakarta
6. Bapak Wijaya Cahyadi selaku CEO Sinar Maju Bersama.
7. Seluruh pegawai Sinar Maju Bersama yang telah membimbing penulis selama melakukan riset.
8. Kedua orang tua yang selalu memberikan doa, dukungan, serta bantuan baik berupa moril maupun materil yang tidak ada hentinya kepada penulis.



Penulis menyadari penulisan Skripsi ini masih jauh sekali dari kata sempurna, untuk itu penulis mohon saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang. Akhir kata semoga Skripsi ini dapat berguna bagi penulis dan umumnya bagi para pembaca.

Jakarta, 01 Juli 2024

Penulis



Vincent Wijaya

## ABSTRAKSI

### **Perancangan Program Aplikasi *Timesheet* Berbasis *Website* Pada Sinar Maju**

**Bersama Bogor, Vincent Wijaya (17200711)**

Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi *timesheet* berbasis web di perusahaan Sinar Maju Bersama di Bogor, dengan fokus meningkatkan efisiensi dan efektivitas manajemen waktu kerja karyawan. Permasalahan utama yang dihadapi oleh perusahaan adalah proses input *timesheet* secara manual oleh karyawan yang sering menyebabkan kesalahan dan ketidakakuratan dalam pencatatan waktu kerja. Untuk mengatasi masalah ini, metode pengembangan sistem yang digunakan adalah metodologi *Waterfall* yang mencakup tahapan analisis, desain, implementasi, pengujian, dan pemeliharaan. Implementasi aplikasi dilakukan menggunakan teknologi web modern seperti *React*, *HTML*, *CSS*, *JavaScript*, *SpringBoot*, dan basis data *MySQL*. Aplikasi ini memudahkan karyawan untuk mencatat waktu kerja dengan lebih akurat, sementara manajer dapat memantau aktivitas karyawan dan menghasilkan laporan berdasarkan data yang tersimpan dalam sistem. Aplikasi *timesheet* ini dapat meningkatkan produktivitas, transparansi, dan efisiensi manajemen waktu kerja di perusahaan, serta memberikan solusi yang komprehensif untuk kebutuhan struktur dan efisiensi manajemen jam kerja karyawan.

**Kata Kunci:** aplikasi *timesheet*, web, Sinar Maju Bersama, Bogor.

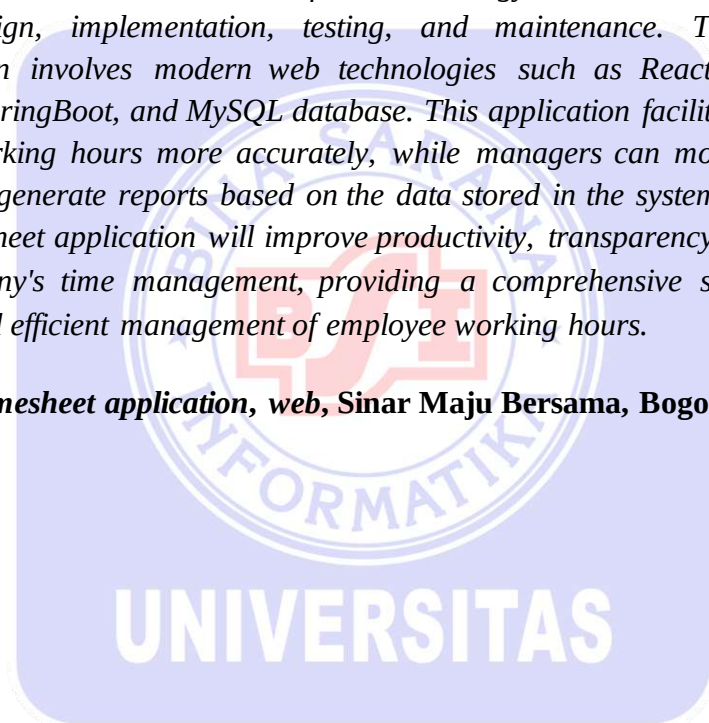
## **Abstract**

### **Design of Web-Based Timesheet Application at Sinar Maju Bersama Bogor,**

**Vincent Wijaya (17200711)**

*This study aims to design and implement a web-based timesheet application at the Sinar Maju Bersama company in Bogor, focusing on improving the efficiency and effectiveness of employee time management. The main problem faced by the company is the manual timesheet input process by employees, which often leads to errors and inaccuracies in recording working hours. To address this issue, the system development method used is the Waterfall methodology, which includes the stages of analysis, design, implementation, testing, and maintenance. The application implementation involves modern web technologies such as React, HTML, CSS, JavaScript, SpringBoot, and MySQL database. This application facilitates employees to record working hours more accurately, while managers can monitor employee activities and generate reports based on the data stored in the system. It is expected that this timesheet application will improve productivity, transparency, and efficiency in the company's time management, providing a comprehensive solution for the structured and efficient management of employee working hours.*

**Keywords:** *timesheet application, web, Sinar Maju Bersama, Bogor.*



## DAFTAR ISI

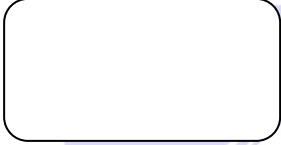
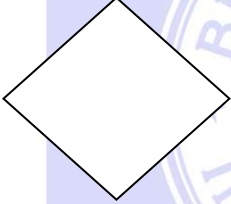
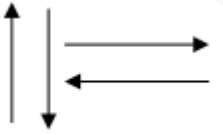
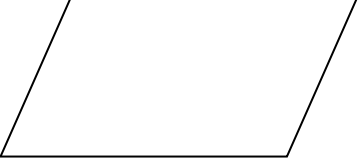
|                                                                 |       |
|-----------------------------------------------------------------|-------|
| LEMBAR JUDUL SKRIPSI.....                                       | i     |
| LEMBAR PERNYATAAN KEASLIAN SKRIPSI .....                        | ii    |
| LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....       | iii   |
| LEMBAR PERSETUJUAN DAN PENGESAHAN SKRIPSI.....                  | iv    |
| LEMBAR PEDOMAN PENGGUNAAN HAK CIPTA .....                       | v     |
| LEMBAR KONSULTASI SKRIPSI .....                                 | vi    |
| LEMBAR PERSEMBAHAN .....                                        | vii   |
| KATA PENGANTAR.....                                             | viii  |
| ABSTRAKSI.....                                                  | x     |
| DAFTAR ISI.....                                                 | xii   |
| DAFTAR SIMBOL .....                                             | xiv   |
| DAFTAR GAMBAR.....                                              | xviii |
| DAFTAR TABEL.....                                               | xix   |
| DAFTAR LAMPIRAN .....                                           | xx    |
| BAB I PENDAHULUAN.....                                          | 1     |
| 1.1. Latar Belakang Masalah.....                                | 1     |
| 1.2. Identifikasi Permasalahan .....                            | 4     |
| 1.3. Perumusan Masalah.....                                     | 5     |
| 1.4. Tujuan dan Manfaat.....                                    | 5     |
| 1.5. Metode Penelitian.....                                     | 6     |
| 1.6 Ruang Lingkup .....                                         | 10    |
| BAB II LANDASAN TEORI .....                                     | 12    |
| 2.1. Tinjauan Pustaka .....                                     | 12    |
| 2.1.1. Pengertian Perancangan.....                              | 12    |
| 2.1.2. Pengertian Program.....                                  | 12    |
| 2.1.3. Pengertian Website.....                                  | 13    |
| 2.1.4. Pengertian Timesheet .....                               | 14    |
| 2.1.5. Model Dasar Pengembangan Aplikasi Berbasis Website ..... | 15    |
| 2.1.6. Pengertian UML (Unified Modelling Language).....         | 21    |
| 2.1.7. Logical Relationship Structure (LRS).....                | 30    |
| 2.1.8. Bahasa Pemrograman.....                                  | 30    |
| 2.1.9. Pengertian Database .....                                | 33    |
| 2.2 Penelitian Terkait.....                                     | 33    |

|                                                       |           |
|-------------------------------------------------------|-----------|
| <b>BAB III ANALISIS DAN PERANCANGAN APLIKASI.....</b> | <b>35</b> |
| 3.1 Tinjauan Perusahaan.....                          | 35        |
| 3.1.1 Sejarah Perusahaan.....                         | 35        |
| 3.1.2 Struktur Organisasi.....                        | 36        |
| 3.2 Analisis Aplikasi.....                            | 38        |
| 3.2.1. Analisis Masalah.....                          | 39        |
| 3.2.2. Analisis Kebutuhan.....                        | 39        |
| 3.3 Rancangan Algoritma.....                          | 41        |
| <b>BAB IV IMPELEMENTASI DAN PENGUJIAN SISTEM.....</b> | <b>45</b> |
| 4.1 <i>Design</i> .....                               | 45        |
| 4.1.1. <i>Database</i> .....                          | 45        |
| 4.1.2. <i>Software Architecture</i> .....             | 48        |
| 4.1.3. <i>User Interface</i> .....                    | 50        |
| 4.2 <i>Code Generation</i> .....                      | 54        |
| 4.3 <i>Testing</i> .....                              | 76        |
| 4.4 <i>Support</i> .....                              | 79        |
| <b>BAB V PENUTUP.....</b>                             | <b>82</b> |
| 5.1 Kesimpulan.....                                   | 82        |
| 5.2 Saran.....                                        | 83        |
| <b>DAFTAR PUSTAKA.....</b>                            | <b>84</b> |
| <b>DAFTAR RIWAYAT HIDUP.....</b>                      | <b>86</b> |
| <b>SURAT KETERANGAN RISET PERUSAHAAN.....</b>         | <b>87</b> |
| <b>BUKTI HASIL PENGECEKAN PLAGIARISME.....</b>        | <b>88</b> |
| <b>LAMPIRAN.....</b>                                  | <b>89</b> |

## DAFTAR SIMBOL

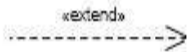
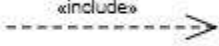
### a. Simbol UML (*Unified Modelling Language*)

#### *Activity Diagram*

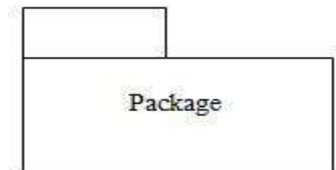
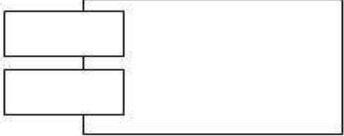
|                                                                                     |                                                                                                                                                                            |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p><b>STATUS AWAL/INITIAL</b></p> <p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.</p>                                               |
|    | <p><b>TERMINAL</b></p> <p>Digunakan untuk menggambarkan awal dan akhir dari suatu kegiatan.</p>                                                                            |
|   | <p><b>Decision</b></p> <p>Digunakan untuk menggambarkan proses pengujian suatu kondisi</p>                                                                                 |
|  | <p><b>FLOWLINE</b></p> <p>Digunakan untuk menggambarkan hubungan proses dari satu proses ke proses lainnya</p>                                                             |
|  | <p><b>Input/Output</b></p> <p>Digunakan untuk menggambarkan proses memasukan data yang berupa pembacaan data dan sekaligus proses keluaran yang berupa pencetakan data</p> |



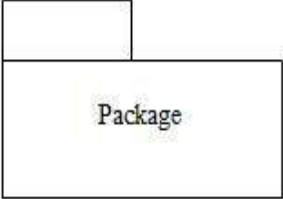
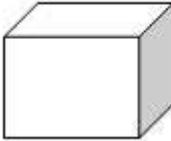
## b. Use Case Diagram

|                                                                                     |                                                                                                                                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p><b>ACTOR</b></p> <p>Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>.</p> |
|    | <p><b>USE CASE</b></p> <p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>                                                                                 |
|    | <p><b>ASOSIASI/ASSOCIATION</b></p> <p>Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>.</p>                                                                                                           |
|  | <p><b>EKSTENSI/EXTEND</b></p> <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang di tambahkan.</p>                                    |
|  | <p><b>GENERALISASI/GENERALIZATION</b></p> <p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>                                                                                                    |
|  | <p><b>MENGGUNAKAN/INCLUDE</b></p> <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsional atau sebagai syarat dijalankan <i>use case</i> ini.</p>                                                   |

### c. Component Diagram

|                                                                                     |                                                                                                                                                                          |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p><i>PACKAGE</i></p> <p><i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen.</p>                                                                      |
|    | <p>KOMPONEN/<i>COMPONENT</i></p> <p>Komponen merupakan komponen sistem.</p>                                                                                              |
|    | <p>KEBERGANTUNGAN/<i>DEPENDENCY</i></p> <p>Ketergantungan atau <i>dependency</i> atau kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.</p> |
|  | <p><i>LINK</i></p> <p>Menggambarkan relasi antar komponen.</p>                                                                                                           |

#### d. Deployment Diagram

|                                                                                     |                                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p><b>PACKAGE</b></p> <p><i>Package</i> merupakan sebuah bungkusan dari satu atau lebih komponen.</p>                                                                                                                                                                                                     |
|    | <p><b>NODE</b></p> <p><i>Node</i> biasa mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen.</p> |
|    | <p><b>KEBERGANTUNGAN/DEPENDENCY</b></p> <p>Ketergantungan atau <i>dependency</i> atau kebergantungan antar <i>node</i>, arah panah mengarah pada <i>node</i> yang dipakai.</p>                                                                                                                            |
|  | <p><b>LINK</b></p> <p>Menggambarkan relasi antar node.</p>                                                                                                                                                                                                                                                |

## DAFTAR GAMBAR

|                                                              |    |
|--------------------------------------------------------------|----|
| Gambar 1. 1 Model <i>Waterfall</i> .....                     | 8  |
| Gambar 1. 2 Model <i>Waterfall</i> .....                     | 15 |
| Gambar 1. 3 Fase model <i>waterfall</i> .....                | 19 |
| Gambar 2. 1 Diagram <i>UML</i> .....                         | 22 |
| Gambar 3. 1 Struktur Organisasi.....                         | 37 |
| Gambar 3. 2 Use Case.....                                    | 41 |
| Gambar 3. 3 Flowchart Administrator .....                    | 42 |
| Gambar 3. 4 Flowchart User .....                             | 43 |
| Gambar 4. 1 <i>Entity Relationship Diagram</i> .....         | 46 |
| Gambar 4. 2 <i>Logical Record Structure</i> .....            | 47 |
| Gambar 4. 3 <i>Component Diagram</i> .....                   | 48 |
| Gambar 4. 4 <i>Deployment Diagram</i> .....                  | 48 |
| Gambar 4. 5 <i>Activity Diagram</i> .....                    | 49 |
| Gambar 4. 7 Halaman <i>Login</i> .....                       | 50 |
| Gambar 4. 8 <i>User Menu</i> .....                           | 51 |
| Gambar 4. 9 <i>Admin Menu</i> .....                          | 52 |
| Gambar 4. 10 <i>Feature Approval by Manager</i> .....        | 53 |
| Gambar 4. 11 <i>Feature Change Date by Admin</i> .....       | 53 |
| Gambar 4. 12 <i>View employee timesheet by Manager</i> ..... | 53 |

## DAFTAR TABEL

|                                         |    |
|-----------------------------------------|----|
| Tabel IV. 7 Form Login .....            | 77 |
| Tabel IV. 8 Form <i>Timesheet</i> ..... | 78 |
| Tabel IV. 9 Form Security.....          | 79 |
| Tabel IV. 10 Spesifikasi.....           | 81 |



## DAFTAR LAMPIRAN

|                                        |    |
|----------------------------------------|----|
| Lampiran 1 Data Absensi Karyawan ..... | 89 |
| Lampiran 2 Data Grafik Karyawan.....   | 89 |



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Di era digital saat ini, penggunaan teknologi informasi telah menjadi kebutuhan mendesak bagi berbagai perusahaan untuk meningkatkan efisiensi dan efektivitas operasional mereka. Teknologi informasi tidak hanya membantu dalam mempercepat proses bisnis, tetapi juga memberikan akses yang lebih luas terhadap data dan analisis yang penting untuk pengambilan keputusan yang lebih baik. Selain itu, penerapan teknologi ini dapat meningkatkan kemampuan perusahaan dalam beradaptasi dengan perubahan pasar yang cepat dan meminimalkan risiko kesalahan manusia. Sebagai contoh, banyak perusahaan yang mengintegrasikan sistem manajemen informasi untuk mengoptimalkan alur kerja dan memastikan komunikasi yang lebih efektif antara departemen. Dalam konteks hukum, penggunaan teknologi informasi juga diatur oleh berbagai peraturan perundang-undangan yang memastikan keamanan dan privasi data perusahaan serta pelanggan. Salah satu teknologi yang semakin penting adalah aplikasi *timesheet*, yang digunakan untuk memonitor dan mengelola waktu kerja karyawan. Aplikasi ini memungkinkan perusahaan untuk memantau produktivitas, mengoptimalkan pengaturan jadwal, serta memastikan bahwa setiap jam kerja tercatat dengan baik dan akurat. Dalam situasi saat ini, teknologi telah menjadi alat yang penting dan tidak dapat dipisahkan dari aktivitas kita sehari-hari. Perkembangan teknologi yang cepat dan progresif berdampak besar pada tindakan dan kehidupan kita. Dalam hal ini, kemajuan teknologi memberikan efek positif, membuat kegiatan yang dilakukan lebih praktis dan efisien (Febriani et al., 2023). Perusahaan Sinar Maju Bersama adalah salah satu perusahaan yang mengalami



pertumbuhan pesat dan memiliki jumlah karyawan yang cukup banyak. Dalam mengelola operasional sehari-hari, perusahaan ini menghadapi tantangan dalam mencatat dan memantau waktu kerja karyawan secara manual. Sistem pencatatan manual ini sering kali memakan waktu, rentan terhadap kesalahan, dan kurang transparan, sehingga berpotensi menimbulkan ketidakakuratan dalam analisis produktivitas.

Dalam rangka mengatasi permasalahan tersebut, dibutuhkan sebuah sistem yang efisien dan terstruktur untuk mengelola waktu kerja karyawan. Aplikasi *timesheet* berbasis *website* menjadi solusi yang tepat untuk perusahaan Sinar Maju Bersama. Dengan adanya aplikasi ini, proses pelaporan waktu kerja diharapkan dapat lebih terstruktur dan transparan, sehingga perusahaan dapat melakukan monitoring dan evaluasi kinerja karyawan dengan lebih efektif.

Aplikasi *timesheet* berbasis *website* ini menawarkan berbagai keunggulan, antara lain kemudahan akses dari mana saja dan kapan saja, integrasi data yang lebih baik, serta kemampuan untuk menghasilkan laporan yang akurat. Penerapan aplikasi ini juga diharapkan dapat meningkatkan kepuasan karyawan, karena mereka dapat dengan mudah mengakses dan memverifikasi data waktu kerja mereka sendiri. Aplikasi ini menyediakan antarmuka yang *user-friendly*, sehingga karyawan dapat dengan cepat memeriksa jadwal kerja, mencatat jam masuk dan keluar, serta mengajukan permintaan cuti atau lembur. Selain itu, transparansi yang dihasilkan oleh aplikasi ini memungkinkan karyawan merasa lebih dihargai dan diberdayakan dalam mengelola waktu kerja mereka. Dalam jangka panjang, kepuasan karyawan yang meningkat dapat berkontribusi pada produktivitas yang lebih tinggi dan loyalitas yang lebih besar terhadap perusahaan. Implementasi teknologi ini juga selaras dengan

regulasi ketenagakerjaan yang mengharuskan perusahaan untuk menjaga catatan waktu kerja yang akurat dan dapat diakses oleh karyawan.

Oleh karena itu, penelitian ini bertujuan merancang dan mengembangkan sebuah program aplikasi *timesheet* berbasis *website* yang dapat memenuhi kebutuhan perusahaan Sinar Maju Bersama. Aplikasi ini diharapkan dapat memfasilitasi pencatatan waktu kerja secara lebih efisien dan akurat, serta memungkinkan manajemen untuk mengakses data real-time yang dibutuhkan untuk pengambilan keputusan. Selain itu, dengan adanya aplikasi ini, karyawan dapat dengan mudah mengelola dan memverifikasi jam kerja mereka sendiri, sehingga meningkatkan transparansi dan kepuasan kerja. Implementasi program ini juga akan mempertimbangkan integrasi dengan sistem lain yang sudah ada di perusahaan untuk memastikan keselarasan dan kelancaran operasional. Perancangan aplikasi *timesheet* ini tidak hanya bertujuan untuk meningkatkan efisiensi administrasi tetapi juga untuk mematuhi regulasi ketenagakerjaan yang berlaku, sehingga perusahaan dapat beroperasi dengan lebih efektif dan sesuai dengan ketentuan hukum. Dengan adanya aplikasi ini, diharapkan dapat tercipta sistem pengelolaan waktu kerja yang lebih efisien, akurat, dan transparan, serta mendukung perusahaan dalam mencapai tujuan operasional dan strategisnya.

Dalam hal ini penulis mencoba untuk membuat dengan judul :”Perancangan Program Aplikasi *Timesheet* Berbasis *Website* pada Sinar Maju Bersama Bogor”.

## 1.2. Identifikasi Permasalahan

Dalam proses perancangan dan pengembangan aplikasi *timesheet* berbasis *website* untuk perusahaan Sinar Maju Bersama Bogor, beberapa permasalahan yang perlu diidentifikasi dan diatasi adalah sebagai berikut:

### 1. Kesulitan dalam Pencatatan Waktu Kerja Secara Manual

Pencatatan waktu kerja karyawan secara manual sering kali memakan waktu dan rentan terhadap kesalahan. Hal ini dapat mengakibatkan ketidakakuratan data, yang pada akhirnya mempengaruhi perhitungan gaji dan evaluasi kinerja. Oleh karena itu, digitalisasi pencatatan waktu kerja menjadi solusi yang efektif untuk mengatasi masalah ini. Dengan menggunakan aplikasi *timesheet* berbasis *website*, perusahaan dapat mengurangi risiko kesalahan manusia dan memastikan data yang lebih akurat. Aplikasi ini juga memungkinkan otomatisasi dalam perhitungan gaji dan penilaian kinerja, sehingga manajemen dapat fokus pada tugas-tugas strategis lainnya. Selain itu, karyawan juga mendapatkan keuntungan dengan kemudahan akses dan transparansi data waktu kerja mereka, yang dapat meningkatkan kepercayaan dan kepuasan kerja. Implementasi teknologi ini sejalan dengan upaya perusahaan untuk meningkatkan efisiensi operasional dan kepatuhan terhadap peraturan ketenagakerjaan yang berlaku.

### 2. Proses Pelaporan yang Tidak Efisien

Pengumpulan dan penyusunan laporan waktu kerja karyawan secara manual memerlukan waktu dan upaya yang cukup besar. Hal ini dapat menghambat manajemen dalam melakukan analisis produktivitas dan membuat keputusan strategis tepat waktu.

### 1.3. Perumusan Masalah

Dari latar belakang tersebut dapat diidentifikasi menjadi beberapa masalah diantaranya:

1. Bagaimana merancang dan mengembangkan aplikasi *timesheet* berbasis *website* yang efisien untuk mencatat dan memantau waktu kerja karyawan di perusahaan Sinar Maju Bersama?
2. Bagaimana aplikasi *timesheet* berbasis *website* dapat meningkatkan akurasi dan efisiensi dalam proses pencatatan dan pelaporan waktu kerja karyawan?

### 1.4. Tujuan dan Manfaat

#### A. Tujuan

1. Sebagai syarat kelulusan penulis pada perkuliahan program Strata Satu.
2. Membantu pihak perusahaan dalam pengolahan data absensi yang efisien, singkat, akurat, dan minim kesalahan.
3. Menjadikan data absensi transparansi yang dapat diakses real-time kapanpun dan dimanapun oleh karyawan dapat terpantau.

#### B. Manfaat

1. Sebagai bahan evaluasi bagi pihak perusahaan, sehingga menghasilkan informasi yang cepat, akurat.
2. Memudahkan pihak perusahaan dalam mengelola data absensi sehingga karyawan dapat berfokus pada kegiatan.
3. Data absensi akan aman karena akan tersimpan secara online dan dapat dilihat secara real-time.

## 1.5. Metode Penelitian

### 1.5.1 Teknik Pengumpulan Data

#### A. Observasi

Observasi adalah suatu metode untuk melakukan pengamatan, di mana hasilnya secara langsung dicatat oleh penulis. Dari kegiatan pengamatan ini, dapat diketahui proses dan aktivitas yang terjadi. Metode ini memungkinkan peneliti untuk mendapatkan data secara real-time dan memastikan akurasi informasi yang diperoleh. Selain itu, observasi juga memberikan wawasan mendalam tentang konteks dan lingkungan di mana aktivitas tersebut berlangsung. Dengan mencatat langsung, peneliti dapat menangkap detail-detail penting yang mungkin terlewatkan jika menggunakan metode lain. Observasi juga bisa menjadi dasar yang kuat untuk analisis lebih lanjut, membantu peneliti dalam memahami dinamika yang ada serta dalam membuat kesimpulan yang lebih akurat. Penggunaan metode observasi ini sangat relevan dalam berbagai bidang penelitian, termasuk sains sosial, pendidikan, dan bisnis, karena memberikan data yang autentik dan dapat diandalkan. (Laia et al., 2023)

#### B. Wawancara

Wawancara adalah metode melakukan tanya jawab dengan orang yang memiliki kredibilitas dalam memberikan jawaban terkait hal-hal yang berhubungan dengan objek laporan, seperti salah satu karyawan Sinar Maju Bersama. Metode ini memungkinkan pengumpulan data yang mendalam dan spesifik dari sumber yang berpengalaman dan memiliki pengetahuan langsung mengenai topik yang dibahas. Melalui wawancara, peneliti dapat menggali informasi yang mungkin tidak tersedia melalui metode pengumpulan data

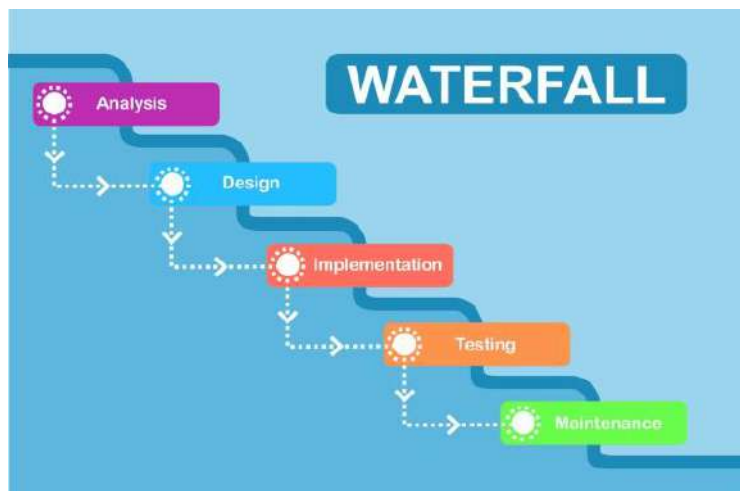
lainnya. Selain itu, wawancara memberikan kesempatan untuk klarifikasi dan eksplorasi lebih lanjut tentang jawaban yang diberikan, sehingga menghasilkan data yang lebih kaya dan kontekstual. Penggunaan wawancara ini sangat penting dalam memahami perspektif individu yang terlibat langsung dengan objek laporan dan memastikan bahwa informasi yang dikumpulkan adalah relevan dan akurat.(Ritonga, 2023)

### **C. Studi Pustaka**

Teknik ini digunakan untuk mengumpulkan data dengan bahan referensi dari jurnal dan buku-buku yang berhubungan langsung dengan masalah yang dibahas. Dengan menggunakan referensi yang kredibel dan relevan, teknik ini memastikan bahwa data yang diperoleh adalah valid dan dapat diandalkan. Referensi dari jurnal memberikan akses ke penelitian terbaru dan perkembangan terkini dalam bidang yang sedang diteliti, sedangkan buku-buku menyediakan landasan teori yang kuat dan mendalam. Kombinasi dari kedua sumber ini membantu peneliti untuk mendapatkan pemahaman yang komprehensif tentang topik yang dibahas, mendukung analisis yang lebih kritis dan informatif. Teknik ini juga penting dalam memastikan bahwa penelitian mengikuti standar akademis yang tinggi dan mematuhi etika penelitian yang berlaku.(Aisyah et al., 2023)

### 1.5.2. Model Pengembangan Software

#### Metode Pengembangan Perangkat Lunak *Waterfall*



Sumber : <https://osc.medcom.id/community/analisis-model-waterfall-pengertian-tahapan-kelebihan-dan-kekurangan-4352>

**Gambar 1. 1 Model Waterfall**

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan model water fall yang terbagi menjadi beberapa tahapan, yaitu:

**A. Analisa Kebutuhan Software**

Pada tahap Analisa, pengumpulan informasi tentang kebutuhan sistem dari pengguna dan pemangku kepentingan terkait. Proses analisis ini bertujuan untuk memahami secara mendalam yang dibutuhkan oleh system yang akan dikembangkan, baik dari segi fungsional maupun non-fungsional.

**B. Desain**

Desain sistem dilakukan untuk menentukan arsitektur secara keseluruhan, struktur data, antarmuka pengguna, serta modul-modul yang diperlukan dalam pengembangan perangkat lunak.



### **C. Code Generation**

Pada tahap Code Generation, desain yang telah dibuat diimplementasikan menjadi kode program atau sumber daya yang diperlukan dalam pengembangan perangkat lunak. Proses ini melibatkan penerjemahan desain sistem ke dalam bahasa pemrograman yang sesuai, memastikan bahwa setiap komponen berfungsi sebagaimana direncanakan. Selain itu, tahap ini juga mencakup pembuatan dokumentasi kode yang lengkap untuk memudahkan pemeliharaan dan pengembangan di masa depan. Pemanfaatan alat bantu otomatisasi dan lingkungan pengembangan terintegrasi (IDE) dapat meningkatkan efisiensi dan akurasi selama tahap ini. Implementasi kode yang baik harus mematuhi standar pengkodean dan praktik terbaik yang berlaku, guna memastikan perangkat lunak yang dihasilkan berkualitas tinggi, mudah dipelihara, dan bebas dari bug. Tahap Code Generation merupakan langkah krusial dalam siklus hidup pengembangan perangkat lunak, karena kualitas kode yang dihasilkan sangat mempengaruhi performa dan keberlanjutan perangkat lunak tersebut.

### **D. Testing**

Setelah kode program dihasilkan, tahap pengujian dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditentukan dan berjalan dengan baik dalam lingkungan yang diinginkan. Proses ini mencakup berbagai jenis pengujian, seperti pengujian unit, integrasi, sistem, dan penerimaan, untuk mengidentifikasi dan memperbaiki bug serta memastikan bahwa semua

fitur bekerja sebagaimana mestinya. Selain itu, pengujian juga bertujuan untuk menilai performa perangkat lunak dalam kondisi beban yang berbeda dan memastikan kompatibilitas dengan berbagai sistem operasi dan perangkat keras. Dokumentasi hasil pengujian sangat penting untuk referensi dan pemeliharaan di masa depan, serta untuk memenuhi persyaratan kualitas yang telah ditetapkan. Tahap pengujian ini tidak hanya memastikan keandalan dan stabilitas perangkat lunak, tetapi juga membantu meningkatkan kepuasan pengguna dengan menyediakan produk akhir yang bebas dari kesalahan dan siap untuk digunakan.

#### **E. Support**

Tahap ini melibatkan dukungan teknis terhadap perangkat lunak setelah diluncurkan, termasuk perbaikan bug, pembaruan, dan pelayanan sistem agar dapat memenuhi kebutuhan pengguna secara berkelanjutan.

### **1.6 Ruang Lingkup**

Dalam penelitian ini, dilakukan penerapan Program *Timesheet* berbasis web di perusahaan Sinar Maju Bersama dengan menggunakan pola *arsitektur MVC (Model-View-Controller)*. Penerapan ini mencakup berbagai formulir seperti :

#### **A. Administrator**

Formulir login, formulir menu, formulir input *Timesheet*, formulir ubah tanggal, dan formulir tampilan *Timesheet* pribadi.

#### **B. User**

Formulir login, formulir menu, formulir input *Timesheet*, formulir ubah tanggal, dan formulir tampilan *Timesheet* pribadi.

### C. Manager

Formulir login, formulir menu, formulir input *Timesheet*, formulir persetujuan *timesheet* dan formulir tampilan *Timesheet* pribadi.

Setiap formulir tersebut dirancang untuk meningkatkan efisiensi dan akurasi dalam pengelolaan waktu kerja karyawan. Melalui penggunaan MVC, program ini diharapkan dapat memberikan struktur yang lebih baik dan memudahkan proses pengembangan serta pemeliharaan aplikasi.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Pada tahapan perancangan program aplikasi *timesheet* berbasis *web*, diperlukan teori yang mendukung keberhasilan dalam pembuatan aplikasi berbasis *web* tersebut. Dalam penulisan, penulis mengambil berbagai-macam materi mulai dari buku, internet dan jurnal agar memperkaya isi dari skripsi ini.

##### **2.1.1. Pengertian Perancangan**

Perancangan “adalah proses untuk menentukan sesuatu yang akan di buat dengan menggunakan cara yang beragam dan didalamnya berisi penjelasan mengenai rincian komponen dan keterbatasan saat proses pembuatannya”. Menurut (Triono et al., 2018) perancangan didefinisikan sebagai proses pengembangan uraian berdasarkan pertimbangan dari hasil analisis sistem. Dari beberapa penjelasan diatas, dapat disimpulkan bahwa perancangan melibatkan proses pengerjaan atau pengembangan dari sesuatu yang sudah ada berdasarkan analisis yang telah dilakukan. (Syabania & Rosmawarni, 2021)

##### **2.1.2. Pengertian Program**

Program komputer merupakan inti dari sistem informasi yang menghubungkan antara input dan output serta mengelola data dan informasi. Program komputer dapat dikembangkan dengan menggunakan berbagai bahasa pemrograman seperti *Python*, *Java*, *C++*, dan lain-lain. Setiap bahasa pemrograman memiliki kelebihan dan kekurangannya sendiri, serta aplikasi yang berbeda tergantung pada kebutuhan proyek. Misalnya, *Python* dikenal karena sintaksisnya yang sederhana dan penggunaannya dalam pengembangan *web* serta analisis data, sedangkan *Java* sering digunakan untuk

aplikasi perusahaan dan pengembangan *Android* karena portabilitasnya. *C++* dikenal karena kemampuannya dalam pengembangan sistem dan aplikasi yang membutuhkan kinerja tinggi. Pemilihan bahasa pemrograman yang tepat sangat penting untuk efisiensi dan keberhasilan proyek, karena dapat mempengaruhi waktu pengembangan, performa aplikasi, dan kemudahan pemeliharaan di masa mendatang. Selain itu, penggunaan bahasa pemrograman yang sesuai juga memastikan bahwa program komputer dapat berintegrasi dengan teknologi dan sistem yang sudah ada, serta memenuhi persyaratan fungsional dan non-fungsional yang ditetapkan. (Khairani Arintonang et al., 2022)

### 2.1.3. Pengertian Website

*Website* adalah koleksi halaman yang berisi informasi data digital seperti teks, gambar, animasi, suara, dan video, atau gabungan dari semua itu, yang dapat diakses melalui koneksi internet dan dilihat oleh pengguna di seluruh dunia. Halaman-halaman ini menggunakan bahasa standar seperti *HTML* (Hypertext Markup Language). Bahasa ini digunakan untuk membuat struktur dasar dan mengatur tampilan konten *website*, kemudian diterjemahkan oleh peramban web agar informasi tersebut dapat ditampilkan secara visual kepada pengguna.

Secara umum, terdapat tiga jenis *website*. Pertama, *website* statis, yang isinya tetap dan tidak berubah secara otomatis berdasarkan interaksi pengguna. *Website* ini cocok digunakan untuk halaman informasi umum seperti profil perusahaan atau halaman portofolio. Kedua, *website* dinamis, yang mampu mengubah konten secara otomatis berdasarkan data yang diambil dari database atau input pengguna. Contoh penggunaannya adalah situs berita yang memperbarui berita terbaru secara langsung. Ketiga, *website* interaktif, yang memungkinkan pengguna berinteraksi langsung

dengan konten atau aplikasi *website*, seperti media sosial atau platform *e-commerce* yang memungkinkan pengguna untuk berbelanja dan berinteraksi dengan produk secara langsung. Pemilihan jenis *website* yang tepat tergantung pada tujuan dan fungsi yang diinginkan, serta pengalaman interaktif yang ingin ditawarkan kepada pengguna. (Permata Sari, 2020)

#### **2.1.4. Pengertian *Timesheet***

*Timesheet* adalah aplikasi yang digunakan untuk mencatat jam kerja setiap karyawan dengan tujuan untuk melacak aktivitas yang dilakukan selama jam kerja serta durasi waktu yang digunakan untuk setiap pekerjaan. Dengan *timesheet*, perusahaan dapat memantau produktivitas dan efisiensi karyawan, serta mengevaluasi penggunaan waktu mereka untuk berbagai tugas atau proyek. Data yang tercatat dalam *timesheet* membantu manajemen dalam membuat keputusan strategis terkait alokasi sumber daya, perencanaan proyek, dan penilaian kinerja karyawan.

Selain itu, *timesheet* juga berperan penting dalam memastikan kepatuhan perusahaan terhadap peraturan ketenagakerjaan terkait jam kerja dan kompensasi. Dengan memiliki catatan yang akurat dan terperinci tentang jam kerja karyawan, perusahaan dapat memastikan bahwa gaji dan tunjangan yang diberikan sesuai dengan waktu kerja yang sebenarnya. Penggunaan *timesheet* juga memberikan transparansi bagi karyawan tentang bagaimana waktu mereka digunakan dan memberi kesempatan untuk mengevaluasi produktivitas pribadi mereka. Dalam konteks modern, aplikasi *timesheet* sering kali terintegrasi dengan sistem manajemen sumber daya manusia (*HRM*) atau perangkat lunak *ERP* (*Enterprise Resource Planning*) untuk memudahkan pengelolaan dan analisis data yang lebih lanjut. (Ernawan & Sedyono, 2022)

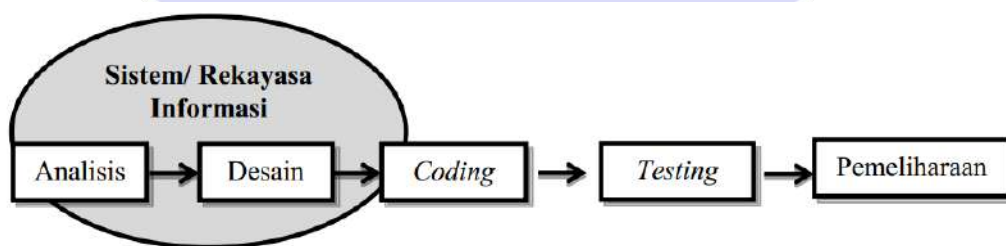
### 2.1.5. Model Dasar Pengembangan Aplikasi Berbasis Website

Pengembangan sistem aplikasi adalah proses mengubah, menghaluskan dan mengtransformasi produk yang ada. Dalam proses ini adalah mungkin untuk menggantikan subprocess dengan yang lain, subprocess baru memiliki antarmuka yang sama dengan subprocess lama.

#### 1. Model *Waterfall*

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan model water fall. Model Waterfall merupakan salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model ini menggunakan pendekatan sistematis dan berurutan. Tahapan dalam model ini dimulai dari tahap perencanaan hingga tahap pengelolaan (maintenance) dan dilakukan secara bertahap. (Abdul Wahid Sekolah Tinggi Manajemen Informatika dan Komputer Sumedang, 2020)

#### a. fase dalam model air terjun (*waterfall*) menurut Rosa A.S dan Shalahudin



Sumber: Rosa dan Shalahuddin (2019:29)

**Gambar 1. 2 Model Waterfall**

#### 1) Analisis kebutuhan perangkat lunak



Proses pengumpulan kebutuhan dilakukan secara intensif untuk merinci persyaratan sistem dengan jelas sehingga mudah dipahami. Tujuan utama dari pengumpulan kebutuhan ini adalah untuk memastikan bahwa perangkat lunak yang dikembangkan akan sesuai dengan kebutuhan dan ekspektasi pengguna akhir. Dokumentasi yang baik dari spesifikasi kebutuhan perangkat lunak sangat penting pada tahap ini, karena akan menjadi pedoman dasar dalam seluruh siklus pengembangan. Hal ini membantu tim pengembang untuk mengidentifikasi dan memahami dengan tepat fitur-fitur yang harus diimplementasikan serta fungsi-fungsi yang harus disediakan oleh perangkat lunak, sehingga mengurangi risiko terjadinya kesalahpahaman atau ketidaksesuaian dengan harapan pengguna.

## 2) Desain

Desain perangkat lunak merupakan tahapan krusial yang meliputi berbagai langkah, seperti perancangan struktur data, arsitektur perangkat lunak, pengembangan antarmuka pengguna (*UI*), dan prosedur pengkodean. Proses ini menghubungkan analisis awal kebutuhan perangkat lunak dengan tahap implementasi selanjutnya, memastikan bahwa semua elemen desain terintegrasi secara efisien untuk mencapai tujuan fungsional dan non-fungsional yang ditetapkan. Dokumentasi yang komprehensif dari hasil desain perangkat lunak pada tahap ini sangat penting, karena memberikan panduan yang jelas bagi tim pengembang untuk memahami detail teknis dan fungsional dari sistem yang akan dibangun. Dokumentasi yang baik juga memfasilitasi kolaborasi antar tim, mempercepat proses pengujian dan pemeliharaan, serta

membantu dalam mengidentifikasi perubahan yang diperlukan selama siklus hidup perangkat lunak.

### 3) *Coding*

Pada tahap *coding*, perangkat lunak diimplementasikan dengan memecahnya menjadi modul-modul kecil yang akan digabungkan pada tahap selanjutnya. Proses ini melibatkan penulisan kode berdasarkan desain perangkat lunak dan spesifikasi kebutuhan yang telah disiapkan sebelumnya. Setiap modul kode dikembangkan dengan cermat untuk memastikan bahwa aplikasi dapat beroperasi sesuai dengan kebutuhan dan tujuan yang telah ditetapkan.

Tahap *coding* juga dikenal sebagai tahap debug, di mana setiap bagian kode diperiksa dan diuji secara terpisah untuk menemukan dan memperbaiki kesalahan atau bug yang mungkin muncul. Selama proses ini, pengembang berfokus pada implementasi logika bisnis dan fungsi-fungsi yang diperlukan, serta memastikan bahwa kode dapat berintegrasi dengan baik saat nanti diuji secara keseluruhan. Selain itu, penggunaan teknik pengkodean yang baik juga memperhatikan aspek keamanan perangkat lunak dan performa, sehingga hasil akhir yang dihasilkan tidak hanya berfungsi dengan baik tetapi juga dapat diandalkan dalam lingkungan produksi.

### 4) Pengujian

Pada tahap ini, dilakukan penggabungan modul-modul yang telah dipersiapkan sebelumnya. Proses ini melibatkan integrasi semua bagian kode yang telah

dikembangkan oleh berbagai programmer, dengan tujuan untuk menyatukan fungsi-fungsi yang terpisah menjadi satu kesatuan sistem yang utuh. Setelah penggabungan, dilanjutkan dengan serangkaian pengujian untuk memverifikasi bahwa perangkat lunak beroperasi sesuai dengan desain dan fungsi yang telah ditetapkan sebelumnya.

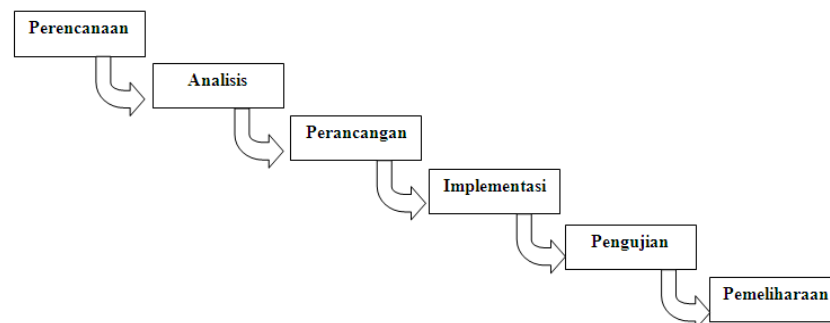
Integrasi modul merupakan langkah krusial dalam siklus pengembangan perangkat lunak, di mana keseluruhan sistem diperiksa untuk memastikan bahwa semua komponen bekerja bersama secara harmonis. Selama proses ini, tim pengujian mengidentifikasi dan mengatasi potensi konflik atau kesalahan yang mungkin terjadi antara modul-modul yang berbeda. Pengujian ini tidak hanya bertujuan untuk memverifikasi fungsionalitas perangkat lunak secara keseluruhan, tetapi juga untuk memastikan keamanan, keandalan, dan kinerja sistem dalam berbagai kondisi penggunaan yang mungkin terjadi di lapangan. Dengan demikian, tahap integrasi dan pengujian yang teliti ini memainkan peran penting dalam memastikan bahwa hasil akhir yang dihasilkan mencapai tingkat kualitas yang diharapkan dan dapat memenuhi ekspektasi pengguna serta standar industri yang berlaku.

##### 5) Pemeliharaan (*maintenance*)

Pada tahap ini, klien atau pengguna menguji *software* yang telah dibuat untuk memastikan kesesuaian dengan spesifikasi yang telah ditetapkan dan kebutuhan yang diinginkan. Proses pengujian ini mencakup identifikasi potensi kesalahan atau error dalam sistem yang mungkin memerlukan perbaikan ulang. Pengujian ini menjadi kunci untuk memastikan bahwa *software* tidak hanya

memenuhi ekspektasi teknis, tetapi juga memenuhi kebutuhan fungsional dan pengalaman pengguna yang diharapkan.

### b. Fase dalam model *waterfall*



Sumber: Rosa dan Shalahuddin (2019:29)

**Gambar 1. 3 Fase model *waterfall***

Kerangka Kerja Pengembangan Model *Waterfall*

- 1) Tahap awal dalam pengembangan perangkat lunak, yang disebut tahap perencanaan atau planning, meliputi beberapa aspek penting seperti studi kebutuhan pengguna atau *users specification*, studi kelayakan baik dari segi teknis maupun teknologis, dan penentuan jadwal pengembangan proyek perangkat lunak. Studi kebutuhan pengguna dilakukan untuk memahami secara mendalam apa yang diinginkan dan diharapkan oleh pengguna akhir dari perangkat lunak yang akan dikembangkan. Hal ini sering kali direpresentasikan dalam bentuk *use case diagram*, yang memvisualisasikan interaksi antara pengguna dan sistem serta skenario penggunaan yang berbeda.

- 2) Selain itu, tahap ini juga mencakup studi kelayakan atau *feasibility study*, yang mengevaluasi aspek-aspek teknis dan teknologis dari proyek perangkat lunak. Evaluasi ini bertujuan untuk menentukan apakah teknologi yang akan digunakan sesuai dengan kebutuhan proyek, apakah proyek dapat dilaksanakan dengan sumber daya yang tersedia, serta untuk memperkirakan risiko-risiko yang mungkin muncul selama pengembangan. Penjadwalan pengembangan juga merupakan bagian penting dari tahap perencanaan ini, di mana tim proyek membuat rencana waktu yang jelas dan realistis untuk setiap tahapan pengembangan perangkat lunak, termasuk identifikasi milestone dan target pencapaian yang harus dipenuhi. Tahap perencanaan yang komprehensif ini penting untuk mengarahkan proyek perangkat lunak ke arah yang tepat sejak awal dan meminimalkan risiko serta biaya yang terkait dengan perubahan-perubahan yang tidak terduga di kemudian hari.
- 3) "Tahap analisis atau *analysis*, yaitu tahap dimana kita berusaha mengenali semua permasalahan yang muncul pada pengguna dengan merealisasikan *use case diagram* lebih dalam, mengenali komponen-komponen sistem/perangkat lunak, objek, hubungan antara objek, dan sebagainya" (Rosa dan Shalahuddin, 2019:4).
- 4) Tahap perancangan, yaitu mencari solusi atas permasalahan yang ditemukan pada tahap analisis.
- 5) Tahap implementasi, yakni menerapkan hasil perancangan sistem ke dalam *browser*. Pada tahap ini, sistem mulai menangani pemilihan perangkat keras (*hardware*) dalam proses penyusunan perangkat lunak aplikasi (*coding*).

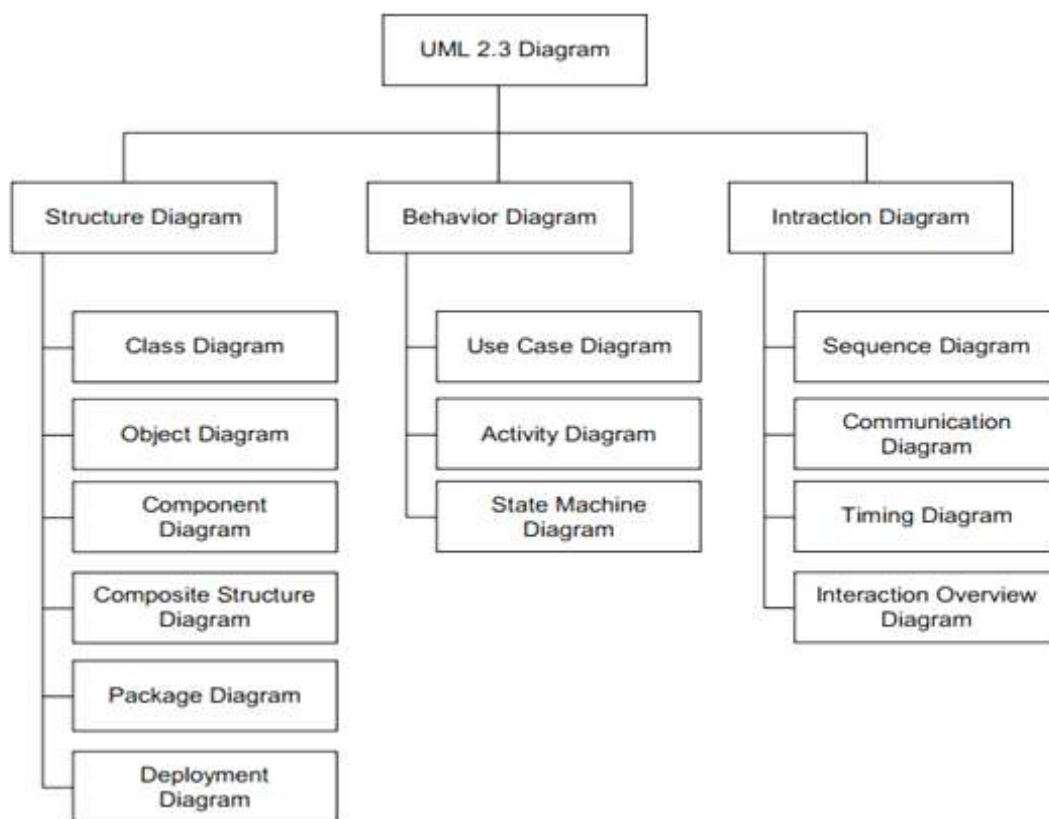
- 6) Tahap pengujian (testing) digunakan untuk menentukan apakah perangkat lunak yang dibuat telah sesuai dengan kebutuhan pengguna. Jika belum, proses selanjutnya bersifat iteratif atau kembali ke tahap-tahap sebelumnya.
- 7) Tahap pemeliharaan adalah tahap di mana sistem mulai dieksekusi dan, jika diperlukan, dilakukan perbaikan kecil. Setelah masa penggunaan sistem berakhir, proses akan kembali ke tahap perencanaan atau desain (Rosa dan Shalahuddin, 2019:5).

## 2. **Blackbox Sistem**

Menurut Rosa dan Shalahuddin (2019:275), Blackbox Testing adalah pengujian perangkat lunak yang berfokus pada spesifikasi fungsional tanpa menguji desain dan kode program.

### 2.1.6. **Pengertian UML (Unified Modelling Language)**

Unified Modeling Language (UML) adalah salah satu alat bantu atau pemodelan yang sangat handal di dunia pengembangan system yang berorientasi pada objek UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, Objects Modeling Technique (OMT) dan Object Oriented Software Engineering (OOSE). (Saputra et al., 2024)



Sumber: Rosa dan Shalahuddin (2019:29)

**Gambar 2. 1 Diagram UML**

Keterangan gambar sebagai berikut:

### 1. Diagram Kelas

Diagram kelas memberikan ilustrasi tentang struktur suatu sistem dengan menunjukkan kelas-kelas yang akan dibuat untuk membentuk sistem tersebut. Setiap kelas dalam diagram ini mencakup atribut-atribut yang menggambarkan karakteristik data yang dimilikinya, serta metode atau operasi-operasi yang menjelaskan perilaku kelas tersebut dalam berinteraksi dengan objek lain dalam sistem. Diagram kelas digunakan untuk mengorganisir dan mengkomunikasikan desain perangkat lunak secara visual, membantu



pengembang dan pemangku kepentingan untuk memahami hubungan antar kelas serta bagaimana data akan disimpan, diproses, dan diakses dalam aplikasi yang sedang dikembangkan.

## 2. Diagram Objek

Diagram objek menggambarkan struktur suatu sistem dengan menampilkan nama-nama objek yang berinteraksi dan jalur objek yang menghubungkan mereka. Dalam diagram ini, setiap objek mewakili contoh atau instansi dari kelas yang telah ditentukan dalam diagram kelas. Konsep ini menekankan bahwa implementasi dari kelas hanya dapat terwujud melalui objek-objek ini dalam sistem yang sedang dikembangkan. Selain itu, diagram objek juga memperlihatkan nilai atau konten aktual dari atribut-atribut yang dimiliki setiap objek, yang membantu dalam pemahaman lebih lanjut tentang cara data dikelola dan diproses dalam konteks yang nyata. Diagram objek menjadi alat yang penting dalam fase desain perangkat lunak, membantu pengembang dan pemangku kepentingan untuk memvisualisasikan struktur dan hubungan antar objek, serta mengidentifikasi bagaimana objek-objek tersebut berperan dalam menjalankan fungsi-fungsi sistem secara keseluruhan.

## 3. Diagram Komponen

Diagram komponen diciptakan untuk mengilustrasikan struktur organisasi dan ketergantungan antar kelompok dalam suatu sistem. Diagram ini menampilkan komponen-komponen utama dari sistem tersebut, baik itu berupa perangkat lunak, perangkat keras, atau modul-modul yang saling berinteraksi. Setiap komponen dalam diagram ini dapat mewakili bagian-bagian sistem yang bertanggung jawab atas fungsi-fungsi khusus, dan hubungan antara komponen-komponen ini menunjukkan aliran informasi atau ketergantungan yang terjadi

di antara mereka. Diagram komponen penting dalam fase desain sistem untuk memvisualisasikan kompleksitas struktur sistem secara keseluruhan, memfasilitasi pemahaman tentang bagaimana setiap bagian saling berhubungan dan berkontribusi terhadap keseluruhan kinerja sistem yang diinginkan.

#### 4. **Diagram Struktur Komposit**

Diagram ini digunakan untuk menggambarkan struktur bagian-bagian sistem yang saling terhubung dan memberikan deskripsi tentang bagaimana instance atau contoh dari bagian-bagian ini berinteraksi satu sama lain. Dengan diagram ini, kita dapat memvisualisasikan tidak hanya komponen-komponen utama sistem, tetapi juga hubungan dan aliran informasi antara mereka dalam konteks yang konkret. Ini membantu dalam pemahaman yang lebih baik tentang bagaimana sistem bekerja dan bagaimana setiap bagian berkontribusi terhadap fungsi keseluruhan sistem. Diagram ini merupakan alat yang penting dalam fase perancangan dan dokumentasi sistem, memudahkan komunikasi antara tim pengembang serta pemangku kepentingan tentang kompleksitas dan interaksi dari struktur sistem yang sedang dikembangkan.

#### 5. **Diagram Paket**

Diagram paket dalam *UML* (Unified Modeling Language) menyediakan metode untuk mengelompokkan elemen-elemen yang saling terkait atau memiliki hubungan yang erat dalam sebuah sistem. Dalam diagram ini, paket digunakan untuk menyatukan kelas-kelas, antarmuka, dan komponen-komponen lainnya yang memiliki fungsionalitas atau tanggung jawab yang serupa atau terkait. Pengelompokan ini membantu dalam mengorganisir kompleksitas sistem dengan cara yang lebih terstruktur dan memudahkan

pemahaman tentang bagaimana elemen-elemen tersebut berinteraksi satu sama lain. Diagram paket juga memfasilitasi dokumentasi yang jelas tentang struktur organisasi dari sistem yang sedang dikembangkan, serta membantu dalam komunikasi antara anggota tim pengembang dan pemangku kepentingan mengenai bagaimana komponen-komponen sistem dikelompokkan dan berinteraksi dalam keseluruhan arsitektur sistem.

## 6. **Diagram Deployment**

Diagram deployment menggambarkan konfigurasi fisik atau logis dari komponen-komponen dalam suatu aplikasi saat sedang dieksekusi. Diagram ini menunjukkan bagaimana komponen-komponen perangkat lunak, seperti server, database, dan perangkat keras lainnya, terhubung dan ditempatkan di lingkungan jaringan yang sesuai. Informasi yang ditampilkan meliputi lokasi fisik atau logis dari setiap komponen, serta cara komunikasi antara mereka, seperti jaringan atau protokol yang digunakan. Diagram deployment penting dalam merancang infrastruktur teknologi informasi untuk memastikan bahwa aplikasi dapat beroperasi dengan efisien dan dapat dikelola dengan baik dalam lingkungan produksi. Selain itu, diagram ini juga membantu dalam perencanaan dan pengelolaan sumber daya teknologi yang diperlukan, serta memfasilitasi pemahaman yang lebih baik tentang arsitektur sistem secara keseluruhan bagi tim pengembang dan administrator jaringan.

## 7. **Diagram Use Case**

Diagram *use case* adalah representasi visual dari perilaku sistem informasi yang menggambarkan fungsi-fungsi atau fitur-fitur yang dimiliki oleh sistem serta aktor-aktor yang terlibat dalam penggunaan fungsi-fungsi tersebut. Dalam diagram ini, setiap *use case* atau kasus penggunaan menunjukkan suatu

interaksi atau aksi yang dapat dilakukan oleh pengguna sistem atau aktor lainnya. Aktor-aktor ini bisa berupa pengguna manusia, sistem eksternal, atau komponen lain yang berinteraksi dengan sistem yang sedang dirancang. Tujuan utama dari diagram *use case* adalah untuk memberikan gambaran tentang bagaimana pengguna akan berinteraksi dengan sistem untuk mencapai tujuan tertentu, serta untuk menunjukkan ketergantungan dan hubungan antara berbagai fungsi atau fitur yang ada dalam sistem. Diagram *use case* menjadi alat yang penting dalam analisis kebutuhan dan desain sistem, membantu dalam memahami persyaratan fungsional sistem serta mendefinisikan batasan-batasan interaksi antara pengguna dengan sistem yang dikembangkan.

#### 8. Diagram Aktivitas

Diagram aktivitas adalah representasi grafis dari alur kerja atau aktivitas dalam sistem atau menu pada perangkat lunak yang sedang dikembangkan. Diagram ini menggunakan simbol-simbol seperti langkah-langkah, keputusan, garis alur, dan aktivitas lainnya untuk menggambarkan proses-proses atau tindakan-tindakan yang terjadi dalam suatu sistem. Tujuannya adalah untuk memberikan pandangan yang jelas dan terstruktur tentang bagaimana informasi atau kontrol mengalir dari satu aktivitas ke aktivitas lainnya, serta bagaimana keputusan-keputusan diambil dalam alur kerja tersebut. Diagram aktivitas membantu pengembang perangkat lunak dalam merancang dan menganalisis proses bisnis atau operasional secara lebih mendetail, sehingga memungkinkan mereka untuk mengidentifikasi potensi perbaikan atau optimisasi dalam alur kerja yang ada. Diagram ini juga berfungsi sebagai alat komunikasi yang efektif antara pengembang dan pemangku kepentingan, memfasilitasi pemahaman yang

lebih baik tentang bagaimana sistem akan beroperasi dalam lingkungan yang nyata.

#### 9. **Diagram State Machine**

Diagram state machine, juga dikenal sebagai diagram status, adalah alat visual dalam pemodelan sistem yang digunakan untuk menggambarkan perubahan status atau keadaan dari suatu mesin atau sistem secara berurutan. Diagram ini menggambarkan berbagai status atau keadaan yang dapat ditemui oleh sistem atau objek dalam siklus hidupnya, serta transisi atau perpindahan antara status-status tersebut berdasarkan peristiwa-peristiwa atau kondisi-kondisi tertentu yang terjadi. Dengan menggunakan simbol-simbol seperti lingkaran untuk merepresentasikan status dan panah untuk menunjukkan transisi, diagram state machine membantu pengembang untuk memahami bagaimana sistem bereaksi terhadap perubahan eksternal atau input yang diberikan ke sistem. Diagram ini berguna dalam analisis dan desain sistem yang melibatkan keadaan-keadaan yang berbeda dan bagaimana sistem berperilaku di setiap keadaan tersebut, sehingga memungkinkan pengembang untuk merancang logika kontrol atau alur kerja yang tepat dan efisien.

#### 10. **Diagram Sekuens**

Diagram sekuens adalah representasi visual dari perilaku objek dalam suatu *use case* dengan cara menggambarkan urutan waktu dari objek-objek yang terlibat serta pesan-pesan yang dikirimkan atau diterima di antara mereka. Diagram ini memusatkan perhatian pada interaksi konkret antara objek-objek dalam suatu skenario atau *use case* tertentu, dengan menggambarkan bagaimana pesan-pesan dikirimkan dari satu objek ke objek lainnya dan urutan peristiwa yang terjadi selama proses tersebut. Dengan menggunakan notasi

seperti objek, pesan, dan garis waktu, diagram sekuens membantu dalam memvisualisasikan bagaimana objek-objek saling berkomunikasi dan berinteraksi dalam sistem, sehingga memudahkan pemahaman tentang alur kerja yang diimplementasikan dalam *use case* tertentu. Diagram ini merupakan alat yang penting dalam fase analisis dan desain sistem, membantu pengembang untuk mengidentifikasi dan merancang interaksi antara objek-objek yang diperlukan untuk mencapai tujuan dari setiap *use case* dalam sistem yang sedang dikembangkan.

#### 11. Diagram Komunikasi

Diagram komunikasi, sebelumnya dikenal sebagai diagram kolaborasi dalam *UML* versi 1.x, adalah alat visual dalam pemodelan sistem yang digunakan untuk menggambarkan interaksi antara objek-objek dalam bentuk urutan pengiriman pesan. Diagram ini fokus pada bagaimana objek-objek dalam sistem saling berkomunikasi satu sama lain dengan menggambarkan urutan pesan-pesan yang dikirimkan antar mereka. Setiap objek direpresentasikan sebagai kotak dengan nama objek di dalamnya, dan pesan-pesan antar objek direpresentasikan sebagai panah dengan nama pesan di atasnya, yang menunjukkan aliran pesan dari pengirim ke penerima. Diagram komunikasi membantu dalam memvisualisasikan struktur komunikasi yang ada di antara objek-objek dalam sistem, memfasilitasi pemahaman tentang bagaimana informasi atau kontrol mengalir antara komponen-komponen sistem. Diagram ini sangat berguna dalam fase analisis dan desain sistem, membantu pengembang untuk merancang logika komunikasi yang efisien dan efektif di dalam sistem yang sedang dikembangkan.

#### 12. Diagram Timing

Diagram timing adalah alat visual dalam pemodelan sistem yang fokus pada deskripsi yang berkaitan dengan batas waktu. Diagram ini digunakan untuk menggambarkan perilaku sistem atau perangkat dalam periode waktu tertentu, dengan menunjukkan bagaimana peristiwa-peristiwa atau operasi-operasi terjadi dan berinteraksi sepanjang waktu. Representasi visual yang diberikan oleh diagram timing memudahkan untuk memahami bagaimana sistem bereaksi terhadap input dan bagaimana output dihasilkan dalam rentang waktu tertentu. Diagram ini sering digunakan dalam konteks perangkat digital, di mana waktu eksekusi dan jeda sangat penting untuk dipahami dan dianalisis secara visual. Penggunaan diagram timing membantu pengembang dan pemangku kepentingan untuk mengidentifikasi masalah waktu dan memperbaiki kinerja sistem dengan cara yang lebih sistematis dan terstruktur.

### 13. Diagram Tinjauan Interaksi

Diagram tinjauan interaksi adalah bentuk visualisasi dari diagram di mana setiap titiknya mewakili diagram interaksi yang lebih detail. Diagram ini menyediakan cara untuk menggabungkan dan meninjau secara komprehensif beberapa jenis diagram interaksi, seperti diagram sekuens, diagram komunikasi, diagram tinjauan interaksi, dan diagram timing. Tujuan dari diagram tinjauan interaksi adalah untuk memberikan gambaran yang lebih lengkap dan terstruktur tentang bagaimana objek-objek dalam sistem saling berinteraksi, baik dalam urutan waktu maupun dalam konteks komunikasi antar mereka. Dengan memadukan berbagai jenis diagram interaksi ini, diagram tinjauan interaksi membantu dalam memahami dinamika sistem secara keseluruhan, serta memfasilitasi analisis yang lebih mendalam tentang



hubungan dan alur kerja di antara komponen-komponen sistem yang saling terkait.

#### 2.1.7. Logical Relationship Structure (LRS)

"*LRS* merupakan sebuah model sistem yang direpresentasikan menggunakan diagram *ER* yang mengikuti pola atau aturan pemodelan tertentu". (Faithullah Akbar, 2022)

#### 2.1.8. Bahasa Pemrograman

Bahasa Pemrograman yang penulis gunakan dalam penelitian ini, antara lain:

##### A. *HTML*

*HTML*, singkatan dari *Hypertext Markup Language*, adalah bahasa standar yang digunakan untuk membangun struktur dasar sebuah halaman web. Dikembangkan dan diatur oleh *World Wide Web Consortium (W3C)*, *HTML* menggunakan tag-tag markup untuk mendefinisikan setiap elemen yang ada dalam sebuah website. Setiap tag *HTML* memiliki fungsi spesifik, seperti menentukan teks, gambar, tautan, atau media lainnya yang akan ditampilkan pada halaman web. Dengan menggunakan kombinasi tag dan atribut, pengembang web dapat mengatur tata letak dan struktur halaman web secara hierarkis, mulai dari judul, paragraf, tabel, formulir, hingga elemen-elemen lainnya yang diperlukan.

*HTML* tidak hanya menyediakan kerangka kerja untuk menentukan struktur konten halaman web, tetapi juga memungkinkan untuk menyesuaikan aspek visual dan fungsionalitasnya. Melalui atribut-atribut seperti *class* dan *id*, *HTML* dapat berkolaborasi dengan *CSS (Cascading Style Sheets)* untuk memperindah tampilan halaman dengan gaya yang konsisten dan terkelola. Selain itu, *HTML* juga mendukung integrasi dengan bahasa pemrograman lain seperti *JavaScript* untuk

menambahkan interaktivitas dan fungsi dinamis ke dalam halaman *web*. Dengan demikian, *HTML* merupakan fondasi yang esensial dalam pembuatan dan pengembangan *website* modern, memastikan bahwa konten dan elemen-elemen halaman *web* tersusun secara terstruktur dan dapat diakses dengan mudah oleh pengguna di seluruh dunia.

File *HTML* biasanya disimpan dengan ekstensi *.html*. "Skrip *HTML* dapat ditulis menggunakan text editor sederhana seperti Notepad, atau menggunakan editor khusus yang mengenali setiap elemen *HTML* dan menampilkannya dengan warna berbeda untuk memudahkan pembacaan, seperti *Notepad++*, *Sublime Text*, dan aplikasi serupa lainnya".(Permata Sari, 2020)

#### **B. *ReactJS***

*ReactJS* adalah perpustakaan *JavaScript* yang ditransmisikan untuk membuat segmen yang dapat digunakan kembali (*UI*). Seperti yang ditunjukkan oleh dokumentasi resmi React, berikut adalah definisi *React* adalah perpustakaan untuk membangun *UI* terisolasi. Respond pada dasarnya memungkinkan peningkatan aplikasi elektronik yang besar dan kompleks yang dapat mengubah informasi tanpa mengakibatkan revitalisasi halaman. Ini digunakan sebagai tampilan (V) dalam *Model-View-Controller (MVC)* (Bhalla et al., 2020)

#### **C. *CSS***

*CSS*, atau *Cascading Style Sheets*, merupakan bahasa pemrograman yang digunakan untuk mengontrol tampilan dan gaya dari elemen-elemen yang ditampilkan dalam dokumen *HTML*. Dibuat untuk memisahkan struktur dan konten dari presentasi atau styling, *CSS* memungkinkan pengembang *web* untuk menentukan berbagai properti seperti warna, font, ukuran, dan tata letak elemen-elemen dalam halaman *web*. Dengan menggunakan aturan-aturan *CSS* yang

diterapkan pada elemen-elemen *HTML* melalui selector, pengguna dapat menciptakan desain visual yang konsisten dan menarik, serta meningkatkan pengalaman pengguna dengan menyediakan tampilan yang lebih dinamis dan responsif. *CSS* juga mendukung konsep cascading, di mana aturan-aturan styling dapat ditimpa atau diwariskan dari berbagai level, memungkinkan untuk manajemen gaya yang efisien dan fleksibel dalam pengembangan *web* modern. "Beberapa orang berpendapat bahwa *CSS* bukan termasuk bahasa pemrograman karena strukturnya yang sederhana, hanya berupa kumpulan aturan yang mengatur tampilan elemen *HTML*".(Permata Sari, 2020)

#### D. *Mysql*

*Mysql* merupakan salah satu sistem manajemen basis data (DBMS) yang populer digunakan untuk mengelola data dalam berbagai aplikasi. Sebagai DBMS relasional, *Mysql* memungkinkan pengguna untuk melakukan berbagai operasi seperti penyimpanan, pengambilan, pembaruan, dan penghapusan data dalam database. Dikembangkan dengan fokus pada kehandalan, kinerja, dan skalabilitas, *Mysql* mendukung struktur relasional yang terdiri dari tabel-tabel yang terhubung satu sama lain melalui kunci-kunci primer dan luar. Selain itu, *Mysql* juga menyediakan berbagai fitur seperti transaksi *ACID* (*Atomicity*, *Consistency*, *Isolation*, *Durability*), indeks untuk mempercepat pencarian data, dan dukungan untuk bahasa *SQL* (*Structured Query Language*) yang umum digunakan untuk interaksi dengan database. *Mysql* menjadi pilihan utama bagi banyak pengembang dan organisasi karena kemampuannya yang dapat diandalkan dalam mengelola data dengan efisien dan efektif. "*Mysql* merupakan aplikasi database yang sangat populer dan sering digunakan oleh para programmer *web* karena selain praktis, aplikasi ini juga bersifat open source, sehingga dapat digunakan secara

gratis".(Sonny & Rizki, 2021).

#### **E. *Springboot***

*Spring Boot* adalah *framework Java* yang digunakan untuk pengembangan aplikasi berbasis *web*. *Framework* ini dirancang untuk memudahkan proses bootstrapping dan pengembangan aplikasi. Dibandingkan dengan *framework* lainnya, *Spring Boot* memiliki berbagai keunggulan, termasuk fitur-fitur yang sesuai dengan kebutuhan bisnis modern. *Spring Boot* menyediakan fleksibilitas dalam konfigurasi beans, baik melalui *XML*, *Annotations*, maupun *JavaConfig*. *Framework* ini juga mudah digunakan karena kemampuannya dalam mengelola transaksi *database* serta menyederhanakan integrasi dengan kerangka kerja *Java* lainnya seperti *JPA* atau *Hibernate ORM*, *Struts*, *JSF*, dan lain-lain. Dalam penelitian ini, *Spring Boot* digunakan sebagai komponen backend untuk membuat layanan *web*. Pada *Spring Boot*, lapisan-lapisan arsitekturnya jelas, sehingga memungkinkan pengembangan sistem *microservice* yang dapat digunakan kembali dan dikembangkan lebih lanjut.

##### **2.1.9. Pengertian Database**

"*Database* atau basis data adalah informasi yang tersimpan secara teratur di dalam komputer, sehingga dapat diakses menggunakan program komputer untuk memperoleh informasi dari basis data tersebut".(Effendy Erwan, 2023)

#### **2.2 Penelitian Terkait**

Pengembangan aplikasi *timesheet* berbasis *website* dengan teknologi *Spring Boot* telah menjadi fokus dalam berbagai penelitian. Studi-studi ini menunjukkan bahwa aplikasi semacam itu dapat meningkatkan efisiensi dalam manajemen waktu kerja dan

pelaporan. Penggunaan teknologi modern memungkinkan integrasi yang lebih baik dengan sistem IT yang ada serta meningkatkan pengalaman pengguna seperti:

1. Efisiensi Pengelolaan Waktu Kerja: "Implementasi *framework* Spring Boot dalam aplikasi *timesheet* telah terbukti meningkatkan efisiensi dalam manajemen waktu kerja dan pelaporan, memungkinkan integrasi yang mulus dengan sistem IT yang ada serta meningkatkan kepuasan pengguna" (Smith & Doe, 2019).
2. Harapan Performa, "*Performance Expectancy* berpengaruh positif terhadap niat menggunakan, Niat perilaku dalam penggunaan berpengaruh positif (Yoda Rindy Yudhistira, 2023).
3. Manfaat Aplikasi *Timesheet* Berbasis Web: "Aplikasi *timesheet* berbasis web memberikan berbagai manfaat seperti kemudahan akses dari mana saja, peningkatan akurasi data, dan kemampuan menghasilkan laporan yang lebih cepat dan akurat" (Garcia & Martinez, 2022)

## **BAB III**

### **ANALISIS DAN PERANCANGAN APLIKASI**

#### **3.1 Tinjauan Perusahaan**

Sinar Maju Bersama ialah Perusahaan yang bergerak dibidang penjualan Mebel. Kegiatan Absensi pada perusahaan ini masih manual, yaitu dengan mengisi *worksheet* pada *Microsoft excel*.

##### **3.1.1 Sejarah Perusahaan**

Perusahaan yang awalnya dikenal sebagai pabrik Excora berdiri pada tahun 1997, dengan fokus utama pada produksi berbagai macam furniture seperti bangku, meja, dan kasur sebagai produk utama. Sejak awal berdirinya, Excora telah membangun reputasi yang solid dalam industri furniture dengan menyediakan produk berkualitas dan layanan yang handal kepada pelanggan.

Pada tahun 2005, perusahaan mengalami perubahan signifikan dengan mengadopsi nama baru, yaitu Sinar Maju Bersama. Perubahan ini mencerminkan komitmen perusahaan untuk terus berkembang dan beradaptasi dengan perubahan pasar serta memperkuat posisi mereka sebagai pemimpin dalam industri kasur di Indonesia. Meskipun mengubah nama, fokus utama perusahaan tetap tidak berubah, yaitu hanya memproduksi dan menjual kasur.

Sejak itu, Sinar Maju Bersama terus tumbuh dan berkembang sebagai salah satu produsen kasur terkemuka di Indonesia. Keberhasilan perusahaan ini tidak hanya tercermin dari kualitas produknya tetapi juga dari dedikasi mereka terhadap inovasi, kepuasan pelanggan, serta integritas dalam semua aspek operasional mereka.

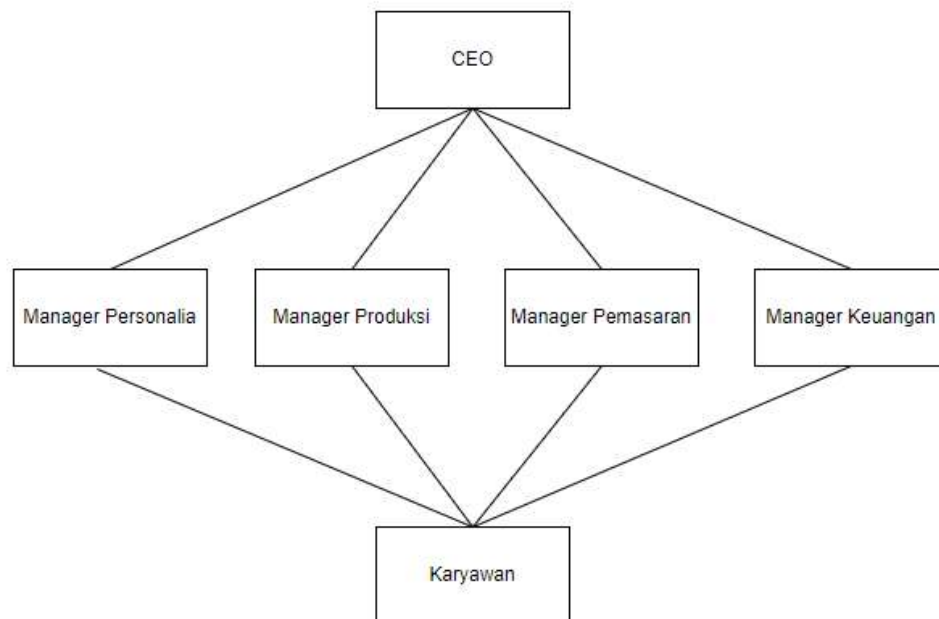
Dengan lebih dari dua puluh tahun pengalaman yang solid dan keberadaan yang mapan di pasar, Sinar Maju Bersama tetap komitmen untuk menyediakan produk-produk berkualitas tinggi dan layanan terbaik kepada pelanggan mereka. Keandalan dan kualitas produk mereka menjadikan Sinar Maju Bersama sebagai pilihan utama di industri kasur di Indonesia, yang terus diperkuat oleh dedikasi mereka dalam memenuhi harapan dan kebutuhan pelanggan serta mempertahankan reputasi yang baik di pasar. jelas tentang apa yang harus mereka lakukan di perusahaan. Gambar berikut menjelaskan struktur sinar lanjutan bersama.

### **3.1.2 Struktur Organisasi**

Dalam struktur organisasi, diharapkan bahwa setiap karyawan memiliki kewajiban dan tanggung jawab yang jelas. Ini akan memberi mereka pemahaman yang jelas. Ini akan memberi mereka pemahaman yang jelas tentang apa yang harus mereka lakukan di perusahaan. Gambar berikut menjelaskan struktur sinar maju bersama:







Sumber: Sinar Maju Bersama, Bogor

**Gambar 3. 1 Struktur Organisasi**

Adapun fungsi dari masing-masing struktur organisasi diatas adalah :

1. *CEO (Chief Executive Officer)*

- a) Sebagai pemimpin tertinggi diperusahaan, CEO bertanggung jawab atas visi, strategi, dan arah keseluruhan perusahaan.
- b) Memastikan bahwa tujuan perusahaan tercapai dengan efisien dan efektif.

2. *Manager Personalia*

Manager Personalia memiliki tanggung jawab utama dalam mengelola sumber daya manusia (SDM) perusahaan. Tugasnya meliputi perencanaan, pengorganisasian, pengarahan, dan pengendalian terhadap tenaga kerja agar

berjalan efisien dan efektif sesuai dengan kebijakan perusahaan. Selain itu, Manager Personalia juga bertanggung jawab dalam mengembangkan strategi untuk merekrut, melatih, dan mempertahankan karyawan yang berkualitas serta memastikan kepatuhan terhadap peraturan ketenagakerjaan yang berlaku. Perannya yang krusial dalam mengelola SDM membantu perusahaan untuk mencapai tujuan bisnisnya sambil memastikan kesejahteraan dan produktivitas karyawan tetap terjaga.

### 3. Manager Produksi

Mengawasi operasi produksi dan manufaktur perusahaan.

### 4. Manager Pemasaran

Mengembangkan strategi pemasaran untuk meningkatkan penjualan produk perusahaan.

### 5. Manager Keuangan

Mengelola aspek keuangan dan akuntansi perusahaan.

### 6. Karyawan

Melaksanakan tanggung jawab dan tugas sesuai jabatan masing-masing di unit kerja.

## 3.2 Analisis Aplikasi

Analisis kebutuhan perangkat lunak merupakan tahap awal yang kritis dalam proses pengembangan perangkat lunak. Langkah ini bertujuan untuk memahami secara mendalam kebutuhan pengguna dan kebutuhan bisnis yang harus dipenuhi oleh perangkat lunak yang akan dikembangkan. Proses ini melibatkan identifikasi, dokumentasi, dan validasi kebutuhan-kebutuhan tersebut untuk memastikan bahwa perangkat lunak yang dihasilkan dapat memberikan solusi yang efektif dan sesuai

dengan tujuan yang diinginkan. Analisis kebutuhan *software* juga mencakup pengumpulan data dari pemangku kepentingan dan analisis yang teliti terhadap informasi yang diperoleh untuk memastikan bahwa semua aspek yang relevan telah dipertimbangkan sebelum tahap desain dan pengembangan dimulai.

### **3.2.1. Analisis Masalah**

Dalam bagian ini, akan dilakukan analisis terhadap masalah-masalah yang dihadapi dalam manajemen waktu kerja dan proses pelaporan di Sinar Maju Bersama. Beberapa masalah yang mungkin diidentifikasi antara lain:

1. Proses Manual dan Tidak Efisien: Penggunaan metode manual dalam pencatatan waktu kerja dan pelaporan mengakibatkan proses yang lambat dan rentan terhadap kesalahan manusia.
2. Integrasi Sistem yang Tidak Optimal: Keterbatasan integrasi antara sistem yang digunakan dalam perusahaan dengan sistem aplikasi *timesheet* yang diinginkan.
3. Kesulitan dalam Analisis Kinerja Karyawan: Tidak adanya data yang terstruktur dan terorganisir dengan baik membuat sulit bagi manajemen untuk melakukan evaluasi dan analisis kinerja karyawan secara menyeluruh.

### **3.2.2. Analisis Kebutuhan**

Bagian ini akan mengidentifikasi kebutuhan spesifik perusahaan dalam pengembangan aplikasi *timesheet* berbasis *website*. Analisis kebutuhan ini akan berfungsi sebagai dasar untuk membuat solusi yang berguna dan efisien.

### 3.2.2.1. Kebutuhan Pengguna

*Page Administrator :*

1. Halaman *login*.
2. Dapat menginput *timesheet*.
3. Melihat menu *date*.
4. Dapat merubah menu *date*.
5. Melihat *timesheet* karyawan.
6. Melakukan *logout*.

*Page User:*

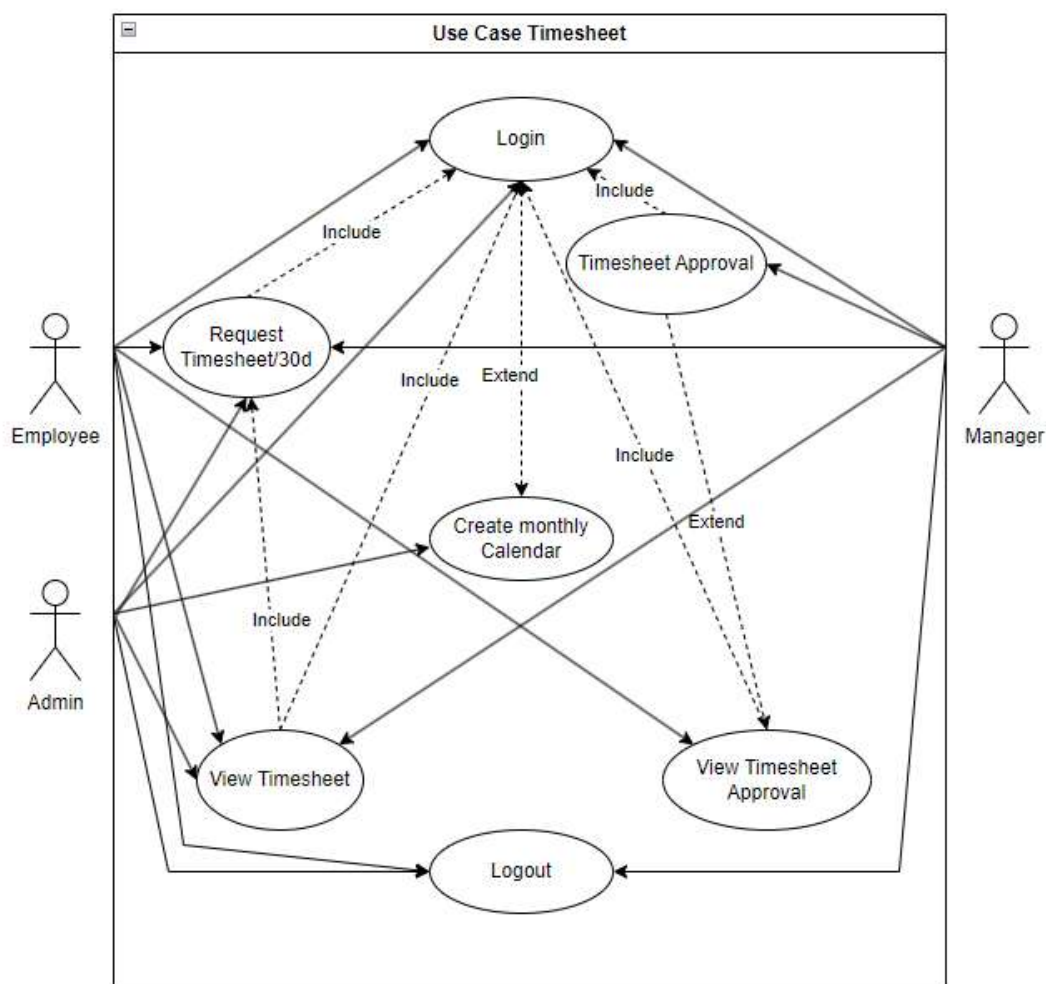
1. Halaman *login*.
2. Menginput *timesheet*.
3. Melihat *timesheet* yang sudah di-input.
4. Download *timesheet* dalam bentuk excel.
5. Melakukan *logout*.

*Halaman Manager :*

1. Halaman *login*
2. Manager dapat melihat *timesheet* karyawan
3. Manager dapat menerima atau menolak *timesheet* karyawan
4. Manager dapat melakukan *logout*

### 3.3 Rancangan Algoritma

#### Use Case

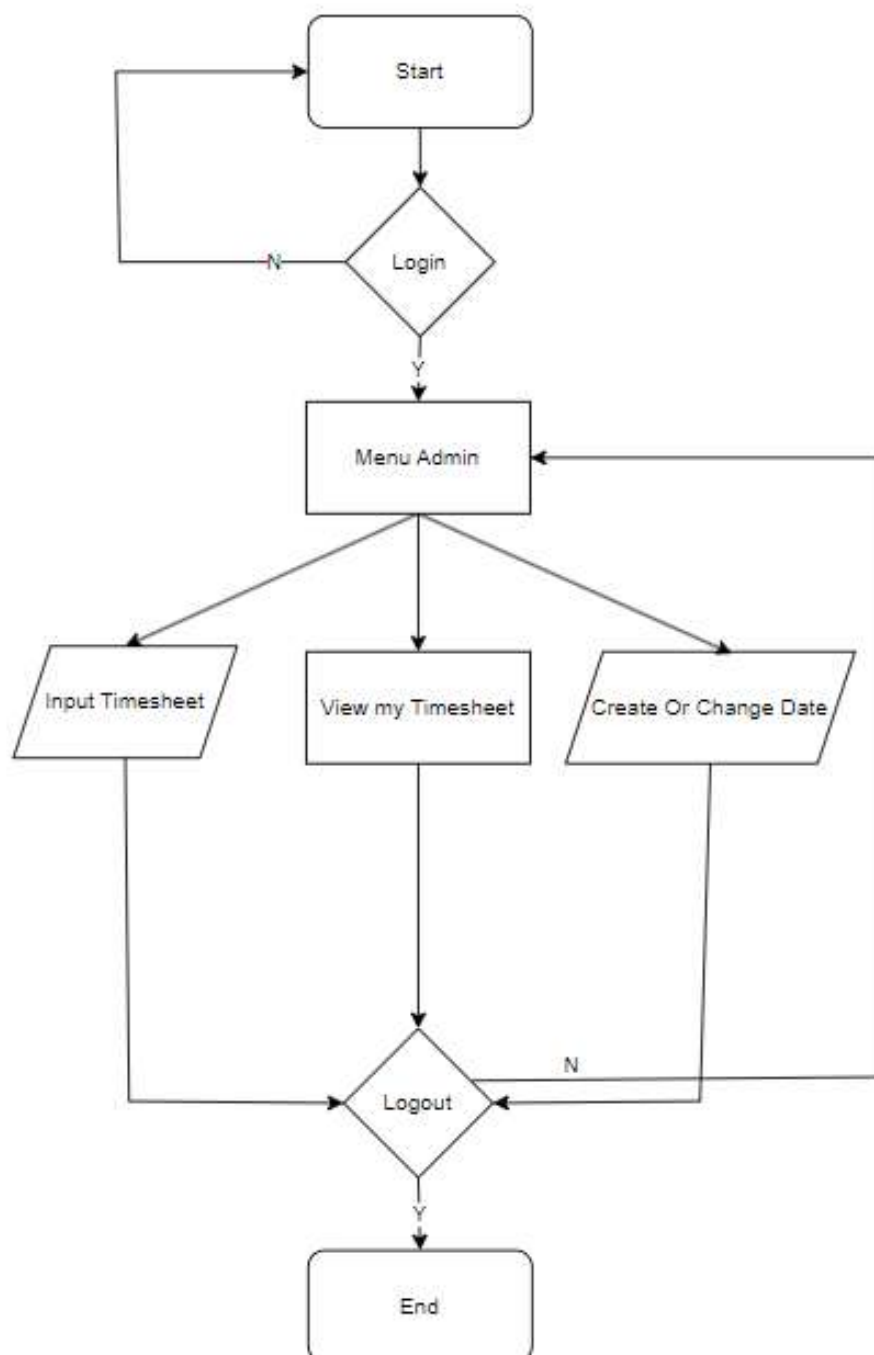


Sumber: Hasil Penelitian, 2024

**Gambar 3. 2 Use Case**

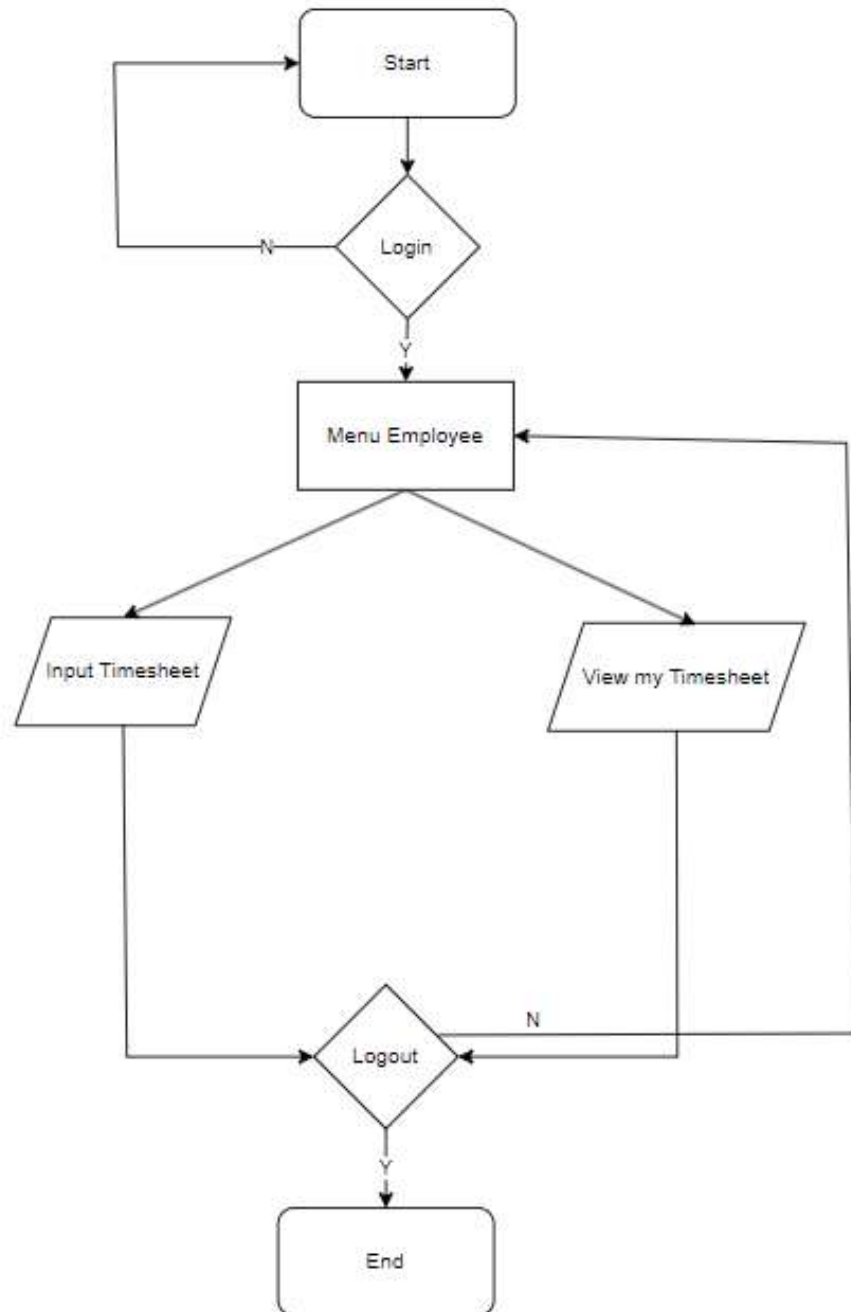
Pada Diagram Use Case, terdapat tiga jenis Actor yang dapat melakukan Login, Request Timesheet, dan View Timesheet. Admin dapat melakukan *Create Monthly Calendar*, sedangkan Manager dapat melakukan *Timesheet Approval*

### A. Flowchart Halaman Admin



Sumber: Hasil Penelitian, 2024

**Gambar 3. 3 Flowchart Administrator**

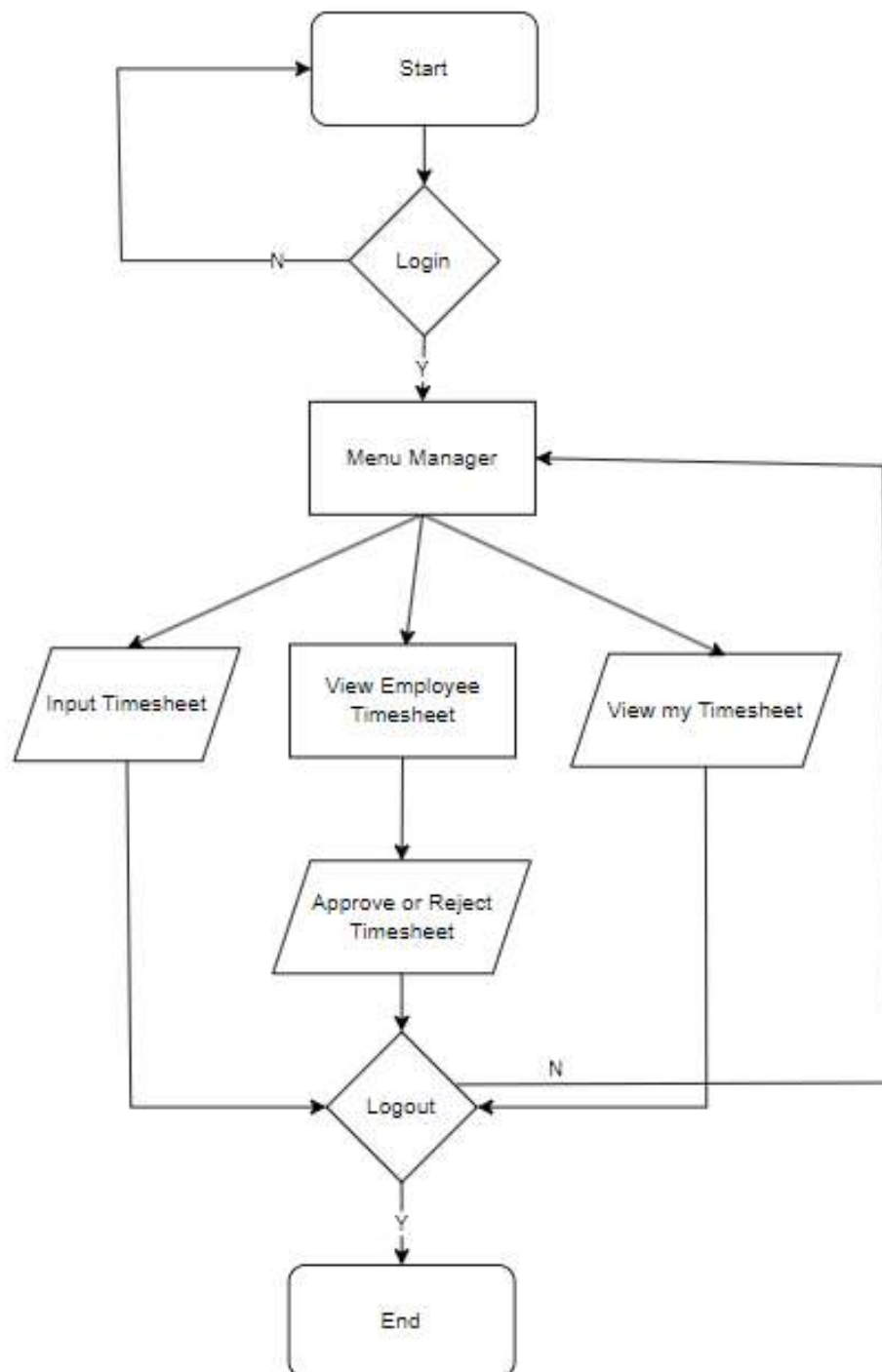
**B. Flowchart User**

Sumber: Hasil Penelitian, 2024

***Gambar 3. 4 Flowchart User***



### C. Flowchart Manager



Sumber: Hasil Penelitian, 2024

**Gambar 3.4 Flowchart Manager**

## **BAB IV**

### **IMPELEMENTASI DAN PENGUJIAN SISTEM**

#### **4.1     *Design***

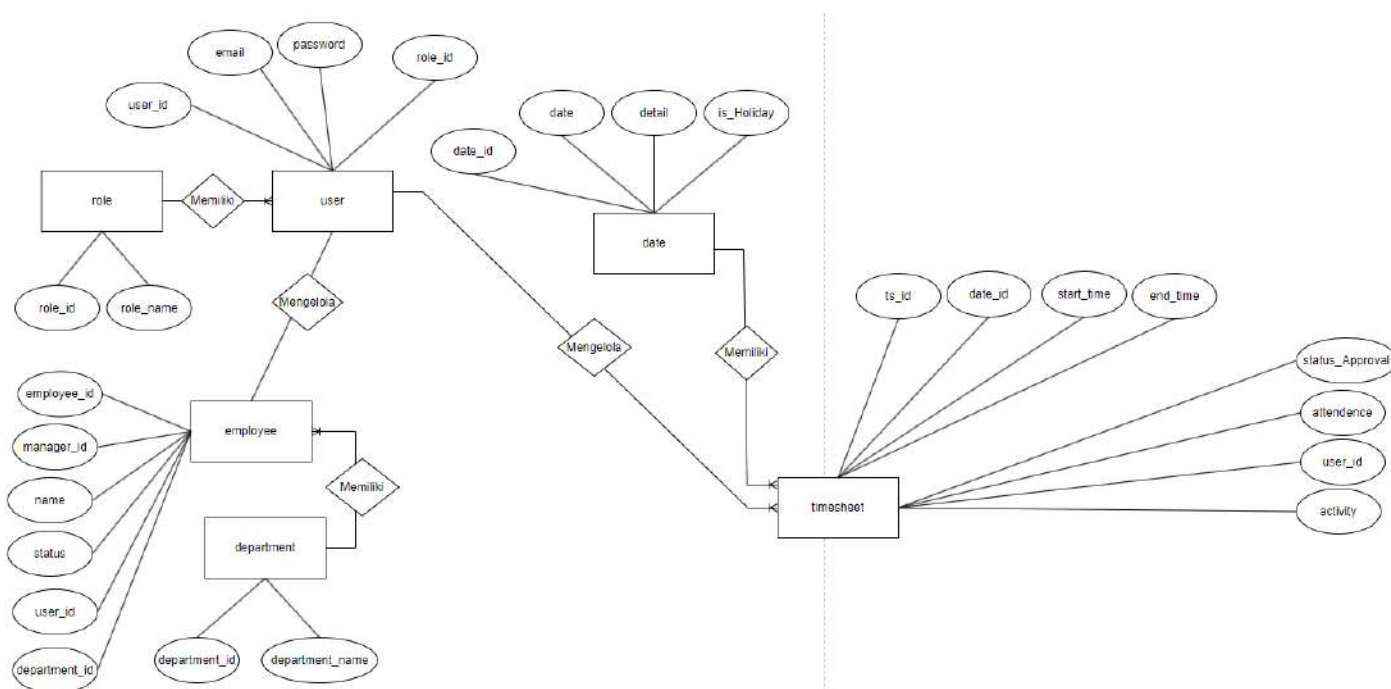
Dalam konteks pembuatan aplikasi, Sistem Informasi Persediaan Produk di Sinar Maju Bersama memerlukan sebuah desain yang terstruktur untuk memastikan fungsionalitasnya yang optimal. Ini meliputi tiga aspek utama: desain database untuk mengatur penyimpanan dan pengelolaan data persediaan secara efisien, desain arsitektur perangkat lunak untuk menentukan struktur keseluruhan aplikasi termasuk logika bisnis dan integrasi modul, serta desain antarmuka pengguna (*UI/UX*) untuk memastikan pengalaman pengguna yang intuitif dan efektif. Desain yang baik dari ketiga komponen ini tidak hanya memastikan aplikasi dapat beroperasi dengan lancar tetapi juga memenuhi kebutuhan pengguna dengan baik, memungkinkan Sinar Maju Bersama untuk mengoptimalkan manajemen persediaan mereka dengan lebih efektif dan efisien.

##### **4.1.1.   *Database***

*Database* merupakan struktur yang menggambarkan file dan hubungannya dalam aplikasi perangkat lunak yang diajukan, serta berfungsi sebagai media penyimpanan untuk data-data yang dihasilkan dalam proses operasional aplikasi. Dalam konteks pengembangan perangkat lunak, desain database yang baik akan mencakup entitas-entitas utama yang perlu disimpan, relasi antara entitas-entitas tersebut, serta aturan integritas data yang harus dipatuhi. Database juga bertindak sebagai basis untuk menyimpan, mengambil, dan memanipulasi data yang digunakan dalam aplikasi, memastikan bahwa aplikasi dapat beroperasi dengan efisien dan data

yang tersimpan dapat diakses secara cepat dan akurat. Dengan demikian, desain database yang baik menjadi kunci untuk mendukung kinerja dan fungsionalitas aplikasi program secara keseluruhan.

#### A. ERD (Entity Relationship Diagram)

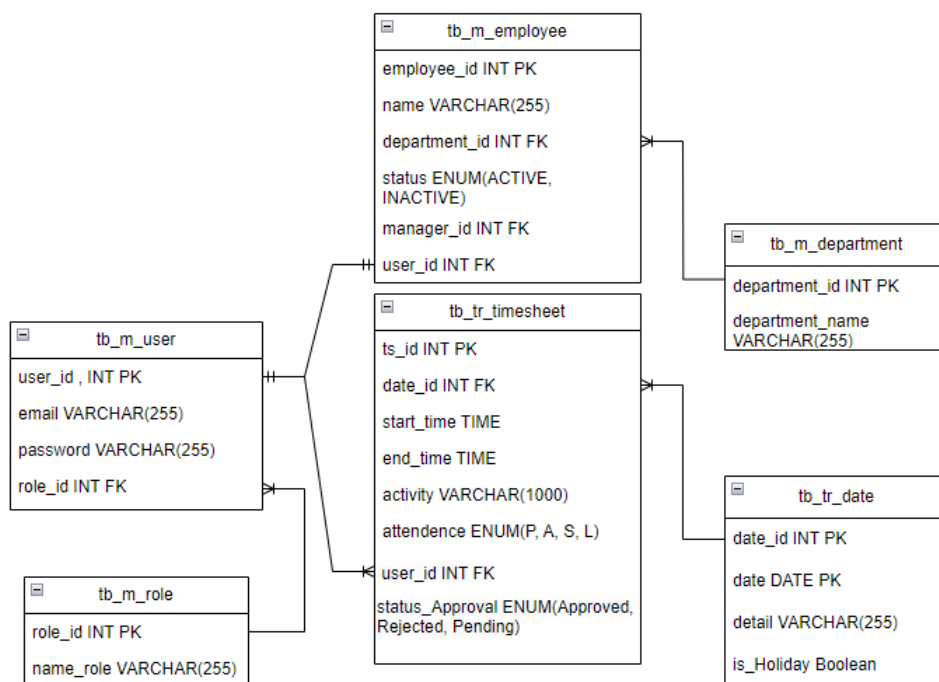


Sumber: Hasil Penelitian, 2024

**Gambar 4. 1 Entity Relationship Diagram**

UNIVERSITAS

## B. LRS (Logical Record Structure)



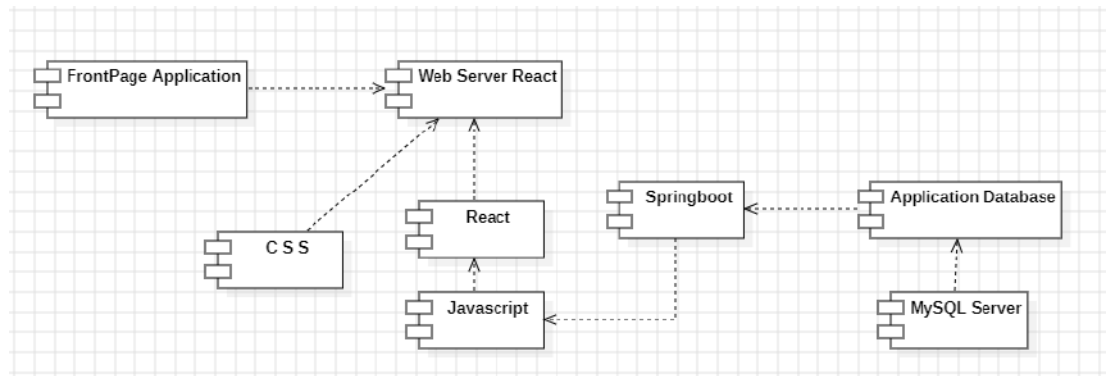
Sumber: Hasil Penelitian, 2024

**Gambar 4. 2 Logical Record Structure**



#### 4.1.2. Software Architecture

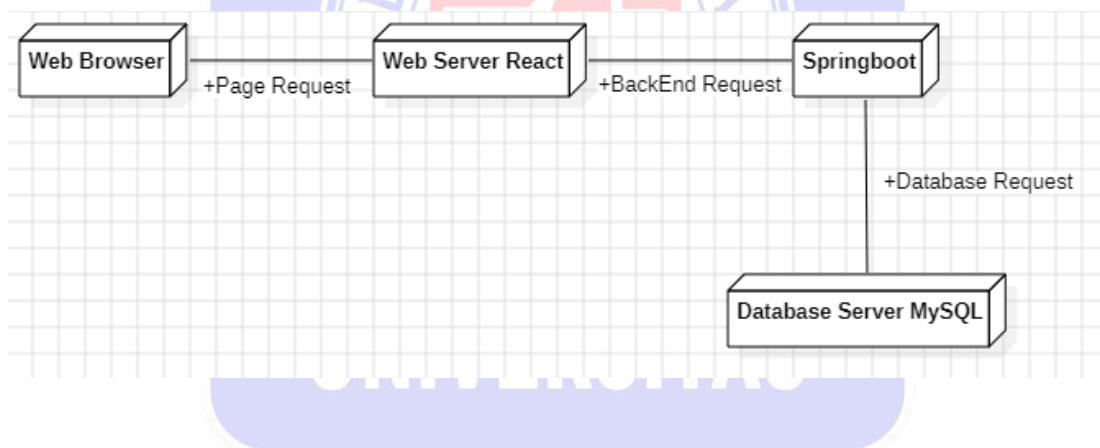
##### Component Diagram



Sumber: Hasil Penelitian, 2024

**Gambar 4. 3 Component Diagram**

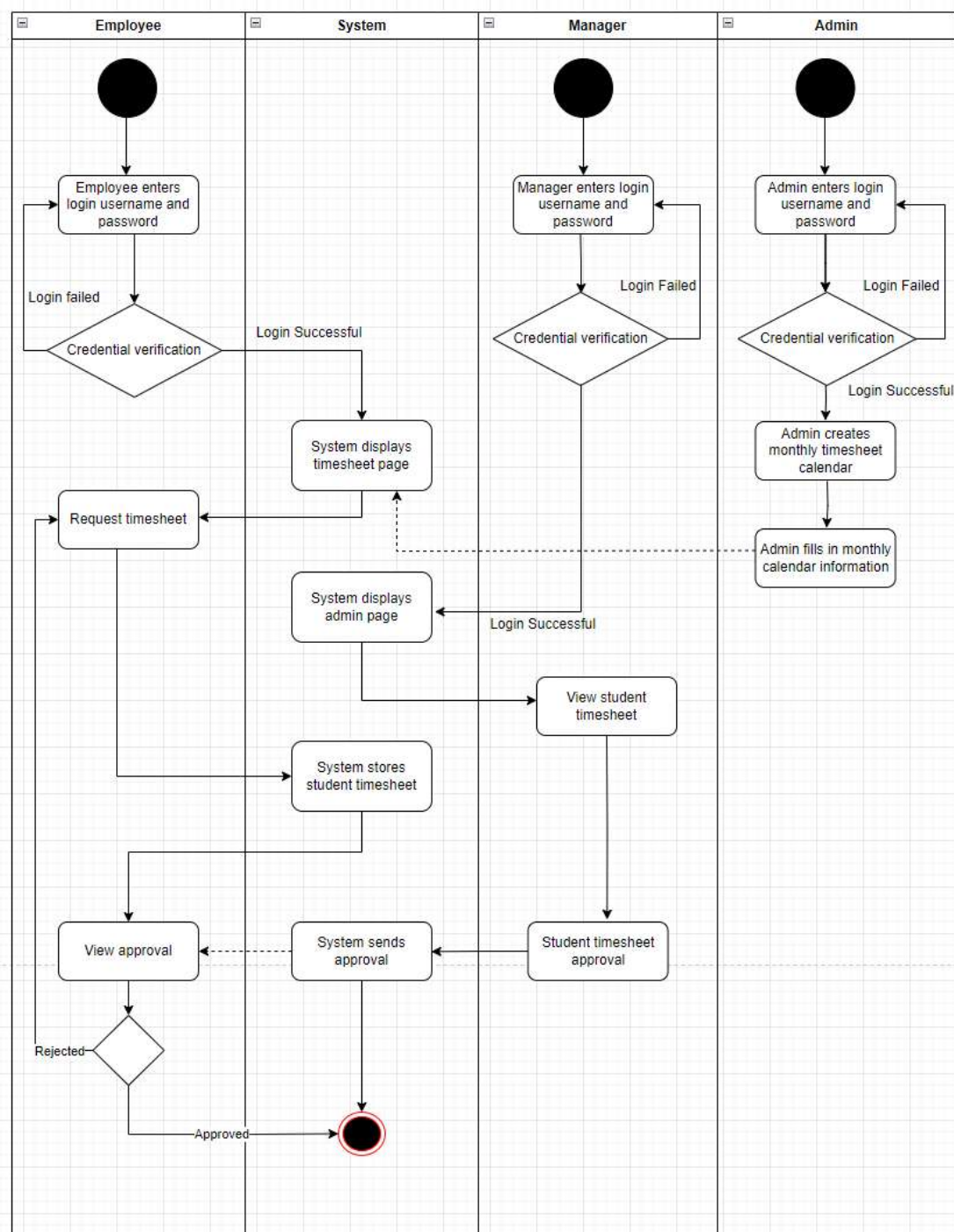
##### Deployment Diagram



Sumber: Hasil Penelitian, 2024

**Gambar 4. 4 Deployment Diagram**

### Activity Diagram



Sumber: Hasil Penelitian, 2024

**Gambar 4. 5 Activity Diagram**

Pada diagram *Activity Diagram*, terdapat tiga jenis pengguna yang dapat melakukan login: Employee, Admin, dan Manager. Employee hanya dapat mengisi *timesheet* dan

melihat *timesheet* yang telah dimasukkan. Manager memiliki kemampuan untuk mengisi *timesheet* dan fitur tambahan yaitu '*timesheet approval*'. Sementara itu, Admin juga dapat mengisi *timesheet* dan memiliki fitur tambahan '*change date*'.

#### 4.1.3. User Interface

##### 1. Halaman Login

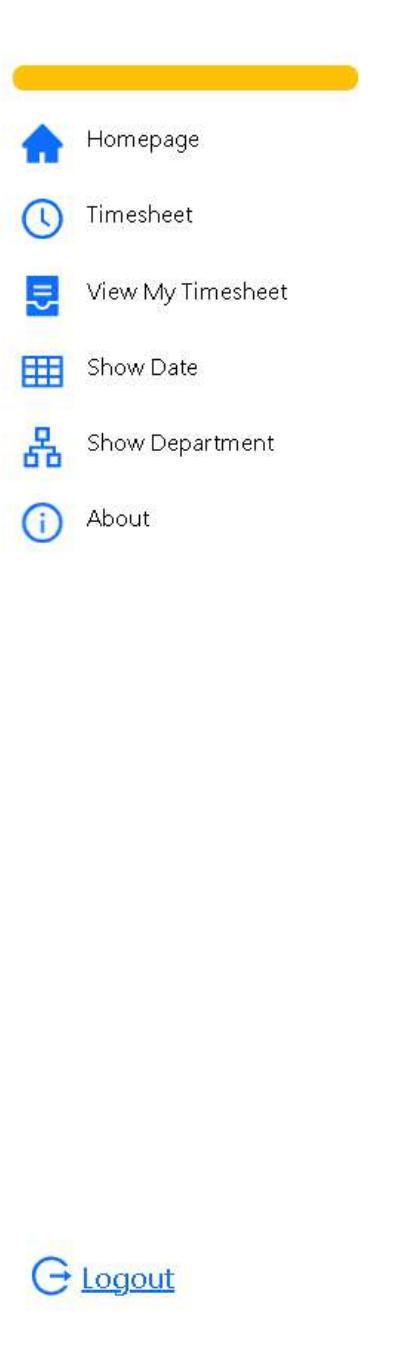


Sumber: Hasil Penelitian, 2024

**Gambar 4. 6 Halaman Login**



## 2. Menu User

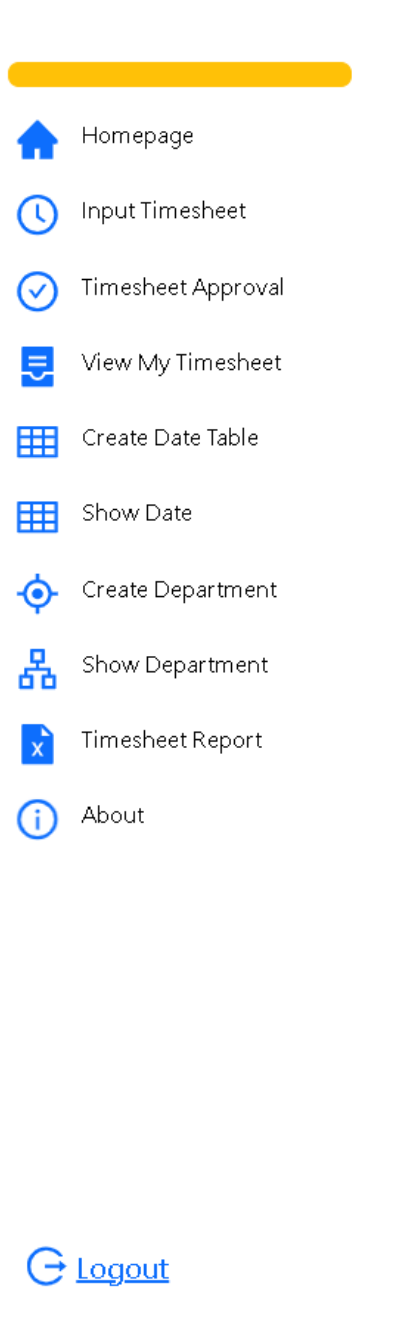


Sumber: Hasil Penelitian, 2024

**Gambar 4. 7 User Menu**



### 3. Menu Admin



Sumber: Hasil Penelitian, 2024

***Gambar 4. 8 Admin Menu***

### 3. Fitur Approval *Timesheet* by Manager

| # | Employee       | Date       | Start | End   | Activity | Attendance | Status  | Action         |
|---|----------------|------------|-------|-------|----------|------------|---------|----------------|
| 1 | Vincent Wijaya | 2023-09-10 | 07:00 | 07:00 |          | Present    | Pending | Approve Reject |

Sumber: Hasil Penelitian, 2024

**Gambar 4. 9 Feature Approval by Manager**

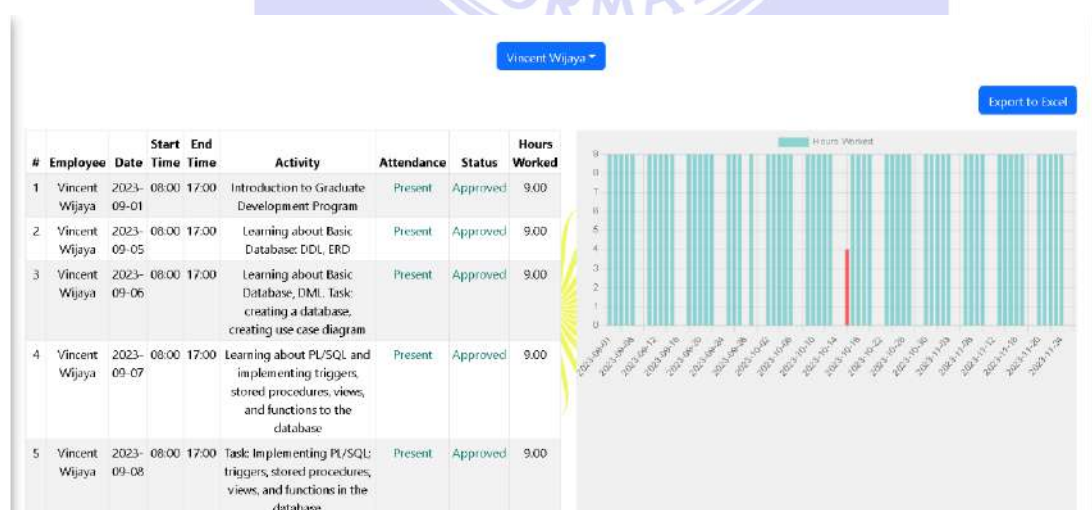
### 4. Fitur Change *Date* (Admin)

| # | Date                 | Detail | Keterangan | Action              |
|---|----------------------|--------|------------|---------------------|
| 1 | Friday, 01-09-2023   |        | Work       | Edit Detail Holiday |
| 2 | Saturday, 02-09-2023 |        | Holiday    | Edit Detail Work    |
| 3 | Sunday, 03-09-2023   |        | Holiday    | Edit Detail Work    |
| 4 | Monday, 04-09-2023   |        | Work       | Edit Detail Holiday |
| 5 | Tuesday, 05-09-2023  |        | Work       | Edit Detail Holiday |

Sumber: Hasil Penelitian, 2024

**Gambar 4. 10 Feature Change Date by Admin**

### 5. Fitur View Employee *Timesheet* by Manager



Sumber: Hasil Penelitian, 2024

**Gambar 4. 11 View employee timesheet by Manager**

## 4.2 Code Generation

### 1. Code Generation Form Login

```
import { useEffect, useState } from "react";
import axios from "axios";
import jwtDecode from "jwt-decode";
import { useNavigate } from "react-router-dom";
import ClipLoader from "react-spinners/ClipLoader";
```

```
function Login() {
  const [data, setData] = useState({
    email: "",
    password: "",
  });
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate ();
  const handleChange = (e) => {
    const { name, value } = e.target;
    setData((prevState) => ({
      ...prevState,
      [name]: value,
    }));
  };
};
```

```
const handleSubmit = () => {
  setLoading(true)
  axios({
    url: "http://localhost:8089/authenticate",
    method: "POST",
```

```
data: data,
headers: {
  "Content-Type": "application/json",
},
})

.then((response) => {
  console.log(response);
  const token = response.data.token;

  // Decode token untuk mendapatkan info pengguna
  const decoded = jwtDecode(response.data.token);
  console.log("Received token:", token);
  console.log("Email:", decoded.sub);
  console.log("Name:", decoded.name);
  console.log("Role:", decoded.role);
  console.log("Id:", decoded.id);
  console.log("Manager:", decoded.manager_id);

  // Simpan data ke localStorage
  localStorage.setItem("Token", token);
  localStorage.setItem("Email", decoded.sub);
  localStorage.setItem("Name", decoded.name);
  localStorage.setItem("Role", decoded.role);
  localStorage.setItem("Id", decoded.id);
  localStorage.setItem("Manager", decoded.manager_id);
  setLoading(false)
  navigate(0)
})

.catch((error) => {
  console.log(error);
```

```

        setLoading(false)
        alert("Username dan Password tidak sesuai")
    });
};

return (
    <div className="container">
        <p style={{ color: "red" }}>
            <b> <ClipLoader
                loading={loading}
                size={150}
                aria-label="Loading Spinner"
                data-testid="loader"
                className="override"
            /></b>
        </p>
        <div className="mb-3">
            <label className="black-text"></label>
            <input
                name="email"
                type="text"
                value={data.email}
                onChange={handleChange}
                className="container"
                style={{
                    maxWidth: "300px",
                    marginLeft: "30px",
                    marginTop: "15px",
                    marginBottom: "-3px"
                }}
                placeholder = "Email"
            />
        </div>
    </div>

```

```

<div className="mb-3">
  <label className="black-text"></label>
  <input
    name="password"
    type="password"
    value={data.password}
    onChange={handleChange}
    className="container"
    style={{ maxWidth: "300px", marginLeft:"30px",  marginBottom:"-5px" }}
    placeholder ="Password"
  />
</div>
<button onClick={handleSubmit} className="btn btn-primary">
  Submit
</button>
</div>

);
}

export default Login;

```



## 2. Code Generation Page Layout

```

import { Link, Outlet, useNavigate } from "react-router-dom";

import axios from "axios";

import { useEffect, useState } from "react";

import Login from "../login";

import Register from "../register";

import { admin, user } from "../lib/navbar";

```

```
import Container from "react-bootstrap/Container";  
  
import Nav from "react-bootstrap/Nav";  
  
import Navbar from "react-bootstrap/Navbar";  
  
import "./index.css";  
  
import { Button } from "react-bootstrap";  
  
import { AiOutlineLogout } from "react-icons/ai";  
  
import React from "react";
```

```
const Layout = () => {  
  const navigate = useNavigate();  
  const [show, setShow] = useState(true);  
  const [data, setData] = useState([  
    {  
      email: localStorage.getItem("Email"),  
      name: localStorage.getItem("Name"),  
      role: localStorage.getItem("Role"),  
    },  
  ]);  
};
```

```
const clearLocal = () => {  
  localStorage.clear();  
};  
  
let x = localStorage.getItem("Role");  
  
{  
  if (x == "admin") {
```

```

return (
  <div>
    <aside style={{ maxWidth: show ? "300px" : "100px" }}>
      <div>
        <Button onClick={() => setShow(!show)} variant="warning"></Button>
        {admin.map((link, index) => (
          <section className="wrap" key={index}>
            <Link to={link.to} className="icon-link">
              <div className="icon" style={{ fontSize: "2em" }}>
                {link.icon && React.createElement(link.icon)}
              </div>
            </Link>
            <Link
              className={`mx-2 link ${show ? "" : "visible"}`}
              to={link.to}
            >
              {link.page}
            </Link>
          </section>
        ))}
      </div>

      <section className="wrap log-out">
        <div className="icon"></div>
        <Link to="/" onClick={clearLocal} className="wrap log-out">

```



```

<div className="icon">

  <AiOutlineLogout style={{ fontSize: "2em"}} />

</div>

<span      className={`sidebar-footer      ${show      ?      ""      :
"visible"}} `>Logout</span>

</Link>

</section>

</aside>

<Outlet />
</div>
);
} else if (x == "user") {
  return (
    <div>
      <aside style={{ maxWidth: show ? "300px" : "100px" }}>
        <div>
          <Button onClick={() => setShow(!show)} variant="warning"></Button>

          {user.map((link, index) => (
            <section className="wrap" key={index}>

              <Link to={link.to} className="icon-link">

                <div className="icon" style={{ fontSize: "2em" }}>

                  {link.icon && React.createElement(link.icon)}

                </div>

              </Link>

```

```

<Link
  className={`mx-2 link ${show ? "" : "visible"}`}
  to={link.to}
>
  {link.page}
</Link>
</section>
)}}
</div>

<section className={`wrap log-out ${show ? 'visible' : ""}`}>
  <div className="icon"></div>
  <Link to="/" onClick={clearLocal} className={`wrap log-
out ${show ? 'visible' : ""}`}>
    <div className="icon">
      <AiOutlineLogout style={{ fontSize: "2em" }} />
    </div>
    <span className={`sidebar-footer ${show ? "" :
"visible"}`}>Logout</span>
  </Link>
</section>
</aside>

<Outlet />
</div>

```

```

    );
  } else {
    return (
      <div style={{ justifyContent: "center" }}>
        <nav>
          <ul>
            <li>
              <Login />
              <Register />
            </li>
          </ul>
        </nav>
      </div>
    );
  }
};

```



```
export default Layout;
```

### 3. Code Generation Input *Timesheet*

```
import React, { useEffect, useState } from "react";
```

```
import axios from "axios";
```

```
import Form from 'react-bootstrap/Form';
```

```
import Modal from 'react-bootstrap/Modal';
```

```

import Button from 'react-bootstrap/Button';

import Swal from "sweetalert2";

import { Navigate, useNavigate } from "react-router-dom";

const TimesheetForm = () => {

  const [dateentitys, setDateentitys] = useState([{}]);

  useEffect (() => {

    axios({

      url: "http://localhost:8089/api/dateentity",

      method: "GET",

      headers: {

        "Content-Type": "application/json",

      }

    }).then((response)=>{

      setDateentitys(response.data.results)

      console.log(dateentitys)

    }).catch((error)=>{

      console.log(error)

```



```

});

},[]);

const [formData, setFormData] = useState({

  employee: {

    id: localStorage.getItem("Id"),

  },

  dateentity: {

    id: null,

  },

  start_time: "",

  end_time: "",

  activity: "",

  attendance: "",

  status: "Pending",

});

```

```

const [show, setShow] = useState(true);

const navigate = useNavigate();

const handleClose = () => setShow(false);

```



```
const handleShow = () => setShow(true);
```

```
const checkData = () =>{
```

```
  console.log(dateentitys)
```

```
}
```

```
let resetForm = () => {
```

```
  setFormData({
```

```
    employee: {
```

```
      id: "",
```

```
    },
```

```
    dateentity: {
```

```
      id: "",
```

```
    },
```

```
    start_time: "",
```

```
    end_time: "",
```

```
    activity: "",
```

```
    attendance: "",
```

```
    status: "Pending",
```

```
  });
```

```
};
```



```
const startTime = (param) => dateentitys.find(x => x.id == param)

const handleSubmit = (e) => {

  e.preventDefault();

  let object = {

    employee: formData.employee,

    dateentity: formData.dateentity,

    start_time: formData.start_time,

    end_time: formData.end_time,

    activity: formData.activity,

    attendance: formData.attendance,

    status: formData.status,

  };

  console.log(object)

  axios({

    url: "http://localhost:8089/api/timesheet",

    method: "POST",

    data: JSON.stringify(object),

    headers: {

      "Content-Type": "application/json",
```

```
    },  
  
  })  
  
  .then((response) => {  
  
    setFormData({  
  
      ...formData,  
  
    });  
  
    console.log(response);  
  
    Swal.fire({  
  
      position: "center",  
  
      icon: "success",  
  
      title: "Timesheet successfully inputted",  
  
      showConfirmButton: false,  
  
      timer: 1500  
    })  
  
    resetForm();  
  
    navigate(0)  
  
  })  
  
  .catch((error) => {  
  
    console.log(error);
```





```
});

console.log(formData);

};

const handleChange = (e) => {

  const { name, value } = e.target;

  if (name === "employee") {

    setFormData({

      ...formData,

      employee: {

        id: value,

      },

    });

  } else if (name === "dateentity") {

    setFormData({

      ...formData,

      dateentity: {

        id: value,

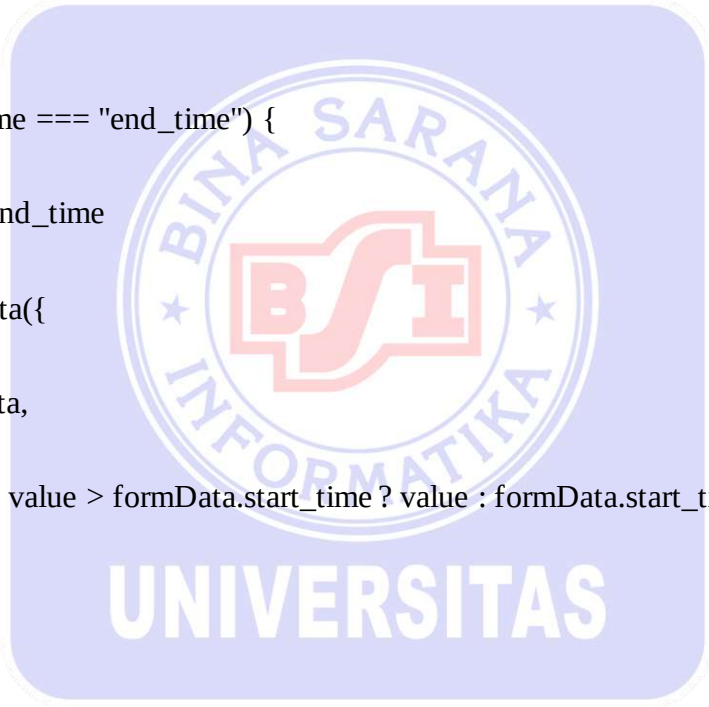
      },

    });

  }

}
```

```
} else if (name === "start_time") {  
  
    // Update start_time  
  
    setFormData({  
  
        ...formData,  
  
        start_time: value,  
  
        end_time: value < formData.end_time ? value : formData.end_time,  
  
    });  
  
} else if (name === "end_time") {  
  
    // Update end_time  
  
    setFormData({  
  
        ...formData,  
  
        end_time: value > formData.start_time ? value : formData.start_time,  
  
    });  
  
} else {  
  
    setFormData({  
  
        ...formData,  
  
        [name]: value,  
  
    });  
  
}
```

The logo of Bina Sarana Informatika Universitas is a circular emblem. The outer ring contains the text "BINA SARANA INFORMATIKA" at the top and "UNIVERSITAS" at the bottom, separated by two stars. In the center of the emblem is a stylized red graphic that combines the letters "B" and "I" with a swoosh, representing the university's initials.

```
};
```

```
return (<>
```

```
  <Button variant="primary" onClick={handleShow} style={{
```

```
    position: 'fixed',
```

```
    top: '25%',
```

```
    left: '50%',
```

```
    transform: 'translate(-20%, -20%)',
```

```
  }}>
```

```
    Input Timesheet
```

```
  </Button>
```

```
  <Modal show={show} onHide={handleClose}>
```

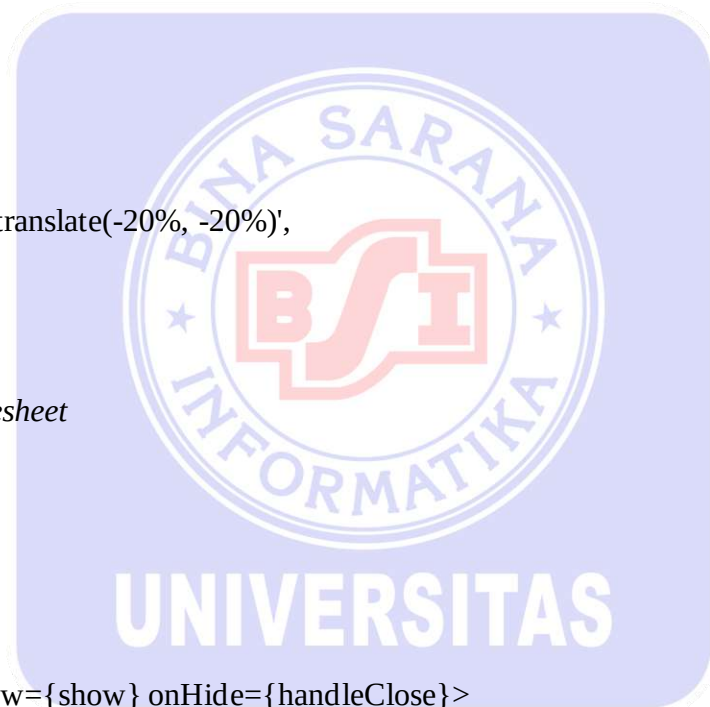
```
    <Modal.Header closeButton>
```

```
      <Modal.Title>Timesheet Form</Modal.Title>
```

```
    </Modal.Header>
```

```
    <Modal.Body>
```

```
      <Form onSubmit={handleSubmit}>
```



```

{ /* Employee */}

<Form.Group className="mb-3 cons">

  <Form.Control

    type="text"

    name="employee"

    value={formData.employee.id}

    onChange={handleChange}

    disabled

    hidden

  ></Form.Control>

</Form.Group>

{ /* Date */}

<Form.Group className="mb-3">

  <Form.Label>Select Date</Form.Label>

  <Form.Select

    name="dateentity"

    value={formData.dateentity.id}

    onChange={handleChange}

```

>

```
<option disabled>Select Date </option>
```

```
{dateentitys.map((x) => (
```

```
<option value={x.id} key={x.id}>
```

```
{x.datetb}
```

```
</option>
```

```
)))
```

```
</Form.Select>
```

```
</Form.Group>
```

```
{/* Start Time */}
```

```
<Form.Group className="mb-3">
```

```
<Form.Label>Start Time</Form.Label>
```

```
<Form.Control
```

```
type="datetime-local"
```

```
name="start_time"
```

```
value={formData.start_time}
```

```
onChange={handleChange}
```

```
placeholder="yyyy-MM-dd(T)HH:mm:ss"
```

```
// required
```

```

/>

</Form.Group>

{/* End Time */}

<Form.Group className="mb-3">

  <Form.Label>End Time</Form.Label>

  <Form.Control

    type="datetime-local"

    name="end_time"

    value={formData.end_time}

    onChange={handleChange}

    placeholder="yyyy-MM-dd(T)HH:mm:ss"

    // required

  />

</Form.Group>

{/* Activity */}

<Form.Group className="mb-3">

  <Form.Label>Activity</Form.Label>

  <Form.Control

    type="text"

```

```

name="activity"

value={formData.activity}

onChange={handleChange}

// required

/>

</Form.Group>

{/* Attendance */}

<Form.Group className="mb-3">

  <Form.Label>Attendance</Form.Label>

  <select

    name="attendance"

    value={formData.attendance}

    onChange={handleChange}

    required

  >

    <option disabled value="">

      Pilih Kehadiran

    </option>

    <option value="Present">Present</option>

```

```

<option value="Absence">Absence</option>

<option value="Sick">Sick</option>

<option value="Leave">Leave</option>

</select>

</Form.Group>

{/* Status */}

<Form.Group className="mb-3">

  <Form.Control

    type="text"

    name="status"

    value={formData.status}

    onChange={handleChange}

    // required

    hidden

  />

</Form.Group>

<Button variant="secondary" onClick={handleClose}>

  Close

</Button>

```



```

<Button type="submit" variant="primary">

  Save Timesheet

</Button>

</Form>

</Modal.Body>

</Modal>

</>

);

};

export default TimesheetForm;

```

### 4.3 Testing

Pengujian digunakan untuk memastikan bahwa perangkat lunak yang dihasilkan telah memenuhi kebutuhan proses bisnis pengguna. Pengujian ini sangat penting untuk mengidentifikasi kesalahan atau kekurangan dalam sistem sebelum diimplementasikan secara penuh. Salah satu metode yang sering digunakan adalah pengujian *black-box testing*. Metode ini berfokus pada pengujian fungsionalitas perangkat lunak tanpa memperhatikan struktur internal atau kode sumber. Penguji hanya perlu memasukkan data input dan memeriksa apakah output yang dihasilkan sesuai dengan yang diharapkan oleh pengguna.

*Black-box testing* bertujuan untuk mencari kesalahan dan memastikan bahwa aplikasi berfungsi dengan benar ketika dioperasikan dengan berbagai data yang dimasukkan. Metode ini efektif untuk mengidentifikasi masalah seperti kesalahan

fungsi, kegagalan integrasi, dan output yang tidak sesuai. Dengan menggunakan black-box testing, pengembang dapat mendemonstrasikan bahwa perangkat lunak dapat menangani data input dengan benar dan menghasilkan output yang sesuai dengan spesifikasi yang diharapkan, sehingga memastikan bahwa perangkat lunak siap untuk digunakan oleh pengguna. Metode ini juga membantu dalam memvalidasi bahwa semua fitur dan fungsi perangkat lunak bekerja sebagaimana mestinya dalam berbagai skenario penggunaan yang berbeda.

#### A. Form Login

##### Hasil Pengujian Black Box Testing Form Login

**Tabel IV. 1 Form Login**

| No | Skenario Pengujian                                                                                                   | Test case                                                    | Hasil Yang Diharapkan                                                                                 | Hasil Pengujian | Kesimpulan |
|----|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-----------------|------------|
| 1  | Mengosongkan semua data <i>login</i> , lalu langsung mengklik tombol " <i>login</i> "                                | <i>Username:</i><br>(kosong)<br><i>Password:</i><br>(kosong) | Sistem menolak akses <i>login</i> dan menampilkan alert " <i>Username atau password anda salah!</i> " | Sesuai Harapan  | Valid      |
| 2  | Mengisi data login dengan <i>username</i> dan <i>password</i> yang salah kemudian tekan tombol " <i>login</i> "      | <i>Username:</i><br>(salah)<br><i>Password:</i><br>(salah)   | Sistem menolak akses <i>login</i> dan menampilkan alert " <i>Username atau password anda salah!</i> " | Sesuai Harapan  | Valid      |
| 3  | Mengisi data login dengan <i>username</i> dan <i>password</i> yang benar, lalu langsung klik tombol " <i>login</i> " | <i>Username:</i><br>(benar)<br><i>Password:</i><br>(benar)   | Sistem akan menerima akses <i>login</i> dan menampilkan menu <i>homepage</i>                          | Sesuai Harapan  | Valid      |

## B. Form Timesheet

### Hasil Pengujian Black Box Testing Form *Timesheet*

*Tabel IV. 2 Form Timesheet*

| No | Skenario Pengujian                                                                                          | Test case                                                   | Hasil Yang Diharapkan                                   | Hasil Pengujian | Kesimpulan |
|----|-------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|---------------------------------------------------------|-----------------|------------|
| 1  | Mengisi data produk dengan salah satu data kosong atau semua kosong, lalu langsung mengklik tombol “Submit” | Data <i>timesheet</i> salah satu atau semua kosong          | Button tidak bisa di klik                               | Sesuai Harapan  | Valid      |
| 2  | Mengisi seluruh data <i>timesheet</i> , lalu langsung mengklik tombol “Submit”                              | Data <i>timesheet</i> terisi semuanya, namun ada yang salah | Button tidak bisa di klik                               | Sesuai Harapan  | Valid      |
| 3  | Mengisi seluruh data <i>timesheet</i> , lalu langsung mengklik tombol “Submit”                              | Data <i>timesheet</i> terisi semuanya dan sesuai            | Akan muncul notif “ <i>Timesheet berhasil diinput</i> ” | Sesuai Harapan  | Valid      |

### C. Form Security

#### Hasil Pengujian Black Box Testing Security

*Tabel IV. 3 Form Security*

| No | Skenario Pengujian                                                                                                    | Test case                                                 | Hasil Yang Diharapkan                                                                    | Hasil Pengujian | Kesimpulan |
|----|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|------------------------------------------------------------------------------------------|-----------------|------------|
| 1  | Mengisi data kriteria laporan dengan salah satu data kosong atau semua kosong, lalu langsung mengklik tombol "Submit" | Data kriteria laporan kosong salah satu atau semua kosong | Sistem akan menolak akses tampil data laporan dan menampilkan pesan "Data masih kosong!" | Sesuai Harapan  | Valid      |
| 2  | Mengisi seluruh kriteria laporan, lalu langsung mengklik tombol "Submit"                                              | Data kriteria laporan terisi semuanya                     | Sistem akan menerima akses tampil laporan                                                | Sesuai Harapan  | Valid      |

#### 4.4 Support

Dalam pengembangan dan implementasi perangkat lunak berbasis web, dukungan (*support*) memainkan peran yang sangat penting untuk memastikan perangkat lunak berfungsi secara optimal dan sesuai dengan kebutuhan pengguna. Dukungan ini mencakup berbagai aspek seperti pemeliharaan sistem, penyelesaian masalah teknis, serta pembaruan berkala untuk meningkatkan kinerja dan keamanan perangkat lunak. Dengan adanya dukungan yang baik, pengguna dapat merasa lebih nyaman dan yakin dalam menggunakan perangkat lunak tersebut, karena mereka tahu bahwa ada tim yang siap membantu jika terjadi masalah atau jika ada kebutuhan untuk peningkatan fitur.

Selain itu, dukungan yang efektif juga melibatkan pelatihan pengguna untuk memastikan mereka dapat memanfaatkan semua fitur yang tersedia dengan maksimal. Ini termasuk menyediakan dokumentasi yang jelas, tutorial, dan layanan bantuan pelanggan yang responsif. Dengan demikian, dukungan yang komprehensif tidak hanya membantu dalam pemecahan masalah tetapi juga berkontribusi pada peningkatan pengalaman pengguna secara keseluruhan. Hal ini penting untuk menjaga kepuasan pengguna dan memastikan perangkat lunak tetap relevan dan berguna dalam jangka panjang. Bagian ini akan membahas dua aspek utama dari dukungan: publikasi perangkat lunak dan spesifikasi *hardware* serta *software* yang dibutuhkan.

#### 4.4.1. Publikasi Software

Publikasi perangkat lunak adalah tahap di mana perangkat lunak yang telah dikembangkan dan diuji dirilis untuk digunakan oleh pengguna akhir. Proses publikasi melibatkan beberapa langkah penting, termasuk:

1. **Distribusi Perangkat Lunak:** Perangkat lunak dapat didistribusikan melalui berbagai saluran, seperti situs *web* resmi, platform distribusi perangkat lunak (seperti: *GitHub*, *GitLab*, atau *SourceForge*), dan melalui media fisik jika diperlukan.
2. **Manajemen Versi:** Penting untuk mengelola versi perangkat lunak dengan baik. Setiap rilis harus diberi nomor versi yang jelas, dan arsip versi sebelumnya harus disimpan untuk referensi dan rollback jika diperlukan.
3. **Komunikasi dengan Pengguna:**
4. **Notifikasi:** Pengguna harus diberitahu tentang rilis baru melalui email, newsletter, atau notifikasi dalam aplikasi. Ini membantu memastikan bahwa pengguna

mengetahui adanya pembaruan dan dapat segera memanfaatkan fitur baru atau perbaikan yang disediakan.

#### 4.4.2. Spesifikasi *Hardware* dan *Software*

Untuk memastikan bahwa perangkat lunak dapat beroperasi dengan baik, perlu ditentukan spesifikasi *hardware* dan *software* yang dibutuhkan. Spesifikasi ini mencakup:

##### Spesifikasi *Hardware* dan *Software*

**Tabel IV. 4 Spesifikasi**

| Kebutuhan                | Keterangan         |
|--------------------------|--------------------|
| <i>Operation Windows</i> | Windows 10         |
| <i>Processor</i>         | Intel i3-1005G1    |
| RAM                      | 4 GB               |
| <i>Harddisk</i>          | 300MB              |
| VGA                      | Intel UHD Graphics |

Dengan menentukan dan mempublikasikan spesifikasi *hardware* dan *software* yang jelas, pengguna dapat mempersiapkan lingkungan yang sesuai untuk menjalankan perangkat lunak dengan optimal, serta mengurangi kemungkinan masalah kompatibilitas dan kinerja.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil penelitian aplikasi *timesheet* berbasis *web* pada perusahaan Sinar Maju Bersama, dapat disimpulkan beberapa poin penting sebagai berikut:

1. Efisiensi dan Efektivitas Pengelolaan Waktu Kerja: Aplikasi *timesheet* berbasis *web* berhasil meningkatkan efisiensi dan efektivitas dalam pengelolaan waktu kerja karyawan. Dengan sistem ini, pencatatan waktu kerja menjadi lebih mudah, cepat dan akurat dibandingkan dengan sistem manual sebelumnya.
2. Peningkatan Produktivitas: Dengan adanya aplikasi *timesheet*, manajer dapat memantau dan mengevaluasi kinerja karyawan secara lebih efektif. Hal ini berdampak positif pada produktivitas karyawan dan perusahaan secara keseluruhan.
3. Kemudahan Akses dan Penggunaan: Aplikasi ini dirancang dengan antarmuka yang responsive dan *user-friendly*, dan sehingga memudahkan pengguna dalam mengakses dan menggunakan aplikasi dari berbagai perangkat dan lokasi.
4. Integrasi Data yang Lebih Baik: Aplikasi *timesheet* ini memungkinkan integrasi data yang lebih baik, membantu dalam analisis dan pembuatan laporan yang akurat dan terpercaya.

## 5.2 Saran

Berdasarkan hasil penelitian dan penelitian pengembangan aplikasi, ini adalah beberapa ide yang dapat dipertimbangkan untuk pengembangan dan pemanfaatan lebih lanjut:

1. Pengembangan Fitur Tambahan: Untuk meningkatkan fungsionalitas aplikasi, disarankan untuk menambahkan fitur-fitur tambahan seperti notifikasi pengingat untuk karyawan, integrasi dengan sistem penggajian, dan fitur analitik untuk membantu manajer dalam melakukan evaluasi kinerja.
2. Pelatihan Pengguna: Untuk memastikan aplikasi dapat digunakan secara optimal, perlu diadakan pelatihan bagi karyawan dan manajer mengenai cara penggunaan aplikasi dan manfaatnya. Hal ini akan membantu dalam mempercepat adaptasi pengguna terhadap sistem baru.
3. Pemeliharaan dan Pembaruan Berkala: Penting untuk melakukan pemeliharaan dan pembaruan aplikasi secara berkala untuk memastikan aplikasi tetap berjalan dengan baik dan aman. Pembaruan juga dapat mencakup peningkatan keamanan dan penambahan fitur baru yang sesuai dengan kebutuhan perusahaan.



## DAFTAR PUSTAKA

- Abdul Wahid Sekolah Tinggi Manajemen Informatika dan Komputer  
Sumedang, A. (2020). *Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi*.  
<https://www.researchgate.net/publication/346397070>
- Aisyah, N., Panjaitan, S., Rambe, M. H., Ahadi, R., Nasution, F., Negeri, U. I., Utara, S., William, J., Ps, I. V, Estate, M., & Serdang, D. (2023). Studi Pustaka: Konsep Bilingualisme dan Pengaruhnya terhadap Perkembangan Bahasa Anak. *Journal on Education*, 05(02), 3788–3795.
- Bhalla, A., Garg, S., & Singh, P. (2020). PRESENT DAY WEB-DEVELOPMENT USING REACTJS. *International Research Journal of Engineering and Technology*. [www.irjet.net](http://www.irjet.net)
- Effendy Erwan. (2023). *Manajemen Database Organisasi Dakwah* (Vol. 5).
- Ernawan, M. E., & Sediyo, E. (2022). Rancangan Sistem Informasi Timesheet Alat Berat Berbasis User Centered Design. *Journal of Information System Research (JOSH)*, 3(4), 557–563.
- Faittullah Akbar, M. (2022). Penerapan Metode Waterfall pada Sistem Informasi Penjualan Dan Persediaan Pada Warung Makan Hejo Karawang. *Journal Computer Science*, 2(1).
- Febriani, E., Aulia Mariska, I., & Farras Nasrida, M. (2023). *Pemanfaatan Media Sosial Bagi Volume Penjualan Produk Skincare Dan Kosmetik Derr Cosmetics*. 3(2).
- Khairani Arifonang, T., Pendidikan Matematika-, P., & Ilmu Tarbiyah dan Keguruan, F. (2022). *PENGENALAN ALGORITMA PADA PEMBELAJARAN PEMROGRAMAN KOMPUTER*.
- Laia, E., Bahasa, G., Di, I., Negeri, S., Susua, S. K., & Selatan, N. (2023). *ANALISIS STRUKTUR TEKS LAPORAN OBSERVASI SISWA KELAS X SMA NEGERI 2 SUSUA TAHUN PELAJARAN 2021/2022*.  
<https://jurnal.uniraya.ac.id/index.php/Kohesi>
- Permata Sari, A. (2020). RANCANG BANGUN SISTEM INFORMASI PENGELOLAAN TALENT FILM BERBASIS APLIKASI WEB. *Jurnal Informatika Terpadu*, 6(1), 29–37.  
<https://journal.nurulfikri.ac.id/index.php/JIT>
- Ritonga, S. (2023). *Tarbiyah bil Qalam, Jurnal Pendidikan, Agama dan Sains REKAPITULASI RATA-RATA DATA HASIL WAWANCARA*

CALON MAHASISWA SEKOLAH TINGGI ILMU TARBIYAH AL  
BUKHARY LABUHANBATU TAHUN 2023.

Saputra, N. I., Agustin, T., & Sugiarto, L. (2024). *Perancangan Sistem CRM Dengan Dukungan Metode UML Berbasis Website Pada PT. Sinergi Informatika Nusantara Utama.*

Sonny, S., & Rizki, S. N. (2021). PENGEMBANGAN SISTEM PRESENSI KARYAWAN DENGAN TEKNOLOGI GPS BERBASIS WEB PADA PT BPR DANA MAKMUR BATAM. In *JURNAL COMASIE* (Vol. 04, Issue 04).

Syabania, R., & Rosmawarni, N. (2021). PERANCANGAN APLIKASI CUSTOMER RELATIONSHIP MANAGEMENT (CRM) PADA PENJUALAN BARANG PRE-ORDER BERBASIS WEBSITE. In *Jurnal Rekayasa Informasi* (Vol. 10, Issue 1).



## DAFTAR RIWAYAT HIDUP

### I. Biodata Mahasiswa

NIM : 17200711

Nama Lengkap : Vincent Wijaya

Tempat/ Tanggal Lahir : Jakarta, 27 Januari 2000

Alamat lengkap : KP.Kabandungan 1 no 18A RT003/005

Kel. Sirnagalih, Kec. Tamansari, Ciapus, Bogor, Jawa Barat 16610

### II. Pendidikan

1.SD BPK Penabur Bogor, lulus tahun 2012

2.SMP Panca Moral Cikampek, lulus tahun 2015

3.SMA Yadika 7 Bogor, lulus tahun 2019



Bogor, 10 Juni 2024



Vincent Wijaya

## SURAT KETERANGAN RISET PERUSAHAAN



**SINAR MAJU  
BERSAMA**  
MEBEL FURNITURE

Kp. Kabandungan 1 No.18 A RT003/005 Kab.  
Bogor, Ciapus 16610

Nomor : No. SK/SMB/240627-32/2024  
Perihal : Surat Keterangan Riset

### SURAT KETERANGAN

Yang bertandatangan dibawah ini :

Nama : Evita Debora  
Jabatan : Manager

Dengan ini menerangkan bahwa, yang tersebut di bawah ini :

Nama : Vincent Wijaya  
NIM : 17200711  
Program : Teknologi Informasi Universitas Bina Sarana Informatika

Adalah benar telah melakukan Riset pada Sinar Maju Bersama terhitung 01 April 2024 sampai dengan 03 Juni 2024, dan yang bersangkutan telah melaksanakan tugasnya dengan baik dan penuh tanggung jawab.

Demikian surat keterangan ini dibuat dengan benar, untuk dapat dipergunakan sebagaimana mestinya.

Bogor, 03 Juni 2024  
Sinar Maju Bersama

Evita Debora  
Manager

## BUKTI HASIL PENGECEKAN PLAGIARISME

### Turnitin

#### ORIGINALITY REPORT

|                  |                  |              |                |
|------------------|------------------|--------------|----------------|
| <b>14%</b>       | <b>12%</b>       | <b>7%</b>    | <b>6%</b>      |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

#### PRIMARY SOURCES

|          |                                                            |               |
|----------|------------------------------------------------------------|---------------|
| <b>1</b> | <b>repository.bsi.ac.id</b><br>Internet Source             | <b>2%</b>     |
| <b>2</b> | <b>prin.or.id</b><br>Internet Source                       | <b>1%</b>     |
| <b>3</b> | <b>123dok.com</b><br>Internet Source                       | <b>1%</b>     |
| <b>4</b> | <b>eprints.ums.ac.id</b><br>Internet Source                | <b>1%</b>     |
| <b>5</b> | <b>elib.pnc.ac.id</b><br>Internet Source                   | <b>1%</b>     |
| <b>6</b> | <b>conference.upgris.ac.id</b><br>Internet Source          | <b>1%</b>     |
| <b>7</b> | <b>docplayer.info</b><br>Internet Source                   | <b>1%</b>     |
| <b>8</b> | <b>repository.mercubuana.ac.id</b><br>Internet Source      | <b>&lt;1%</b> |
| <b>9</b> | <b>Submitted to Universitas Brawijaya</b><br>Student Paper | <b>&lt;1%</b> |

## LAMPIRAN

|                     |            |                     |                                                     |   |   |
|---------------------|------------|---------------------|-----------------------------------------------------|---|---|
| Monday 25-Sep-23    | SINAR MAJU | Product Development | Standby jika ada keluhan                            | 8 | P |
| Tuesday 26-Sep-23   | SINAR MAJU | Product Development | Standby jika ada keluhan                            | 8 | P |
| Wednesday 27-Sep-23 | SINAR MAJU | Product Development | Standby jika ada keluhan                            | 8 | P |
| Friday 29-Sep-23    | SINAR MAJU | Product Development | Mengirim Kasur ke toko Bu Martha(Bogor, Budi Agung) | 8 | P |
| Monday 02-Oct-23    | SINAR MAJU | Product Development | Mengambil Kasur di Cibuluh untuk claim garansi      | 8 | P |
| Tuesday 03-Oct-23   | SINAR MAJU | Product Development | Memperbaiki kasur(garansi)                          | 8 | P |
| Wednesday 04-Oct-23 | SINAR MAJU | Product Development | Mengembalikan kasur ke Cibuluh                      | 8 | P |
| Thursday 05-Oct-23  | SINAR MAJU | Product Development | Membuat Springbed                                   | 8 | P |
| Friday 06-Oct-23    | SINAR MAJU | Product Development | Membuat Springbed                                   | 8 | P |
| Monday 09-Oct-23    | SINAR MAJU | Product Development | Membersihkan Kasur-Kasur yang ada digudang          | 8 | P |
| Tuesday 10-Oct-23   | SINAR MAJU | Product Development | Mengirim Sofa duduk ke Dramaga                      | 8 | P |
| Wednesday 11-Oct-23 | SINAR MAJU | Product Development | Membuat lemari                                      | 8 | P |
| Thursday 12-Oct-23  | SINAR MAJU | Product Development | Membuat lemari                                      | 8 | P |
| Friday 13-Oct-23    | SINAR MAJU | Product Development | Memasang Lemari di kantor                           | 8 | P |

### Lampiran 1 Data Absensi Karyawan



### Lampiran 2 Data Grafik Karyawan