

SKRIPSI

**RANCANG BANGUN SISTEM INFORMASI ABSENSI KARYAWAN
OUTSOURCING BERBASIS *WEB* STUDI KASUS PT. MAYORA INDAH
TBK. PLANT JAYANTI 1**

*Disusun Sebagai Syarat untuk kelulusan
Pada Program Studi Strata 1 Teknik Informatika*

Oleh :

Aldi Prayoga

17.55.201.078



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH TANGERANG**

2021

ABSTRAK

PT. Mayora Indah Tbk plant Jayanti 1 adalah salah satu pabrik yang tergabung dalam PT. MAYORA GROUP yang bergerak di bidang makanan (biskuit). Pabrik ini memiliki total karyawan yang mencapai angka kurang lebih 3000 karyawan. Dari banyak nya jumlah karyawan yang dimiliki, maka bnyak pula terjadi nya masalah dalam bidang absensi karyawan tersebut, baik dalam keterlambatan, izin, sakit, mangkir/alpa, dan banyak lagi lain nya. Kasus yang paling banyak terjadi adalah masalah keterlambatan pengurusan surat izin karyawan yang tidak masuk karna suatu hal tertentu (sakit, keluarga meninggal, dan lain-lain) yang mengakibatkan rusak nya nilai KPI dari bagian Dept. IR &GA. Hal ini terjadi karna kurang terfasilitasi nya sarana untuk karyawan yang berhalangan tersebut memberikan laporan kepada pihak *payroll* PT. Mayora Indah Plant Jayanti 1.

Metode penelitian yang digunakan yaitu metode PIECES dan tahap perancangan sistem menggunakan *Unified Modeling Language* (UML). Tujuan diadakan nya penelitian ini adalah untuk memudahkan karyawan memberikan informasi dan juga memudahkan tim payroll membuat report KPI mingguan sehingga tidak adanya lagi *cash* rusak nya nilai KPI Dept. IR & GA.

Kata Kunci: Absensi, *Payroll*, KPI, IR & GA, Laporan

ABSTRACT

PT. Mayora Indah Tbk Jayanti plant is one of the factories incorporated in PT. MAYORA GROUP which is engaged in food (biscuits). This factory has a total employee number of approximately 3000 employees. From the large number of employees they have, there are also many problems in the field of employee absences, both in delays, permits, illness, loss to follow up, and many others. The most common case is the problem of delay in obtaining employee licenses that do not enter because of a certain matter (sick, family died, etc.) which results in damage to the KPI value from the Dept. IR & GA. This happens because the facilities are not facilitated for employees who are unable to provide a report to the payroll PT. Mayora Indah Plant Jayanti.

The research method used is the PIECES method and the system planning stage using the Unified Modeling Language (UML). The purpose of this research is to make it easier for employees to provide information and also make it easy for the payroll team to make a weekly KPI report so that there is no more cash damaged in the value of KPI Dept. IR & GA

Keywords: Attendance, Payroll, KPI, IR &GA, Report

KATA PENGANTAR

Alhamdulillahirabil'alamin, Puji dan Syukur penulis panjatkan kehadirat Allah SWT, penulis telah menyelesaikan proses penulisan laporan tugas akhir skripsi yang berjudul "RANCANG BANGUN SISTEM INFORMASI ABSENSI KARYAWAN OUTSOURCING BERBASIS *WEB* STUDI KASUS PT. MAYORA INDAH TBK. PLANT JAYANTI 1".

Dalam penyusunan laporan Skripsi ini penulis melibatkan berbagai pihak, untuk itu tidak lupa terima kasih penyusun sampaikan kepada Bapak dan Ibu tercinta yang telah memberikan banyak bantuan baik berupa dukungan, semangat, doa, dan banyak lainnya, penulis juga ingin mengucapkan terima kasih kepada :

1. Allah SWT atas nikmat iman islam dan atas kelancaran dan kesehatan sehingga saya dapat menyelesaikan Laporan tugas akhir Skripsi ini.
2. Kedua orang tua saya , yang telah mendidik dan membesarkan saya dengan penuh kasih sayang, selalu memberikan dukungan moril maupun materil, dan doa yang tiada hentinya. Juga untuk adik-adik tersayang terima kasih untuk semua bentuk doa dan dukungannya.
3. Bapak Dr.H. Ahmad Amarullah, M.Pd selaku Rektor Universitas Muhammadiyah Tangerang.
4. Bapak Ir. Saiful Haq, MSi selaku dekan Fakultas Teknik, Universitas Muhammadiyah Tangerang.
5. Bapak Rohmat Taufik, S.T. M.Kom selaku Wakil Dekan 1 Fakultas Teknik, Universitas Muhammadiyah Tangerang.

6. Bapak Drs. H. Syamsul Bahri, M.Si selaku Wakil Dekan 2 Fakultas Teknik, Universitas Muhammadiyah Tangerang
7. Bapak Syepri Maulana Husain S.Kom, MTI selaku Kepala Prodi Fakultas Teknik Informatika, Universitas Muhammadiyah Tangerang.
8. Ibu Yani Sugiyani, MM., M.kom selaku Dosen Pembimbing yang telah memberikan banyak arahan dan masukan dalam proses penyelesaian laporan kerja praktek ini
9. Bapak dan Ibu dosen Teknik Informatika Universitas Muhammadiyah Tangerang yang telah memberi memberikan bantuan dan ilmu nya.
10. Kepada Bapak Yasir Nasution selaku Dept. Head IRGA PT. Mayora Indah Tbk. Plant Jayanti1 yang telah memberikan saya kesempatan untuk bisa melakukan proses penelitian di PT. Mayora Indah TBK plant Jayanti 1.
11. Kepada teman-teman seperjuangan yang sudah banyak memberikan bantuan dalam banyak hal.

Tangerang, Oktober 2021

Aldi Prayoga

DAFTAR ISI

| | |
|---|------|
| ABSTRAK..... | i |
| ABSTRACT | ii |
| KATA PENGANTAR | iii |
| DAFTAR ISI | v |
| DAFTAR GAMBAR | viii |
| DAFTAR TABEL | ix |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang Masalah..... | 1 |
| 1.2 Identifikasi Masalah | 3 |
| 1.3 Rumusan Masalah..... | 4 |
| 1.4 Batasan Masalah | 4 |
| 1.5 Tujuan Dan Manfaat Penelitian | 4 |
| 1.6 Manfaat Penelitian..... | 5 |
| 1.6.1 Bagi Mahasiswa:..... | 5 |
| 1.6.2 Bagi pihak PT Mayora Indah Tbk.Plant Jayanti 1: | 5 |
| 1.6.3 Bagi Universitas Muhammadiyah Tangerang: | 5 |
| 1.7 Sistematika Penulisan | 6 |
| 1.8 Rencana Kegiatan | 7 |
| BAB II LANDASAN TEORI..... | 8 |
| 2.1 LANDASAN TEORI | 8 |
| 2.1.1 Definisi Informasi..... | 8 |
| 2.1.2 Definisi Sistem..... | 8 |
| 2.1.3 Definisi Sistem Informasi | 10 |
| 2.1.4 Definisi Perancangan..... | 11 |
| 2.1.5 UML | 15 |
| 2.2. TEORI KHUSUS..... | 22 |
| 2.2.1 Definisi Analisa Sistem PIECES..... | 22 |
| 2.2.2 Definisi Pengembangan Sistem Waterfall/ <i>Prototyping/Agile</i> | 24 |
| 2.2.3 Definisi Aplikasi Web Based / Android | 34 |

| | | |
|---|--|-----------|
| 2.2.4 | Definisi Aplikasi Aplikasi MySQL | 35 |
| 2.2.5 | Definisi PHP | 36 |
| 2.2.6 | Definisi XAMPP | 37 |
| 2.2.7 | Definisi Laravel | 38 |
| 2.2.8 | Definisi Bootstrap..... | 38 |
| 2.2.9 | Definisi ERD (<i>Entity Relationship Diagram</i>)..... | 39 |
| 2.2.10 | Definisi Pengujian Sistem <i>Black Box</i> | 40 |
| 2.2.11 | Definisi Problem Shuting..... | 41 |
| 2.3 | <i>Literatur Review</i> | 41 |
| BAB III GAMBARAN UMUM PENELITIAN | | 35 |
| 3.1 | Jenis Penelitian..... | 35 |
| 3.2 | Metode Pemilihan Sampel | 35 |
| 3.3 | Metode PengumpulanData | 36 |
| 3.4 | Metode Analisa data | 38 |
| 3.5 | Metode Pengujian | 40 |
| 3.6 | Tinjauan Tempat Penelitian | 40 |
| 3.6.1 | Sejarah Perusahaan..... | 40 |
| 3.6.2 | Produk Perusahaan | 43 |
| 3.6.3 | Visi dan Misi Perusahaan..... | 43 |
| 3.6.4 | Skema Struktur Organisasi Perusahaan | 44 |
| 3.6.5 | Tugas Dan Tanggung Jawab..... | 45 |
| BAB IV ANALISIS DAN PEMBAHASAN | | 48 |
| 4.1 | Analisa Sistem Yang Sedang Berjalan | 48 |
| 4.1.1 | Proses Absen Karyawan..... | 48 |
| 4.1.2 | Proses Perizinan karyawan | 50 |
| 4.1.3 | Proses Rekap Absensi Oleh Tim <i>Payroll</i> | 52 |
| 4.1.4 | Kelebihan dan kekurangan <i>system</i> yang berjalan..... | 53 |
| 4.2 | <i>Use Case</i> Sistem yang Sedang Berjalan..... | 54 |
| 4.3 | Desain Sistem UML (<i>Unified Modeliing Language</i>) yang Diusulkan... 56 | |
| 4.3.1 | <i>Use Case</i> yang diusulkan | 57 |
| 4.3.2 | Tabel deskripsi penjelasan <i>use case</i> yang diusulkan..... | 58 |
| 4.3.3 | Activity diagram system yang diusulkan | 62 |

| | | |
|----------------------------------|---|----|
| 4.3.4 | <i>Sequence diagram system</i> yang diusulkan | 68 |
| 4.3.5 | <i>Class diagram system</i> yang diusulkan | 73 |
| 4.4 | ERD (Entity Relationship Diagram)..... | 74 |
| 4.5 | Rancangan Tabel..... | 75 |
| 4.6 | Perancangan Antar Muka | 76 |
| 4.7 | Implementasi Antar Muka dan Pemrograman | 80 |
| 4.8 | Pengujian Sistem Black Box Testing | 82 |
| BAB V KESIMPULAN DAN SARAN | | 80 |
| 5.1 | Kesimpulan | 80 |
| 5.2 | Saran | 81 |
| DAFTAR PUSTAKA | | 82 |
| LAMPIRAN | | 83 |

DAFTAR GAMBAR

| | | |
|-------------|--|----|
| Gambar 3.1 | Mayora Head Office | 40 |
| Gambar 3.2 | Struktur umum PT Mayora Indah Tbk plant Jayanti 1 | 44 |
| Gambar 3.3 | Struktur <i>Departemen IR-GA</i> | 45 |
| Gambar 4.1 | <i>Use case</i> yang sedang berjalan | 55 |
| Gambar 4.2 | <i>Usecase</i> system yang diusulkan..... | 58 |
| Gambar 4.3 | Activity diagram karyawan masuk system izin | 63 |
| Gambar 4.4 | Activity diagram <i>login Section Head</i> | 63 |
| Gambar 4.5 | Activity diagram <i>login Payroll</i> | 64 |
| Gambar 4.6 | Activity diagram <i>input</i> izin karyawan | 65 |
| Gambar 4.7 | Activity diagram <i>approve</i> izin karyawan oleh <i>Section Head</i> | 66 |
| Gambar 4.8 | Activity diagram <i>approve</i> izin karyawan oleh <i>Payroll</i> | 67 |
| Gambar 4.9 | Activity diagram <i>input</i> izin karyawan di <i>puzzle</i> | 68 |
| Gambar 4.10 | <i>Sequense</i> diagram <i>login</i> | 69 |
| Gambar 4.11 | <i>Sequense</i> diagram <i>input</i> surat izin karyawan | 70 |
| Gambar 4.12 | <i>Sequense</i> diagram <i>approve</i> izin oleh <i>Section Head</i> | 71 |
| Gambar 4.13 | <i>Sequense</i> diagram <i>approve</i> izin oleh <i>Payroll</i> | 72 |
| Gambar 4.14 | <i>Sequense</i> diagram <i>input</i> izin oleh <i>Payroll</i> di <i>puzzle</i> | 73 |
| Gambar 4.15 | <i>Class</i> diagram yang di usulkan | 74 |
| Gambar 4.16 | <i>ERD (Entity Relationship Diagram)</i> | 75 |
| Gambar 4.17 | Tampilan Menu <i>Login</i> | 77 |
| Gambar 4.18 | Tampilan <i>Input</i> Surat izin karyawan | 78 |
| Gambar 4.19 | Halaman <i>Approve</i> oleh <i>Section Head</i> | 79 |
| Gambar 4.20 | Halaman <i>Approve</i> oleh <i>Payroll</i> | 79 |
| Gambar 4.21 | Halaman <i>Login</i> | 80 |
| Gambar 4.22 | Halamann Utama | 80 |
| Gambar 4.23 | Halaman <i>Input</i> Izin | 81 |
| Gambar 4.24 | Halaman <i>Approve Section Head</i> | 81 |
| Gambar 4.25 | Halaman <i>Approve Payroll</i> | 81 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 1.1 Rencana Kegiatan..... | 7 |
| Tabel 2.1 Smbol <i>Use Case Diagram</i> | 16 |
| Tabel 2.2 Simbol <i>Activity Diagram</i> | 18 |
| Tabel 2.3 Simbol <i>Squence Diagram</i> | 18 |
| Tabel 2.4 Simbol <i>Class Diagram</i> | 20 |
| Tabel 4.1 Daftar <i>Use Case</i> Sistem Berjalan | 55 |
| Tabel 4.2 Penjelasan <i>use case diagram Login</i> karyawan | 58 |
| Tabel 4.3 Penjelasan <i>use case</i> diagram melaporkan surat izin..... | 59 |
| Tabel 4.4 Penjelasan <i>use case</i> diagram Login Section Head..... | 59 |
| Tabel 4.5 Penjelasan <i>use case</i> diagram Approve Section Head | 60 |
| Tabel 4.6 Penjelasan <i>use case</i> diagram Login payroll | 61 |
| Tabel 4.7 Penjelasan <i>use case</i> diagram menerima surat izin..... | 61 |
| Tabel 4.8 Penjelasan Tabel Izin Karyawan | 75 |
| Tabel 4.9 Penjelasan Tabel <i>Approve Section Head</i> | 75 |
| Tabel 4.10 Penjelasan Tabel <i>Approve Payroll</i> | 76 |
| Tabel 4.11 Tampilan Black Box Testing menu <i>login</i> Karyawan | 82 |
| Tabel 4.12 Tampilan Black Box Testing menu <i>login Section Head</i> | 83 |
| Tabel 4.13 Tampilan Black Box Testing menu <i>login Payroll</i> | 83 |
| Tabel 4.14 Tampilan Black Box Testing menu <i>input Izin</i> | 84 |
| Tabel 4.15 Tampilan Black Box Testing menu <i>Approve Section Head</i> | 84 |
| Tabel 4.16 Tampilan Black Box Testing menu <i>Approve Payroll</i> | 85 |

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kemajuan ilmu pengetahuan dan teknologi yang sangat pesat semakin mendorong manusia untuk meningkatkan bahkan menciptakan suatu teknologi baru yang bermanfaat bagi manusia. Teknologi yang berkembang sesuai dengan tuntutan zaman diharapkan menjadi sarana penunjang untuk menangani permasalahan yang timbul di perusahaan atau organisasi. Salah satu bukti kemajuan teknologi adalah komputer, Perusahaan ataupun instansi yang ingin maju dan sukses harus mengikuti sistem yang terkomputerisasi. Penggunaan komputer faktanya dapat mengefisiensikan di dalam berbagai aspek kehidupan manusia.

Seiring dengan perkembangan dunia usaha, komputer memiliki peranan penting dalam mempermudah penyelesaian suatu pekerjaan dan membantu mengembangkan sistem pelayanan yang maksimal guna menunjang kinerja perusahaan. Cara untuk meningkatkan kinerja perusahaan salah satunya dengan membangun suatu sistem informasi. Sistem informasi bagi perusahaan dapat meningkatkan produktifitas dan memperlancar aktifitas perusahaan, terutama dalam memperoleh informasi, mengolah dan menggunakan informasi tersebut. Sistem informasi juga memberikan dukungan bagi perkembangan sistem terkomputerisasi , Pengolahan data secara terkomputerisasi dapat mengolah data

lebih cepat dan akurat, selain itu juga dapat mengolah data dalam skala yang besar.

PT. Mayora Indah Tbk plant Jayanti 1 adalah salah satu perusahaan yang bergerak dibidang biskuit yang di produksi oleh PT. Mayora Group, dimana pada divisi *Payroll* pada departemen IRGA di perusahaan ini dihadapkan dengan proses administrasi perizinan bagi karyawan *outsourcing* yang masih terdapat beberapa kendala yang menyebabkan beberapa proses tidak berjalan secara efisien. Khusus nya pada proses pelaporan surat izin karyawan *outsourcing* yang seharusnya dapat berjalan dengan cepat dan tepat waktu baik dari proses karyawan memberikan surat izin ke tim *payroll*, proses administrasi tim *Payroll* dalam menginput surat izin, dan proses tim payroll dalam membuat laporan mingguan mengenai absensi karyawan. Karena sampai saat ini masih sering terjadinya telat input surat izin dari karyawan *outsourcing* yang di akibatkan karena beberapa factor seperti kurang nya akses komunikasi antara karyawan dengan tim *payroll*, adanya pembagian *shift* kerja yang menyebabkan tidak semua karyawan bisa mengurus surat izin tepat waktu karena jam *operasional* dari tim *payroll* adalah dari pukul 08:00 WIB sampai dengan 16:00 WIB sedangkan jam kerja karyawan terbagi menjadi 3 (tiga) *shift* yaitu *shift* 1 (07:00-15:00 WIB), *shift* 2 (15:00-23:00 WIB), *shift* 3 (23:00-07:00 WIB). Selain itu dari jumlah karyawan yang saat ini mencapai kurang lebih 3000 orang, karyawan *outsourcing* mencapai di angka tertinggi yaitu 2550 karyawan. Yang dimana dengan jumlah yang banyak ini, tentu nya sangat berpengaruh apabila banyak nya karyawan yang tidak hadir sesuai dengan jadwal bekerjanya yang bisa menghambat alur dari

produksi dan juga bagi yang tidak mengurus ketidakhadiran mereka yang menyebabkan status karyawan banyak yang menjadi **Alpa/mangkir**.

Untuk itu maka di perlukan suatu sistem yang lebih efisien untuk proses admintrasi perizinan karyawan yang dimana akan berdampak kepada buruk nya nilai KPI (*Key Performa indeks*) dari Departemen IRGA yang banyak mengakibatkan kerugian karena menyangkut performa kerja tim payroll dan juga *achievement* yang akan mengalami penurunan jika terus memakai *system* yang kurang efisiensi seperti saat ini. Selain itu, *system* yang akan diusulkan diharapkan juga menjadi tolak ukur dari performa kerja karyawan yang bisa lebih di pantau oleh atasan atau *Section head* masing-masing. Berdasarkan uraian diatas maka dilakukan penelitian sistem tentang informasi absensi karyawan outsourcing PT. Mayora Indah Tbk Jayanti 1, sehubungan dengan itu maka penelitian ini akan mengambil judul **“RANCANG BANGUN SISTEM INFORMASI ABSENSI KARYAWAN OUTSOURCING BERBASIS WEB STUDI KASUS PT. MAYORA INDAH TBK. PLANT JAYANTI 1”**

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, adapun masalah yang teridentifikasi adalah:

- a. Terjadinya telat input surat izin karyawan *outsourcing*
- b. Lambatnya proses adminitrasi dalam hal perizinan karyawan outsourcing
- c. Kurangnya komunikasi antara karyawan outsourcing dengan tim payroll.
- d. Rusaknya nilai KPI tim Payroll dari Dept. IR & GA dalam bidang absensi

1.3 Rumusan Masalah

Berdasarkan uraian identifikasi masalah diatas, maka penulis memiliki rumusan masalah sebagai berikut:

- a. Bagaimana sistem yang berjalan pada sistem perizinan karyawan *outsourcing* PT. Mayora Indah Tbk. Plant Jayanti 1?
- b. Bagaimana Menganalisis sistem sistem perizinan karyawan *outsourcing* PT. Mayora Indah Tbk. Plant Jayanti 1 menggunakan metode *UML(Unified Modeling Language)*?
- c. Bagaimana menganalisa proses pelaporan izin karyawan *outsourcing* PT. Mayora Indah Tbk. Plant Jayanti 1?

1.4 Batasan Masalah

Dalam penelitian ini, maka penulis membatasi penulisan Kerja Praktek sebagai berikut:

- a. Penelitian ini dilakukan pada divisi *Payroll* PT.Mayora indah Tbk Jayanti 1
- b. Penelitian ini menganalisis sistem absensi dan perizinan karyawan *outsourcing* PT. Mayora Indah Tbk Jayanti 1.
- c. Penggambaran sistem menggunakan *UML(Unified Modeling Language)*
- d. Penelitian dilakukan di PT. Mayora Indah Tbk Jayanti 1 yang berada di Jl. Raya Serang KM. 31 Desa Sumur Bandung, Kec. Jayanti, Kab. Tangerang
- e. Penelitian dilakukan pada tanggal 01 jul 2021 sampai dengan 30 Agustus 2021

1.5 Tujuan Dan Manfaat Penelitian

Adapun tujuan dari penelitian ini adalah :

- a. Sebagai syarat untuk memenuhi kelulusan tugas akhir.
- b. Mempermudah proses perizinan bagi karyawan outsourcing dan mempercepat pembuatan laporan KPI Dept. IRGA PT Mayora Indah Tbk. Plant Jayanti 1.
- c. Menganalisis rancangan sistem perizinan baru pada PT Mayora Indah Tbk. Plant Jayanti 1 dengan metode *UML (Unified Modeling Language)*

1.6 Manfaat Penelitian

1.6.1 Bagi Mahasiswa:

Adapun manfaat penelitian ini bagi mahasiswa adalah:

- a) Menambah wawasan tentang apa saja di dapat dari kegiatan perusahaan / dunia kerja
- b) Dapat memahami masalah-masalah yang ada dalam proses perizinan karyawan outsourcing PT. Mayora Indah Tbk. Plant Jayanti 1
- c) Dapat memahami proses kerja dari suatu sistem yang sedang berjalan.
- d) Sebagai acuan untuk menyusun tugas akhir.

1.6.2 Bagi pihak PT Mayora Indah Tbk.Plant Jayanti 1:

Adapun manfaat penelitian ini bagi perusahaan adalah:

- a) Dapat mengetahui kelemahan sistem yang sedang berjalan.
- b) Sebagai bahan analisa pada sistem yang berjalan

1.6.3 Bagi Universitas Muhammadiyah Tangerang:

Adapun manfaat penelitian ini bagi Universitas adalah:

- a) Dapat di manfaatkan sebagai bahan evaluasi untuk mengukur tingkat keberhasilan dari lembaga atau instansi perusahaan.
- b) Kampus dapat mengukur kemampuan mahasiswa dalam menyelesaikan tugas akhir.
- c) Memperoleh berbagai kasus yang berharga yang dapat digunakan sebagai contoh dalam memberikan materi perkuliahan dan menemukan berbagai permasalahan untuk pengembangan penelitian.
- d) Kampus dapat menyalurkan mahasiswanya ke berbagai lembaga hukum, teknologi, pendidikan, bisnis, perusahaan dan akan membuat kampus lebih dikenal.

1.7 Sistematika Penulisan

Pembahasan tugas akhir (Skripsi) dibagi kedalam bab per bab untuk memudahkan didalam pembahasan sistem. Tiap-tiap bab masih merupakan kesatuan dengan beberapa perincian sebagai berikut:

BAB I: PENDAHULUAN

Pada BAB I ini akan di bahas tentang Latar Belakang Masalah, Identifikasi Masalah, Batasan Masalah, Tujuan Dan Manfaat Penelitian, Sistematika penulisan dan rencana kegiatan.

BAB II: TINJAUAN PUSTAKA

Pada BAB II ini membahas tentang teori-teori yang digunakan sebagai referensi, rujukan serta pendukung dalam penyusunan laporan tugas akhir

BAB III: METODOLOGI PENELITIAN

Pada BAB III ini membahas tentang metode-metode penelitian yang dilakukan dan sejarah, tujuan, visi dan misi, dan struktur organisasi pada PT. Mayora Indah Tbk. Plant Jayanti 1

BAB IV: ANALISIS DAN PEMBAHASAN

Pada BAB IV ini menjelaskan dan menganalisa sistem yang berjalan, kelebihan dan kekurangan, penggambaran sistem menggunakan (*Unified Modeling Language*) UML, *software* dan *hardware* yang di gunakan dan menjelaskan tentang sistem yang akan di usulkan

BAB V: KESIMPULAN DAN SARAN

Pada BAB V ini kesimpulan dalam melakukan analisa selama kerja praktek di PT. Mayora Indah Tbk. Plant Jayanti 1 dan juga saran dan masukan dalam membantu masalah yang ada di proses perizinan karyawan

1.8 Rencana Kegiatan

Adapun rencana kegiatan pelaksanaan tugas akhir (skripsi) di gambarkan pada tabel 1.1 :

Tabel 1.1 Rencana Kegiatan

BAB II

LANDASAN TEORI

2.1 LANDASAN TEORI

2.1.1 Definisi Informasi

Informasi adalah data yang diolah menjadi bentuk yang berguna untuk membuat keputusan karena informasi menurunkan ketidakpastian/ meningkatkan pengetahuan (Rusdiana & Irfan, 2014).

Menurut McLeod yang dikutip oleh Yakub (2012), Informasi merupakan sebuah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya

Dari dua definisi di atas, penulis dapat menyimpulkan bahwa informasi adalah sebuah bentuk data yang kompleks yang digunakan sebagai media pemberitahuan kepada para pembaca, pendengar, pengamat, dll.

2.1.2 Definisi Sistem

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa

elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara di mana yang berperan sebagai penggerakya yaitu rakyat yang berada dinegara tersebut.

Kata "sistem" banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka.

Berikut adalah definisi sistem menurut para ahli :

a. Menurut Mustakini (2009)

Secara etimologi sistem berasal dari kata sistem yang berarti susunan atau cara. Sistem dapat di kelompokkan kedalam dua pendekatan yaitu sistem yang di letakkan pada pendekatan prosedur dan sistem yang di letakkan pada komponen atau elemen.

b. Menurut Azhar Susanto (2013)

Sistem adalah kumpulan atau *group* dari sub sistem atau bagian maupun komponen apapun baik fisik ataupun *non* fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu.

c. Menurut Romney dan Steinbart (2015)

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian

besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

2.1.3 Definisi Sistem Informasi

Sistem informasi adalah kumpulan informasi di dalam sebuah basis data menggunakan model dan media teknologi informasi digunakan di dalam pengambilan keputusan bisnis sebuah organisasi. Di dalam suatu organisasi, informasi merupakan sesuatu yang penting di dalam mendukung proses pengambilan keputusan oleh pihak manajemen. Sistem ini memanfaatkan perangkat keras dan perangkat lunak komputer, prosedur manual, model manajemen dan basis data.

Dari definisi di atas terdapat beberapa kata kunci:

1. Berbasis komputer dan sistem manusia dan mesin

Berbasis komputer: perancang harus memahami pengetahuan komputer dan pemrosesan informasi.

Sistem manusia mesin: ada interaksi antara manusia sebagai pengelola dan mesin sebagai alat untuk memroses informasi. Ada proses manual yang harus dilakukan manusia dan ada proses yang terotomasi oleh mesin. Oleh karena itu diperlukan suatu prosedur/manual sistem.

2. Sistem basis data terintegrasi

Adanya penggunaan basis data secara bersama-sama (sharing) dalam sebuah data base manajemen system.

3. Mendukung operasi

Informasi yang diolah dan di hasilkan digunakan untuk mendukung operasi organisasi.

Sistem informasi memiliki komponen berupa subsistem yang merupakan elemen elemen yang lebih kecil yang membentuk sistem informasi tersebut misalnya bagian input, proses, output. Tanpa ketiga itu sistem informasi tidak dapat berjalan dengan baik.

- *Input*: sekumpulan data yang akan kita olah menjadi sebuah informasi yang nantinya akan kita sajikan bagi masyarakat.
- *Proses*: suatu kegiatan dimana kita mengolah seluruh data yang ada untuk menghasilkan suatu informasi .
- *Output*: informasi-informasi yang dapat dengan mudah di peroleh, di mengerti dan dimanfaatkan oleh masyarakat.

2.1.4 Definisi Perancangan

Perancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru, jika sistem itu berbasis komputer, maka rancangan dapat menyertakan spesifikasi jenis peralatan yang akan digunakan. (Zakiyudin, 2013). Perancangan sistem menurut penulis adalah langkah atau tindak lanjut dari sebuah rencana guna menentukan suatu proses atau data yang dibutuhkan oleh sistem.

Langkah-langkah tahapan perancangan yaitu:

1. Menyiapkan rancangan sistem yang terperinci
2. Mengidentifikasi berbagai alternatif konfigurasi system
3. Mengevaluasi berbagai alternatif konfigurasi system
4. Memilih konfigurasi terbaik
5. Menyiapkan usulan penerapan
6. Menyetujui atau menolak penerapan sistem.

Tujuan perancangan sistem secara umum adalah memberikan gambaran secara umum atau global kepada pemakai tentang sistem yang akan dikembangkan dan berfungsi sebagai persiapan untuk tahap perancangan secara rinci, yang mana perancangan sistem secara umum akan mengidentifikasi komponen-komponen SIM dengan tujuan untuk dikomunikasikan dengan pemakai.

Berikut ini adalah aplikasi pendukung yang digunakan untuk proses perancangan system yang di buat

1. Draw.io

Draw.io adalah sebuah website yang didesain khusus untuk menggambarkan diagram secara online. Semua fitur yang ada pada situs ini bisa kalian nikmati hanya dengan bermodalkan browser yang mendukung HTML 5. Situs ini punya tampilan yang sangat responsive dan menggunakan fitur-fiturnya dengan mudah baik diakses menggunakan smartphone dan juga PC. Draw.io menawarkan

kemudahan untuk membuat diagram tanpa batasan jumlah diagram yang dibuat. Mantapnya lagi, situs ini terintegrasi dengan layanan penyimpanan file milik google yaitu Google Drive.

2. Balsamiq *Mock Up*

Balsamiq Mockups adalah salah satu software yang digunakan dalam pembuatan desain atau *prototyping* dalam pembuatan tampilan user interface sebuah aplikasi. Dengan menggunakan Balsamiq Mockup kita dimudahkan dalam pembuatan user interface karena Balsamiq Mockup sudah menyediakan tools yang dapat memudahkan dalam membuat desain *prototyping* aplikasi yang akan kita buat. Software ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

Beberapa alasan kami menggunakan aplikasi ini adalah :

- Low-Fi Sketch Wireframes: Samar, rendah-fidelity wireframes membiarkan Anda berfokus percakapan desain pada fungsi
- Komponen UI & Icon: 75 komponen built-in antarmuka pengguna dan 187 ikon, ditambah seluruh banyak komponen yang dihasilkan.
- Click-Through Prototype: Menghubungkan memungkinkan Anda menghasilkan klik melalui prototipe untuk demo & pengujian kegunaan
- Ekspor ke PNG atau PDF: Saham atau hadir maket dengan menggunakan link tertanam ekspor PDF, atau menggunakan alat pihak ke-3 untuk ekspor ke kode.

Balsamiq merupakan aplikasi yang disediakan untuk para designer guna mendesign mockups, dimana mockups itu menurut wiki adalah sebagai sebuah model dari suatu struktur atau alat baik *full size* ataupun berupa miniatur yang digunakan untuk pembelajaran, demo, *test* desain, promosi, dan lain-lain.

3. Visual Studio Code

Visual Studio Code adalah Software yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan built-in dukungan untuk JavaScript, naskah dan Node.js dan memiliki array beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C # , Python, dan PHP. Hal ini didasarkan sekitar Github ini Elektron, yang merupakan versi cross-platform dari Atom komponen kode-editing, berdasarkan JavaScript dan HTML5. Editor ini adalah fitur lengkap lingkungan pengembangan terpadu (IDE) dirancang untuk pengembang yang bekerja dengan teknologi cloud yang terbuka Microsoft. Visual Studio Code menggunakan open source NET perkakas untuk memberikan dukungan untuk ASP.NET C # kode, membangun alat pengembang Omnisharp NET dan compiler Roslyn. Antarmuka yang mudah untuk bekerja dengan, karena didasarkan pada gaya explorer umum, dengan panel di sebelah kiri, yang menunjukkan semua file dan folder Anda memiliki akses ke panel editor di sebelah kanan, yang menunjukkan isi dari file yang telah dibuka. Dalam hal ini, editor telah dikembangkan dengan baik, dan menyenangkan pada mata. Ia juga memiliki fungsi yang baik, dengan intellisense dan autocomplete bekerja dengan baik untuk JSON, CSS, HTML, {kurang}, dan Node.js.

Visual Studio Code telah dirancang untuk bekerja dengan alat-alat yang ada, dan Microsoft menyediakan dokumentasi untuk membantu pengembang bersama, dengan bantuan untuk bekerja dengan ASP.NET 5, Node.js, dan Microsoft naskah, serta alat-alat yang dapat digunakan untuk membantu membangun dan mengelola aplikasi Node.js. Visual Studio Code benar-benar sedang ditargetkan pada pengembang JavaScript yang ingin alat pengembangannya lengkap untuk scripting server-side mereka dan yang mungkin ingin usaha dari Node.js untuk kerangka berbasis NET. Visual Studio Code, adalah belum solid, lintas platform kode Editor ringan, yang dapat digunakan oleh siapa saja untuk membangun aplikasi untuk Web

2.1.5 UML

Menurut Nugroho (2010) *UML (Unified Modeling Language)* adalah 'bahasa' pemodelan untuk sistem atau perangkat lunak yang berparadigrama 'berorientasi objek'. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Bahasa pemodelan grafis telah ada diindustri perangkat lunak sejak lama. Pemicu utama di balik semuanya adalah bahwa bahasa pemrograman berada pada tingkat abstraksi yang terlalu tinggi untuk memfasilitasi diskusi tentang desain. Dengan menggunakan notasi-notasi seperti *UML*, alur logika dari perangkat lunak yang akan dikembangkan bisa mudah untuk dipahami.

Menurut Herlawati (2011), bahwa beberapa literature menyebutkan bahwa *UML* menyediakan sembilan jenis diagram, yang lain menyebutkan delapan

karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi.

Berdasarkan pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa “*Unified Modelling Language(UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek (*Object Oriented Programming*)”.

Berikut ini adalah symbol-simbol dari UML (*Unified Modelling Language*)

1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Tabel 2.1 Smbol *Use Case Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-------------------|--|
| 1 |  | <i>Actor</i> | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> . |
| 2 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>). |

| | | | |
|----|---|-----------------------|---|
| 3 |  | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>). |
| 4 |  | <i>Include</i> | Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> . |
| 5 |  | <i>Extend</i> | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan. |
| 6 |  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | <i>System</i> | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | <i>Use Case</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
| 9 |  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | <i>Note</i> | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu:

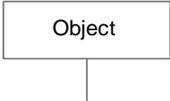
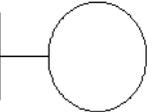
Tabel 2.2 Simbol *Activity Diagram*

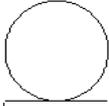
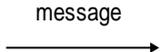
| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|----------------------------|---|
| 1 |  | <i>Activity</i> | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 |  | <i>Action</i> | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 |  | <i>Initial Node</i> | Bagaimana objek dibentuk atau diawali. |
| 4 |  | <i>Activity Final Node</i> | Bagaimana objek dibentuk dan dihancurkan |
| 5 |  | <i>Fork Node</i> | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |
| 6 |  | Decision | Pilihan untuk mengambil keputusan |

3. Diagram Urutan (*Sequence Diagram*)

Menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu:

Tabel 2.3 Simbol *Sequence Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------------|---|
| 1 |  | <i>Actor</i> | <i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. |
| 2 |  | <i>Object</i> | <i>Object</i> atau biasa juga disebut partisipan merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma. |
| 3 |  | <i>Lifeline</i> | <i>Lifeline</i> mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah objek. |
| 4 |  | <i>Message Return</i> | Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi. |
| 5 |  | <i>Activation</i> | <i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah objek yang melakukan suatu aksi. |
| 6 |  | <i>Boundary</i> | Boundary terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke sistem lainnya adalah termasuk dalam kategori. |
| 7 |  | <i>Control</i> | <i>Control</i> berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan. |

| | | | |
|----|---|-----------------------|--|
| 8 |  | <i>Entity</i> | <i>Entity</i> digunakan menangani informasi yang mungkin akan disimpan secara permanen. <i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data. |
| 9 |  | <i>Message</i> | <i>Message</i> , digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object-object</i> . |
| 10 |  | <i>Self - Message</i> | <i>Self-message</i> atau panggilan mandiri mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri. |

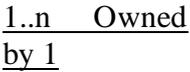
4. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan anturan-aturan dan tanggungjawab entitas yang menentukan perilaku sistem.

Class Diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan.

Class Diagram secara khas meliputi: kelas (*Class*), *Relasi Assosiations*, *Generalitation* dan *Agfregation*, attribute (*Attributes*), zoperasi (*Operation* atau *method*) dan *visability*. Tingkatakses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinaly*.

Tabel 2.4 Simbol *Class Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|--------------------|--|
| 1 |  | <i>Composition</i> | <p>Jika sebuah <i>class</i> tidak bisa berdirisendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/<i>solid</i>.</p> |
| 2 |  | <i>Aggregation</i> | <p><i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”. Sebuah <i>aggregation</i> digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid.</p> |
| 3 |  | <i>Class</i> | <p><i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi objek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i>. Bagian tengah mendefinisikan <i>property/atribut class</i>. Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i>.</p> |
| 4 |  | <i>Association</i> | <p>Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i>, dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i>. Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat</p> |

| | | | |
|---|------|-----------------------|---|
| | | | menampilkan hukum-hukum multiplisitas pada sebuah relationship (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>). |
| 5 | ← | <i>Generalization</i> | Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi objek. Sebuah <i>generalization</i> dilambangkan dengan sebuah panah dengan kepala. Panah yang tidak solid yang mengarah ke kelas “ <i>parent</i> ”-nya/induknya. |
| 6 | ←--- | <i>Dependency</i> | Kadangkala sebuah class menggunakan class yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik. |

2.2. TEORI KHUSUS

2.2.1 Definisi Analisa Sistem PIECES

Menurut Wukil Ragil (2010), metode *PIECES* adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi dan pelayanan pelanggan. Analisis ini disebut dengan *PIECES* Analysis (*Performance, Information, Economy, Control, Efficiency and Service*).

Analisis *PIECES* ini sangat penting untuk dilakukan sebelum mengembangkan sebuah sistem informasi karena dalam analisis ini biasanya akan ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Metode ini menggunakan enam variable evaluasi yaitu :

1. *Performance* (kinerja)

Kinerja merupakan variable pertama dalam metode analisis *PIECES*. Dimana memiliki peran penting untuk menilai apakah proses atau prosedur yang ada masih mungkin ditingkatkan kinerjanya, dan melihat sejauh mana dan seberapa handalkah suatu sistem informasi dalam berproses untuk menghasilkan tujuan yang diinginkan. Dalam hal ini kinerja diukur dari:

- a. *throughput*, yaitu jumlah pekerjaan/*output/deliverables* yang dapat dilakukan/ dihasilkan pada saat tertentu.
- b. *response time*, yaitu waktu yang dibutuhkan untuk menyelesaikan serangkaian kegiatan untuk menghasilkan *output/deliverables* tertentu.

2. *Information* (informasi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan menjadi semakin baik. Informasi yang disajikan haruslah benar-benar mempunyai nilai yang berguna. Hal ini dapat diukur dengan :

- a. Keluaran (*outputs*): Suatu sistem dalam memproduksi keluaran.

b. Masukan (*inputs*): Dalam memasukkan suatu data sehingga kemudian diolah untuk menjadi informasi yang berguna.

3. *Economic* (ekonomi)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya (nilai gunanya) atau diturunkan biaya penyelenggaraannya.

4. *Control* (pengendalian)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan/ kecurangan menjadi semakin baik pula.

5. *Efficiency* (efisiensi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi, dan harus lebih unggul dari pada sistem manual.

6. *Service* (layanan)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan. Buatlah kualitas layanan yang sangat user friendly untuk end – user (pengguna) sehingga pengguna mendapatkan kualitas layanan yang baik.

2.2.2 Definisi Pengembangan Sistem Waterfall/ Prototyping/Agile

Waterfall adalah suatu metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematis dan sekuensial yang

mulai pada tingkat kemajuan sistem pada seluruh analisis, design, kode, pengujian dan pemeliharaan. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut dengan “*classic life cycle*” atau model waterfall. Model ini adalah model yang muncul pertama kali yaitu sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE).

Proses pada metodologi *Waterfall*:

1. *Requirement Analysis*.

Seluruh kebutuhan *software* harus bisa didapatkan dalam fase ini, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

2. *System Design*

Tahap ini dilakukan sebelum melakukan *coding*. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

Dalam tahap ini dilakukan pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap

berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

4. Integration & Testing

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah software yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.

5. Operation & Maintenance

Ini merupakan tahap terakhir dalam model waterfall. Software yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

Keuntungan *Waterfall*:

1. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
2. Document pengembangan system sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.

3. Metode ini masih lebih baik digunakan walaupun sudah tergolong kuno, daripada menggunakan pendekatan asal-asalan. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

Kelemahan Waterfall:

1. Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
2. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan yang berakibat pada tahapan selanjutnya.
3. Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan.
4. Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
5. Pada kenyataannya, jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.

Prototyping

Prototyping adalah proses *iterative* dalam pengembangan sistem dimana *requirement* diubah ke dalam sistem yang bekerja (*working system*) yang secara terus menerus diperbaiki melalui kerjasama antara user dan analis. Prototype juga bisa dibangun melalui beberapa tool pengembangan untuk menyederhanakan proses.

Tujuan *Prototype*:

Prototyping model sendiri mempunyai tujuan yaitu mengembangkan model awal *software* menjadi sebuah sistem yang final.

Proses-proses dalam model *prototyping* secara umum adalah sebagai berikut:

1. Pengumpulan kebutuhan

Developer dan klien akan bertemu terlebih dahulu dan kemudian menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.

2. Perancangan

Perancangan dilakukan dengan cepat dan rancangan tersebut mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.

Evaluasi *Prototype*

Klien akan mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*.

Selain itu, untuk memodelkan sebuah perangkat lunak dibutuhkan beberapa tahapan di dalam proses pengembangannya. Tahapan inilah yang akan menentukan keberhasilan dari sebuah *software* itu. Pengembang perangkat lunak harus memperhatikan tahapan dalam metode *prototyping* agar *software* akhirnya

dapat diterima oleh penggunanya. Dan tahapan-tahapan dalam prototyping tersebut adalah sebagai berikut:

1. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format dan kebutuhan keseluruhan perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun *prototyping*.

Membangun *prototyping* dengan membuat perancangan sementara yang berpusat pada penyajian kepada pelanggan (misalnya dengan membuat input dan contoh *outputnya*).

3. Evaluasi *protootyping*

Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah keempat akan diambil. Jika tidak, maka prototyping diperbaiki dengan mengulang langkah 1, 2, dan 3.

4. Mengkodekan *system*

Dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji *system*

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box*, *Black Box*, *Basis Path*, pengujian arsitektur dan lain-lain.

6. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika sudah, maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah 4 dan 5.

7. Menggunakan *system*

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan

Keunggulan *prototyping*:

1. Komunikasi akan terjalin baik antara pengembang dan pelanggan.
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan setiap pelanggannya.
3. Pelanggan berperan aktif dalam proses pengembangan sistem.
4. Lebih menghemat waktu dalam pengembangan sistem.
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya

Kelemahan *prototyping*:

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
2. Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat prototyping lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan sebuah kerangka kerja(blueprint) dari sistem.
3. Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik dan benar.

AGILE

Agile Development Methods adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. *Agile development methods* merupakan salah satu dari Metodologi pengembangan perangkat lunak yang digunakan dalam pengembangan perangkat lunak. Agile memiliki pengertian bersifat cepat, ringan, bebas bergerak, dan waspada. Sehingga saat membuat perangkat lunak dengan menggunakan *agile development methods* diperlukan inovasi dan responsibiliti yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang.

Dalam proses pengembangan agile kita mengenal dengan iterasi atau perulangan. Jika suatu proyek pengembangan *software* dikerjakan dengan menggunakan metode agile, maka selama waktu pengerjaannya akan selalu dijumpai proses pengembangan yang dilakukan berulang. Setiap perulangan (*iterasi*) meliputi berbagai kegiatan yang wajib dilakukan dalam proyek pengembangan software itu sendiri, yaitu:

a. *Requierements Analysis*

Langkah ini merupakan analisa terhadap kebutuhan *system*. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau *study literature*. Seorang *system* analisis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen user requierements atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan *system*. Dokumen inilah yang akan menjadi acuan *system* analis untuk menerjemahkan ke dalam bahasa pemrograman.

b. Desain

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada: struktur data arsitektur perangkat lunak, *representasi interface*, dan detail (algoritma) *procedural*. Tahapan ini akan menghasilkan dokumen yang

disebut *software requirement*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

c. *Coding*

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh *computer*. Dilakukan oleh *programmer* yang akan menterjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu system. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini.

d. *Testing*

Testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

Kelebihan dari agile:

1. Meningkatkan kepuasan kepada klien.
2. Dapat melakukan review pelanggan mengenai software yang dibuat lebih awal.
3. Pembangunan *system* dibuat lebih cepat.
4. Mengurangi resiko kegagalan implementasi software dari segi non-teknis.
5. Jika pada saat pembangunan system terjadi kegagalan kerugian dari segi materi relatif kecil.

Kekurangan dari agile:

1. *Developer* harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
2. Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
3. Tidak cocok dalam skala tim yang besar (>20 orang).
4. Perkiraan waktu release dan harga perangkat lunak sulit ditentukan

2.2.3 Definisi Aplikasi Web Based / Android

Web aplikasi (*Web Application*) merupakan aplikasi yang dapat diakses melalui web dengan menggunakan jaringan internet atau intranet. Mari kita bahas lebih lengkap tentang aplikasi web dan cara membuatnya.

Sebelum pembahasan lebih lanjut, pengertian dari *website* itu sendiri adalah halaman situs yang berisi informasi dan dapat diakses oleh banyak orang. Kemunculan *website* didasari atas perkembangan teknologi informasi dan komunikasi.

Dari perkembangan teknologi informasi, maka terciptalah suatu jaringan antar komputer yang saling berkaitan satu sama lainnya. Jaringan yang disebut internet ini digunakan secara terus menerus dan berkembang menjadi pesan elektronik meliputi *email*, *transmisi file* serta komunikasi lain secara dua arah antar individu atau komputer.

Web aplikasi merupakan sebuah perangkat lunak komputer yang dikodekan dalam bahasa pemrograman yang mendukung perangkat lunak berbasis web

seperti HTML, CSS, JavaScript, Ruby, Python, Php, Java serta bahasa pemrograman lainnya.

Bahasa Pemrograman HTML (*Hypertext Markup Language*) yakni bahasa standar yang digunakan untuk membangun aplikasi web. Sebuah aplikasi web akan selalu ada struktur HTML di dalamnya. Oleh karena itu HTML adalah bahasa pemrograman yang sangat wajib untuk dikuasai.

2.2.4 Definisi Aplikasi Aplikasi MySQL

MySQL adalah sebuah database management system (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. *Database management system* (DBMS) MySQL multi pengguna dan multi alur ini sudah dipakai lebih dari 6 juta pengguna di seluruh dunia.

MySQL adalah DBMS yang open source dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah database server yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

Seperti yang sudah disinggung di atas, MySQL masuk ke dalam jenis RDBMS (*Relational Database Management System*). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah database terdapat satu atau beberapa tabel.

SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada relational database atau database yang terstruktur. Jadi MySQL adalah *database management system* yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan database server.

2.2.5 Definisi PHP

PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum (*wikipedia*). PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di <http://www.php.net>.

PHP disebut bahasa pemrograman *serverside* karena PHP diproses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada web browser (client).

Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti *wikipedia*, *wordpress*, *joomla*, dll.

Saat ini PHP adalah singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: *Hypertext Preprocessor*.

PHP dapat digunakan dengan gratis (free) dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP License*, sedikit berbeda dengan lisensi *GNU General Public License (GPL)* yang biasa digunakan untuk proyek *Open Source*.

Kemudahan dan kepopuleran PHP sudah menjadi standar bagi programmer web di seluruh dunia. Menurut wikipedia pada february 2014, sekitar 82% dari web server di dunia menggunakan PHP. PHP juga menjadi dasar dari *aplikasi CMS (Content Management System)* populer seperti Joomla, Drupal, dan WordPress.

2.2.6 Definisi XAMPP

Menurut Riyanto (2015), XAMPP adalah sebuah software yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolah data MySQL yang dijalankan dikomputer secara lokal. XAMPP berperan sebagai web server pada komputer. XAMPP juga dapat disebut sebuah CPanel server virtual, yang dapat membantu Anda melakukan preview sehingga dapat memodifikasi website tanpa harus online atau terakses dengan internet. Software XAMPP dibuat dan dikembangkan oleh Apache Friends. Perangkat lunak komputer ini memiliki kelebihan untuk bisa berperan sebagai server web Apache untuk simulasi pengembangan website. Tool pengembangan web ini mendukung teknologi web populer seperti PHP, MySQL, dan Perl. Melalui program ini, programmer web dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. XAMPP juga dilengkapi fitur manajemen basis data *PHPMyAdmin* seperti pada

server hosting sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis database secara mudah.

2.2.7 Definisi Laravel

Menurut Hakim dalam buku Konsep dan Implementasi Pemrograman Laravel 7 Edisi Best Practice (2020), “Laravel adalah *framework* berbasis bahasa pemrograman PHP yang dapat digunakan untuk membangun aplikasi web mulai dari proyek berskala kecil (sederhana) hingga besar (kompleks). Laravel dibangun dengan konsep MVC (*Model-View-Controller*), selain itu Laravel merupakan *framework* PHP yang modern dengan tingkat keamanan (*security*) yang bagus. Pada setiap release versi terbarunya selalu menambahkan sekuritas dan fitur-fitur web terkini.”

2.2.8 Definisi Bootstrap

Menurut Herlangga (2015) Bootstrap adalah sebuah *framework* css yang dapat digunakan untuk mempermudah membangun tampilan web. Bootstrap pertama kali dikembangkan pada pertengahan 2010 di Twitter oleh Mark Otto dan Jacob Thomson.

Menurut Wandela dan Elisa (2019) dalam Maydianto dan Muhammat Rasid Ridho (2021) Bootstrap merupakan sebuah *front-end framework* yang mengedapankan tampilan untuk *mobile device phone*. Kelebihan saat memakai Bootstrap yaitu diciptakan memakai Less, teknologi CSS yang sederhana dan mudah untuk digunakan. Dengan adanya Less, maka pengembang bisa melakukan akses yang lebih mudah informasi dan fungsi warna, variabel, dan operasi penggunaan (Christian et al., 2018).

2.2.9 Definisi ERD (*Entity Relationship Diagram*)

Menurut Rosa A.S., dan Shalahuddin (2015) mendefinisikan bahwa “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan”.

Menurut pendapat Kronke (2006) Entity Relationship Diagram (ERD) adalah suatu pemodelan konseptual yang didesain secara khusus untuk mengidentifikasi entitas yang menjelaskan data dan hubungan antar data, yaitu dengan menuliskan dalam cardinality. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database. Dalam sistem Entity Relationship Diagram (ERD), terdapat beberapa istilah penting diantaranya:

1. Entitas (Entity)

Suatu entitas yang dapat berupa orang, tempat, obyek, atau kejadian yang dianggap penting bagi perusahaan, sehingga segala atributnya harus dicatat dan disimpan dalam basis data.

2. Atribut (Attribute)

Setiap entitas mempunyai karakteristik tertentu yang dinamakan dengan atribut.

3. Relasi (relationship)

Hubungan antara dua atau lebih entitas yang saling berkaitan. Menurut Romney (2009) ada tiga tipe relasi (relationship), yaitu:

a. One-to-one relationship (1:1) Dimana maximum cardinality setiap entity adalah 1. Contoh: Satu nasabah bank hanya memiliki satu account.

b. One-to-many relationship (1:N). Dimana maximum cardinality dari suatu entity adalah 1 dan maximum cardinality dari entity lain adalah N. Contoh: Satu nasabah bank dapat memiliki lebih dari satu account.

c. Many-to-many relationship (M:N). Dimana maximum cardinality kedua entity yang berhubungan adalah N. Contoh: Satu nasabah dapat memiliki beberapa account dan satu account dapat dimiliki oleh beberapa nasabah (rekening bersama).

Dapat disimpulkan bahwa entity relationship diagram (ERD) merupakan teknik pemodelan struktur data secara konseptual yang menggambarkan entitas lengkap dengan atributnya dan hubungan yang terjadi antar entitas tersebut.

2.2.10 Definisi Pengujian Sistem *Black Box*

Black Box Testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan cek fungsional perangkat lunak. Jadi analogi seperti yang kita lihat kotak hitam, kita dapat melihat hanya penampilan luar, tanpa mengetahui apa di balik bungkus hitamnya. Sama seperti pengujian kotak hitam, mengevaluasi hanya penampilan eksternal (antarmuka), fungsional tanpa mengetahui apa yang sebenarnya terjadi dalam codingan.

Tujuan ***Black Box Testing*** adalah untuk mencari kesalahan/kegagalan dalam operasi tingkat tinggi, yang mencakup kemampuan dari perangkat lunak, operasional/tata laksana, skenario pemakai. Fungsi dari pengujian ini berdasarkan kepada apa yang dapat dilakukan oleh sistem. Untuk melakukan pengujian

perilaku seseorang harus mengerti lingkup dari aplikasi, solusi bisnis yang diberikan oleh aplikasi, dan tujuan sistem dibuat.

Contoh pengujian pada aplikasi internet banking, maka pengujian yang dilakukan adalah menjalankan aplikasi, memeriksa apakah semua fungsi pada aplikasi berjalan dengan baik serta mengecek tampilan dari aplikasi tersebut apakah sesuai dengan design yang sudah ditentukan atau belum.

2.2.11 Definisi Problem Shuting

Problem Shutting adalah sebuah istilah dalam bahasa Inggris, yang merujuk kepada sebuah bentuk penyelesaian sebuah masalah. *Problem Shutting* merupakan pencarian sumber masalah secara sistematis sehingga masalah tersebut dapat diselesaikan. *Troubleshooting*, kadang-kadang merupakan proses penghilangan masalah, dan juga proses penghilangan penyebab potensial dari sebuah masalah. *Troubleshooting*, pada umumnya digunakan dalam berbagai bidang, seperti halnya dalam bidang komputer, administrasi sistem, dan juga bidang elektronika dan kelistrikan.

2.3 Literatur Review

Sebagai salah satu upaya yang perlu dilakukan dalam penelitian untuk mengidentifikasi kesenjangan (*identify gaps*), menghindari pengulangan pembuatan (*reinventing the wheel*), mengidentifikasi metode yang pernah dilakukan, serta mengetahui penerapan metode yang dilakukan orang lain dalam area penelitian yang mendekati bidang yang sama dengan penelitian ini, Maka

berikut lima penelitian yang memiliki korelasi searah dengan penelitian yang akan dibahas:

- **Qurotul Aini, Yuliana Isma Graha, Siti Ria Zuliana (2017)** Dalam penelitian jurnal di STMIK Raharja Jurusan Sistem Informasi yang berjudul “Penerapan Absensi QRCode Mahasiswa Bimbingan Belajar pada Website berbasis Yii Framework” Perkembangan internet berkembang begitu pesat dan terus berinovasi dan menciptakan terobosan yang baru. Dalam meningkatkan kualitas sistem pembelajaran pada saat bimbingan belajar berlangsung yaitu sistem layanan absensi scanning qrcode untuk merekam kehadiran mahasiswa dengan baik. Dalam penelitian ini menggunakan sistem PenA (Penilaian Absensi) memberikan layanan dalam proses absensi scanning qrcode, menghindari terjadinya kecurangan yang dilakukan mahasiswa dimana mahasiswa dapat menitipkan absensi yang dilakukan oleh antar mahasiswa dalam bimbingan berlangsung. Penelitian ini menggunakan Aplikasi absensi dengan menggunakan Website berbasis *Framework Yii*, *xampp* sebagai *web server* nya, PHP bahasa scripting-nya dan *MySQL* sebagai database manajemen sistem (DBMS). Yang dimana penelitian ini akan memudahkan Pimpinan dan Pembimbing mahasiswa mendapatkan rekapitulasi absensi secara *online*, dan data absensi lebih terstruktur dan tidak akan mudah tercecer lagi, karena sudah tersimpan secara terintegrasi melalui pengolahan database yang baik.

- **Qianwen Miao, Fu Xiao, Haiping Huang, Lijuan Sun, and Ruchuan Wang (2020)** Dalam penelitian jurnal yang berjudul “*Smart Attendance System Based on Frequency Distribution Algorithm with Passive RFID Tags* (Sistem Absensi Cerdas Berdasarkan Algoritma Distribusi Frekuensi dengan Tag RFID Pasif)” Informasi kehadiran staf selalu menjadi bagian penting dari manajemen perusahaan. Namun, beberapa karyawan mungkin mengirim orang lain untuk memencet kartu waktu mereka, yang menghambat keaslian kehadiran dan efektivitas pencatatan. Oleh karena itu, perlu dikembangkan anti-kecurangan yang inovatif sistem untuk kehadiran di kantor. Identifikasi Frekuensi Radio (*RFID*) menawarkan solusi baru untuk menyelesaikan masalah tersebut karena kemampuan anti-interferensi yang kuat dan tidak mengganggu. Dalam proses ini dilakukan pengujian dengan pendeteksi manusia berbasis *WiFi* dan berbasis *RFID* teknologi. Dari hasil penelitian ini didapat perangkat sistem kehadiran yang mempresentasikan metodologi desain, implementasi, dan evaluasi dari sistem yang dapat membedakan berbagai target sesuai dengan sinyal fase unik individu di LOS link, seperti yang dikumpulkan dari RFID reader. Untuk meningkatkan akurasi identifikasi sistem, menggunakan histogram distribusi frekuensi dan K-means algoritma untuk mengekstrak sidik jari.
- **Albertus L. Setyabudhi (2017)** dalam penelitian jurnal di STT Ibnu Sina yang berjudul “Perancangan Sistem Informasi Pengolahan Data Absensi dan Pengambilan Surat Cuti Kerja Berbasis Web”. Data absensi dan surat izin cuti kerja yang telah diolah sangat diperlukan untuk bahan evaluasi dalam

pemberian cuti kerja kepada karyawan. Pengolahan data absensi dan pengambilan surat izin cuti kerja adalah sarana untuk menyimpan data absensi dan surat izin cuti kerja secara sistimatis, sehingga memudahkan penemuan kembali jika diperlukan. Pengelolaan absensi dan cuti kerja yang bersifat manajerial maupun teknis administratif selalu berhubungan dengan data, dalam bentuk yang tercetak (manual) maupun data elektronik (media komputer). Kegiatan administrasi absensi maupun cuti kerja akan berpengaruh pada keadaan data perorangan karyawan maupun keseluruhan serta semua proses yang ada dalam sebuah perusahaan. Beberapa perusahaan di Batam dalam sistem pengolahan data absensi dan pengambilan surat izin cuti kerja masih menggunakan sistem semi komputerisasi yaitu dengan menggunakan aplikasi micorosoft excel dan microsoft word yang selanjutnya akan dicetak dan disimpan dalam bentuk hardcopy di sebuah almari khusus. Hal ini memiliki resiko yang sangat tinggi untuk kehilangan data-data tersebut. Kehilangan data karena hardcopy yang ada telah rusak, ataupun data file yang lama tergantikan dengan file yang baru. Tentunya ini akan berdampak pada terhambatnya proses pembuat laporan, evaluasi dan pemberian cuti kerja sehingga membuat proses menjadi kurang efektif dan efisien. Tujuan dari penelitian ini adalah merancang, mengelola dan mengimplementasikan pengambil cuti kerja pada perusahaan. Rancangan sistem informasi ini menggunakan pemodelan data flow diagram, relation diagram. Implementasi dari sistem informasi ini ditujukan agar proses pengambilan surat izin cuti

kerja dapat lebih efektif dan efisien dengan menggunakan metode berbasis *WEB*.

- **Al Husain, Abdul Haqy Aji Prastian, Andre Ramadhan (2017)** dalam penelitian jurnal *Technomedia Journal (TMJ)* yang berjudul “Perancangan Sistem Absensi Online Menggunakan Android Guna Mempercepat Proses Kehadiran Karyawan Pada PT. Sintech Berkah Abadi” Sistem absensi saat ini perlu memerlukan pembaharuan sistem yang baru. Karena dengan melakukan hal ini akan berdampak sangat baik bagi perusahaan dilihat dari sisi efisien, efektif, dan cepat. Dalam hal ini perusahaan berlomba membuat sistem absensi yang dapat berjalan pada teknologi saat ini seperti smartphone android. Hal tersebut membutuhkan sebuah design rancangan sebelum sistem ini dibuat sehingga dapat mengetahui permasalahan dalam perusahaan. Tujuan dari penelitian ini adalah Mempermudah karyawan dalam absensi dengan android. Pengakses yang sangat cepat dalam hal absensi karena menggunakan android dan jaringan WLAN. Dengan adanya sistem absensi online ini, karyawan dapat mengoptimalkan pekerjaan dengan sangat baik. Karena sistem absensi ini diakses sangat mudah dan waktu pun dapat digunakan secara efisien.
- **Subiantoro dan Sardiarinto (2018)** Dalam penelitian yang berjudul “Perancangan Sistem Absensi Pegawai Berbasis WEB, Studi Kasus : Kantor Kecamatan Purwodadi”. Kantor Kecamatan Purwodadi Kabupaten Purworejo merupakan salah satu instansi Pemerintah yang mempunyai tanggung jawab terhadap pelayanan masyarakat dalam bidang pelayanan sosial. Jumlah

pegawai yang dimiliki saat ini 48 personel dan dapat bertambah sesuai kebutuhan instansi. Semakin besar instansi pemerintah maka akan berdampak semakin besar pula dengan Sumber Daya Manusia (SDM) yang dibutuhkan. Berdasarkan hal itu maka instansi pemerintah dituntut untuk melakukan manajemen pengolahan data absensi yang baik, cepat dan efisien. Oleh karena itu tujuan penelitian ini menggunakan aplikasi absensi berbasis web pegawai yang dapat meminimalisir kehilangan dan kesalahan pencatatan data baik dalam proses absensi itu sendiri maupun pembuatan laporan absensi.

BAB III

GAMBARAN UMUM PENELITIAN

3.1 Jenis Penelitian

Penelitian yang peneliti lakukan menggunakan metode penelitian campuran (kuantitatif & kualitatif), dimana pengumpulan data akan dilakukan dengan cara metode survei menggunakan daftar pernyataan/*questioner* dan melakukan wawancara langsung ke pihak-pihak terkait.

Teori pengembangan sistem menggunakan metode pengembangan sistem model *waterfall*. Tahapan analisis dan perancangan sistem menggunakan pendekatan berorientasi objek, implementasi hasil analisis dan perancangan berbasis PHP dan database MySQL, Kualitas system perizinan yang diimplementasikan diuji berdasarkan pengujian *black-box*.

3.2 Metode Pemilihan Sampel

Metode pengambilan sampel yang digunakan adalah *purposive sampling*, dengan mengambil responden dipilih oleh peneliti menurut ciri-ciri spesifik yang dimiliki sampel.

Adapun kriteria orang yang dipilih sebagai sampel penelitian ini adalah:

1. Mengetahui proses absensi dan perizinan karyawan *outsourcing* Mayora Jayanti 1.
2. Mengetahui tentang pentingnya laporan KPI.
3. Menyadari pentingnya data dan pentingnya memiliki *system* perizinan yang teratur.

4. Mengetahui pentingnya pengelolaan data sebagai tim *payroll*.
5. Mengetahui dengan implementasi aplikasi sistem informasi laporan absensi karyawan.

Pemilihan responden sampel yang akan disertakan dalam proses pengumpulan data/informasi berdasarkan pertimbangan kriteria yang dijelaskan sebelumnya, maka peneliti menilai sampel yang representatif adalah:

1. Dept. Head IR & GA
2. Supervisor CI (continuous improvement)
3. Staf Payroll Mayora Jayanti 1

3.3 Metode PengumpulanData

Metode pengumpulan data ialah teknik atau cara-cara yang dapat dilakukan oleh peneliti untuk mengumpulkan data (Sudaryono 2010:125).

Metode pengumpulan data yang digunakan dalam penelitian adalah metode campuran dimana dilakukan pengumpulan data dengan cara penyebaran daftar pernyataan/*questioner* dan melakukan wawancara/ *interview*.

Metode Pengumpulan data yang dijalankan dalam penelitian ini adalah:

1. Metode Wawancara.

Peneliti telah menyiapkan daftar pertanyaan yang berkaitan dengan pengumpulan informasi, untuk ditanyakan kepada : Dept Head IR & GA (Bpk. Yasir Nasution), Supervisor CI (Bpk. Gema), Staf Payroll Mayora Jayanti 1 (Ibu Wida Yuningsih).

Beberapa pertanyaan yang diajukan pada saat wawancara.

- a. Apa saja kendala yang terjadi dari alur perizinan karyawan?
- b. Apakah system yang selama ini berjalan sudah sesuai?
- c. Kenapa masih banyak karyawan yang telat mengurus surat izin?
- d. Apa akibat nya jika karyawan telat mengurus surat izin?
- e. Mengapa perizinan bisa berpengaruh terhadap KPI?
- f. Kapan waktu yang tepat bagi karyawan mengurus surat izin?

Kesimpulan yang peneliti dapat dari hasil wawancara adalah:

- a. Masih kurang nya kesadaran karyawan terhadap system perizinan yang sudah memiliki batas waktu.
- b. Penyerahan surat izin yang masih manual.
- c. Jam kerja karyawan terbagi 3 shift.

2. Metode Observasi.

Observasi adalah suatu kegiatan pengamatan langsung terhadap profil organisasi dan objek penelitian. Kesimpulan yang peneliti dapat dari hasil observasi adalah banyak nya karyawan telat mengurus surat izin melebihi batas (3 hari), terutama apabila izin tersebut terjadi di akhir week, karna pada awal week laporan absensi week sebelumnya sudah harus di serahkan kepada Dept. CI.

3. Metode Studi Pustaka.

Peneliti mengumpulkan data dengan cara mempelajari, meneliti, dan membaca buku, jurnal, skripsi, dan tesis yang berhubungan dengan pengembangan sistem informasi absensi dan perizinan karyawan. Kesimpulan yang peneliti dapat dari hasil studi pustaka yang peneliti jalankan adalah ada banyak cara atau metode yang bisa digunakan untuk memudahkan karyawan

memberikan surat izin kepada tim payroll.

3.4 Metode Analisa data

Metode analisis data merupakan tahapan proses penelitian dimana data yang sudah dikumpulkan dikelola untuk diolah dalam rangka menjawab permasalahan yang ada. Manajemen dan proses pengolahan data inilah yang disebut dengan analisis data. Analisis data sendiri memiliki tujuan untuk mendeskripsikan data agar mudah dipahami, kemudian membuat informasi yang menarik mengenai karakteristik dari populasi data berdasarkan sampel data yang dibuat berdasarkan hasil pengujian hipotesis.

Hasil analisa ini terbagi menjadi beberapa bagian, yaitu:

1. *Performance* (kinerja)

Dari segi *performance* menurut analisa yang sudah dilakukan, system yang sudah berjalan saat ini masih kurang efektif dikarenakan masih adanya beberapa aspek yang membuat proses adminitrasi berjalan kurang maksimal sehingga tidak bisa menyelesaikan dengan cepat.

2. *Information* (informasi)

Dari segi informasi, system yang berjalan saat ini tentu masih belum maksimal dikarenakan informasi izin yang akan dilakukan oleh karyawan masih terlambat sampai kepada tim payroll dikarenakan beberapa kendala yang terjadi, diharapkan system baru yang akan diajukan bisa menjadi solusi yang dapat menyelesaikan lambat atau lama nya informasi karyawan yang melakukan izin kepada tim payroll.

3. *Ekonomi*

Dari segi ekonomi, sistem yang berjalan saat ini tentu masih dapat diperbaiki, karena system yang akan di usulkan dapat mengurangi pemakaian kertas yang berlebihan. Karena selama ini masih menggunakan form izin yang menggunakan kertas 3 (tiga) rangkap yang tentunya akan menggunakan pengeluaran untuk membuat form dari setiap yayasan *outsourcing*.

4. **Control (pengendalian)**

Dari segi pengendalian, system yang berjalan cukup baik karena setiap atasan mengetahui bahwa karyawan nya melakukan izin, tetapi para atasan tidak bisa mengetahui siapa saja yang sudah melakukan izin karena tidak adanya report laporan ataupun arsip yang mereka miliki. Sistem yang diusulkan akan memungkinkan para atasan untuk mengetahui siapa saja yang sudah melakukan izin selama masa kerja nya karena memiliki *report* dari *approve* yang mereka lakukan.

5. **Efficiency**

Dari segi efisiensi, *system* yang sudah berjalan masih termasuk kedalam golongan kurang efisien dari proses waktu dan juga proses informasi.

6. **Service (layanan)**

Dari segi layanan, *system* yang berjalan masih sangat bisa diperbaiki atau ditingkatkan, karena seperti yang sudah berjalan layanan terhadap penerimaan form izin masih terhalang oleh jam operasional penerimaan surat izin. *System* yang akan diusulkan diharapkan akan sangat memudahkan karyawan mendapatkan layanan pelaporan izin yang memudahkan tanpa harus terikat oleh jam operasional dari tim *payroll*.

3.5 Metode Pengujian

Pada tahap ini penulis menggunakan metode Kotak Hitam, (*Black Box*). Sebuah objek adalah kotak hitam. Konsep ini menjadi dasar implementasi objek. Dalam operasi hanya developer yang dapat memahami detail proses yang ada didalam kotak tersebut, sedangkan user tidak perlu mengetahui apa yang dilakukan yang penting mereka dapat menggunakan objek untuk memproses kebutuhan mereka. Kotak hitam berisi kode dan data.

1. Encapsulation, yaitu proses menyembunyikan detail implementasi sebuah objek. Untuk mengakses data objek tersebut adalah melalui interface. Untuk berkomunikasi dengan objek digunakan message.
2. Message adalah permintaan agar objek menerima untuk membawa metode yang ditunjukkan oleh perilaku dan mengembalikan result dari aksi tersebut kepada objek pengirim (sender)

3.6 Tinjauan Tempat Penelitian

3.6.1 Sejarah Perusahaan



Gambar 3.1 Mayora Head Office

Sejak pertama kali didirikan pada tahun 1977, PT. Mayora Indah Tbk telah menjadi salah satu industri makanan penting di Indonesia. Sebagai hasil dari pertumbuhan negara di bidang ekonomi dan pergeseran pola konsumtif terhadap produk sosial yang lebih praktis, PT Mayora menawarkan pertumbuhan pesat selama bertahun-tahun. PT. Mayora Indah Tbk, merupakan kelompok bisnis yang memproduksi makanan terkemuka di Indonesia. Mayora Indah telah berkembang menjadi salah satu perusahaan *Fast Moving Consumer Goods Industry* yang telah diakui keberadaannya secara *global*. Terbukti bahwa PT. Mayora Indah Tbk, telah menghasilkan berbagai produk berkualitas yang saat ini menjadi merek-merek terkenal di dunia, seperti Kopiko, Danisa, Astor, Energen, Torabika dll.

Perusahaan ini pertama kali didirikan sejak 17 Februari 1977 sebagai sebuah industri biskuit rumah sederhana yang hingga sekarang mampu berkembang dengan pesat menjadi salah satu kelompok usaha yang ter-integrasi di Indonesia. Perkembangan perusahaan juga ditorehkan dengan merubah status perusahaan menjadi perusahaan terbuka seiring dengan pencatatan saham perusahaan untuk pertama kali di Bursa Efek Jakarta sejak tanggal 4 Juli 1990. Pada tahun-tahun berikutnya perusahaan terus melakukan ekspansi cepat untuk menjadi sebuah perusahaan yang berbasis ASEAN. Salah satu usaha-nya yakni mendirikan fasilitas produksi dan beberapa kantor pemasaran yang terletak di beberapa negara di Asia Tenggara.

Dengan inovasi-inovasi terbaru yang dilakukan perusahaan semakin memperkokoh posisi PT. Mayora Indah Tbk, di pasar *global*. Terbukti bahwa

produk-produk Mayora tidak hanya mampu memenuhi konsumen yang ada di dalam negeri saja, namun telah menjangkau konsumen luar negeri bahkan hampir menyebar di seluruh dunia. Hasil ini dapat dicapai berkat dukungan dari jaringan distribusi yang kuat selain tersedia-nya fasilitas dengan sistem logistik dan pengelolaan gudang yang modern. Selain itu, perusahaan telah menerapkan tiga visi utama perusahaan yang menjadi acuan pengelolaan di antaranya menjadi produsen makanan dan minuman berkualitas yang dipercaya oleh konsumen baik di pasar domestik dan internasional serta mengendalikan pangsa pasar yang signifikan dalam setiap kategori, memberikan nilai tambah bagi seluruh pemegang saham perusahaan dan memberikan kontribusi positif terhadap lingkungan dan negara di mana perusahaan beroperasi. Produk-produk PT. Mayora Indah Tbk, dibagi menjadi beberapa lini produk dengan merek-merek terkenal, antara lain Biskuit dengan pabrik biskuit terbesar di Asia Tenggara (Roma Marie, Slai O'lai, Better, Danisa dan Sari Gandum), Permen yang menjadi salah satu pelopor permen kopi dan menjadi merek permen nomor 1 di dunia (Kopiko, Kiss, dan Tamarin), Wafer & Chocolate yang menjadi pelopor hadir-nya wafer roll dan coklat pasta dengan kualitas tinggi (Astor, Beng-beng, Superstar, Zuperr Keju, dan choki-choki), Kopi yang merupakan produsen kopi instan terbesar di Indonesia dan Asia Tenggara (Torabika Brown Coffe, Torabika 3inOne, Torabika Cappucino, Toabika Duo), Mayora Nutrition (Energen Oat Milk), Bubur (Super Bubur), Mie Instan (Mie Gelas, Bakmi Mewah Rasa), Minuman (Vitazone, Teh Pucuk Harum, Kopiko 78°C, Q Guava, Le- Minerale) dan beberapa varian produk lainnya. Saat ini produk-produk tersebut telah didistribusikan ke lebih dari 52

negara di dunia seperti Amerika Serikat, Rusia, Australia, Belanda, Jerman, Jepang, Iran, Italia, Inggris, Spanyol, Korea Selatan, Saudi Arabia, Portugal dan beberapa negara lainnya. Untuk memenuhi tuntutan pasar yang semakin meningkat, PT Mayora Indah Tbk menjadi *go public* melalui *Initial Public Offering (IPO)* pada tahun 1990. Sebuah langkah sukses yang nyata dalam wujud pabrik-pabrik di Tangerang, Bekasi dan Surabaya, yang mempekerjakan lebih dari 5.000 pekerja.

Salah satu nya adalah PT. Mayora Indah Plant Jayanti 1 yang menjadi tempat penelitian kali ini dilakukan, yang dimana *Plant* ini baru berdiri pada tahun 2013 yang beralamat di Jl. Raya Serang KM. 31Desa Sumur Bandung Kec. Jayanti Kab. Tangerang – Banten.

3.6.2 Produk Perusahaan

PT. Mayora Indah Plant Jayanti 1 adalah sebuah perusahaan yang bergerak di bidang makanan, Berikut adalah beberapa produk yang ditawarkan:

1. Roma Malkist Abon, Malkist Cream, Malkist Cheese, Malkist Crackers, Malkist Cokelat.
2. Roma Kelapa
3. Roma Sandwich Coklat dan Kacang
4. Danisa Butter Cookies dan Royal Choice

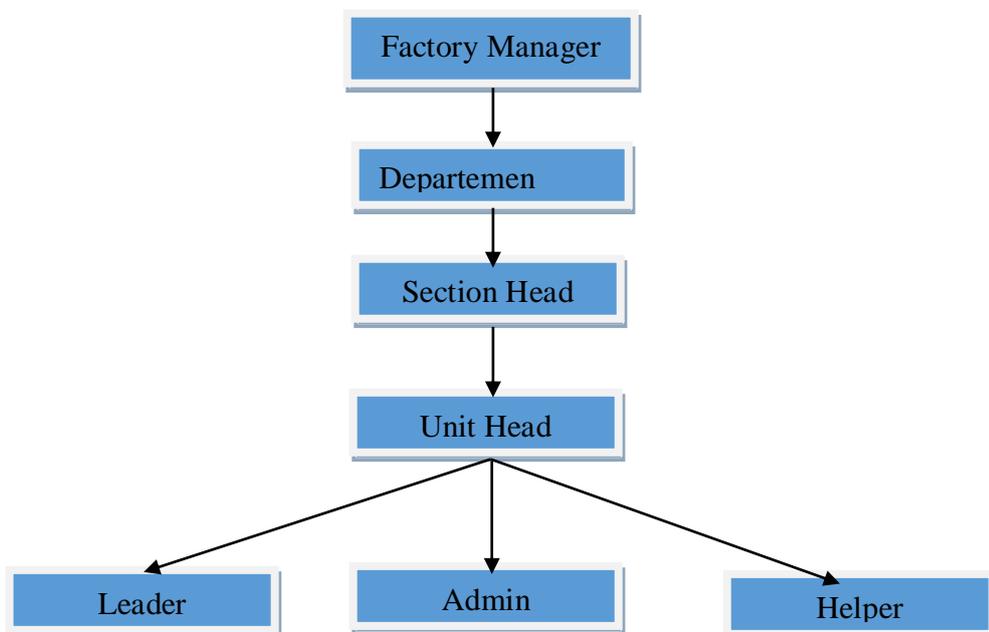
3.6.3 Visi dan Misi Perusahaan

Visi dan Misi Perusahaan

- Menjadi produsen makanan dan minuman yang berkualitas dan terpercaya di mata konsumen domestik maupun internasional dan menguasai pangsa pasar terbesar dalam kategori produk sejenis.
- Dapat memperoleh Laba bersih operasi diatas rata rata industri dan memberikan value added yang baik bagi seluruh stakeholders Perseroan.
- Dapat memberikan kontribusi positif terhadap lingkungan dan negara dimana Perseroan berada.

3.6.4 Skema Struktur Organisasi Perusahaan

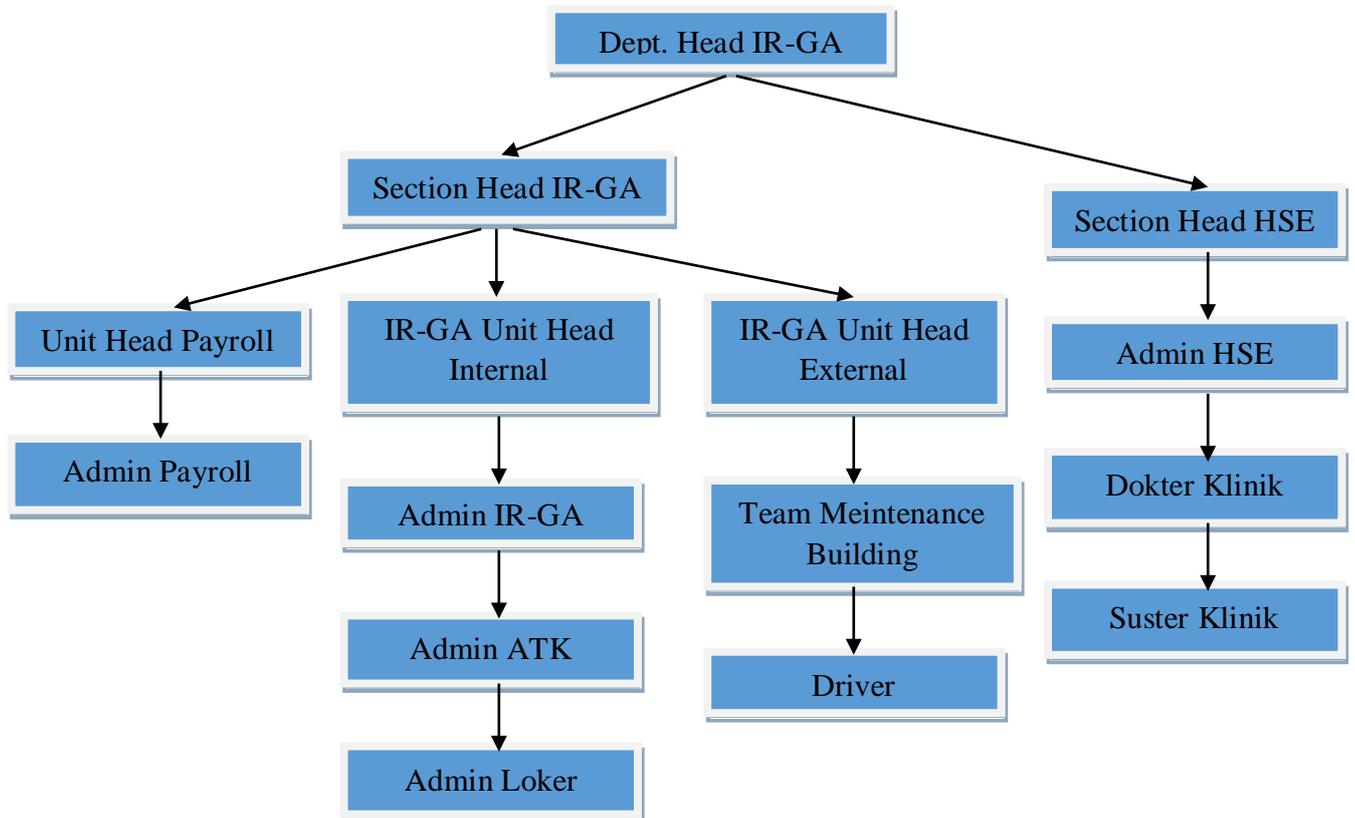
Berikut gambaran besar struktur organisasi PT Mayora Indah Tbk plant Jayanti 1.



Gambar 3.2 Struktur umum PT Mayora Indah Tbk plant Jayanti 1.

Berikut gambaran struktur organisasi Bagian *Departemen IR-GA PT.*

Mayora Indah Tbk plant Jayanti 1.



Gambar 3.3 Struktur *Departemen IR-GA.*

3.6.5 Tugas Dan Tanggung Jawab

Departement Head IR-GA

Fungsi Dasar dari Dept. Head IR-GA adalah sebagai berikut :

- Mengatur semua kegiatan pendukung operasional pabrik.

Tugas-tugas dari Dept. Head IR-GA adalah sebagai berikut :

- Melakukan pengawasan dan pengaturan sumber daya untuk kelancaran operasional pabrik.

- Melakukan dan mengawasi pelaksanaan GMP, K3 dan 5R dipabrik.
- Menjaga hubungan baik dengan masyarakat dilingkungan pabrik dan pemerintah daerah setempat.
- Menjaga stabilitas dan keamanan pabrik.
- Melaksanakan Administrasi penggajian dan pemenuhan kebutuhan dan kesejahteraan karyawan.
- Menangani keluhan-kesah karyawan akibat hubungan kerja.

Wewenang dari Dept. Head IR-GA adalah sebagai berikut :

- Melakukan bimbingan, penilaian, dan teguran kepada seluruh karyawan.
- Memberikan sanksi atas tindak pelanggaran karyawan.
- Melakukan keputusan praktis terhadap pengamanan pabrik secara internal dan eksternal.
- Melakukan evaluasi terhadap pihak ke-3 (*vendor/supplier* yang berhubungan dengan IRGA).

Tanggung jawab dari Dept. Head IR-GA adalah sebagai berikut :

- Bertanggung jawab membantu kelancaran proses pendukung operasional pabrik.
- Bertanggung jawab atas stabilitas keamanan dilingkungan pabrik.
- Bertanggung jawab atas semua aset perusahaan, pemeliharaan aset, dan kebersihan area pabrik.

Section Head IR-GA

Fungsi Dasar dari Section Head IR-GA adalah sebagai berikut :

- Mengawasi dan mengatur fungsi-fungsi kerja yang unit berada dibawahnya.

Wewenang dari Section Head IR-GA adalah sebagai berikut :

- Melakukan pengawasan, penilaian dan pengaturan pekerjaan *Security*, UH GA, *Receptionist*, *Office Girl/Office Boy*, *Cleaning Service eksternal/ internal*, *Pest Control*, *Catering*, Mobil Operasional dan jemputan Karyawan.
- Melakukan koordinasi dengan *Section Head* Departemen terkait dalam hal kebersihan area pabrik.
- Melakukan counseling dan coaching karyawan dan membuat surat-surat yang terkait dengan pelanggaran disiplin karyawan atas persetujuan *Dept Head IR-GA*.

Tanggung Jawab dari Section Head IR-GA adalah sebagai berikut :

- Melakukan & memonitor dokumen-dokumen yang berhubungan dengan perijinan.
- Melakukan & mengurus seluruh astek karyawan dan juga pengurusan klaim Jamsostek.
- Melakukan pengawasan, penilaian dan pengaturan pekerjaan *Security*, Staff IR-GA, *Receptionist*, *Office Girl/ Office Boy*, *Cleaning Service eksternal/ internal*, *Pest Control*, *Catering*, Mobil Operasional dan Jemputan Karyawan.
- Melakukan hubungan dengan pihak-pihak di lingkungan pabrik untuk memelihara hubungan baik dengan masyarakat sekitar.
- Mewakili Personal & Dept Head IRGA dalam berhubungan dengan Instansi Pemerintah dengan persetujuan Dept. Head terlebih dahulu.

- Melakukan Monitoring terhadap kebutuhan dan penggunaan seragam karyawan dan alat-alat kerja.
- Melakukan & mengatur penggunaan ruangan dalam pelaksanaan training dan ikut serta dalam pelaksanaan ISO.
- Melaksanakan tugas-tugas lainnya yang langsung diperintahkan oleh atasan.

Unit Head Payroll

Fungsi Dasar dari *Unit Head Payroll* adalah sebagai berikut :

- Melaksanakan rencana kerja yang diarahkan oleh atasan
- Melaksanakan pengawasan secara terus-menerus

Wewenang dari *Unit Head Payroll* adalah sebagai berikut :

- Melakukan pemeriksaan terhadap kehadiran karyawan
- Melakukan registrasi *finger scan*
- Mengingatkan semua Administrasi Departemen terkait apabila menyerahkan SPL sudah ditandatangani oleh Dept.Head terkait
- Melakukan input data ke dalam sistem *payroll*
- Melakukan *terminate* karyawan dalam sistem payroll

Tanggung Jawab dari *Unit Head Payroll* adalah sebagai berikut :

- Membuat Laporan persiapan gaji an lembur karyawan bulanan
- Membuat laporan upah (daftar upah) secara komputerisasi
- Memberiakan laporan bulanan kepada manajemen
- Menjalankan tugas-tugas lainnya yang langsung diberikan oleh atasan

Admin Payroll

Fungsi Dasar dari *Admin Payroll* adalah sebagai berikut :

- Menjalankan proses administrasi penggajian

Wewenang dari *Admin Payroll* adalah sebagai berikut :

- Melakukan pemeriksaan terhadap kehadiran karyawan
- Melakukan registrasi finger scan
- Melakukan input data ke dalam sistem SAP Payroll

Tanggung Jawab dari *Admin Payroll* adalah sebagai berikut :

- Membuat Laporan persiapan gaji dan lembur karyawan harian.
- Membuat laporan upah (daftar upah) secara komputerisasi.
- Membuat slip gaji sekaligus memasukan slip ke dalam amplop gaji harian.
- Membuat laporan tanda terima gaji karyawan harian.
- Menyerahkan daftar gaji dan slip gaji.
- Membuat laporan rekapitulasi absen pihak ketiga (security, borongan, cleaning service).
- Input karyawan baru ke master gaji dan melakukan registrasi.
- Input surat izin karyawan
- Melakukan change schedule kerja karyawan
- Menjalankan tugas – tugas lainnya yang langsung diberikan oleh atasan

BAB IV

ANALISIS DAN PEMBAHASAN

4.1 Analisa Sistem Yang Sedang Berjalan

Dari hasil wawancara dan observasi yang telah dilakukan di PT. Mayora Indah Tbk Plant Jayanti, sesuai pembahasan yang diambil mengenai analisa informasi absensi karyawan *outsourcing*, Berikut adalah alur dari sistem yang berjalan saat ini :

4.1.1 Proses Absen Karyawan

Pada proses ini akan menjelaskan alur dari karyawan melakukan absen masuk sampai absen pulang dalam sehari bekerja. Berikut ini adalah alur nya :

1. Pembagian *shift* kerja karyawan

Di PT. Mayora Indah Jayanti 1 semua karyawan bekerja dalam satu minggu sebanyak 40 jam kerja sesuai dengan peraturan pemerintah, yang dimana memiliki beberapa shift, diantaranya nya :

Hari Senin- Jumat (8 jam)

- Non shift : 08:00 – 16:00
- Shift 1 : 07:00 – 15:00
- Shift 2 : 15:00 – 23:00
- Shift 3 : 23:00 – 07:00

Hari Sabtu (5 jam)

- Non shift : 08:00 – 13:00
- Shift 1 : 07:00 – 12:00

- Shift 2 : 12:00 – 17:00
- Shift 3 : 17:00 – 22:00

2. Awal Masuk karyawan

Di proses ini, diawali dari karyawan yang masuk di shift yang sudah di tentukan oleh atasan nya masing-masing, sebelum bekerja karyawan melakukan *finger* masuk di mesin-mesin *finger* yang sudah disediakan. Finger yang dilakukan harus di bawah jam *shift* yang sudah di tentukan, karena kalau karyawan *finger* di atas jadwal yang sudah di tentukan akan menyebabkan nya muncul nya “datang terlambat”

3. Jam istirahat karyawan

Setiap karyawan yang bekerja 8 jam memiliki hak untuk istirahat selama 1 jam, bagi karyawan yang kerja terkena shift jam istirahat menyesuaikan dengan kebutuhan di dalam produksi atau di bagian bekerja. Sedangkan yang bekerja nonshift jam istirahat selalu pukul 12:00. Untuk jam kerja hanya 5 jam, tidak memiliki jam istirahat.

4. Jam pulang karyawan

Di proses ini karyawan melakukan finger pulang apabila sudah melewati jam jam kerja nya. Karyawan harus melakukan finger melewati jam pulang yang di tentukan, karna apabila melakukan melakukan finger sebelum jam nya maka akan menyebabkan munculnya “Pulang cepat”

5. Jam lembur karyawan

Lembur karyawan ditentukan oleh atasan nya masing masing, lemburan terbagi dari beberapa jenis, yaitu lemburan hari biasa, lemburan full pada hari sabtu, dan lemburan hari libur atau hari minggu.

Perhitungan lembur :

Hari senin – sabtu : -1 jam pertama x 1,5 gaji normal perjam

-Jam berikutnya x 2 gaji normal perjam

Hari minggu/libur : - perjam x 2 gaji normal perjam

4.1.2 Proses Perizinan karyawan

Pada proses ini setiap karyawan *outsourcing* yang tidak masuk bekerja karena beberapa alasan seperti sakit, menikah, ada saudara meninggal, dan keperluan lain nya, harus mengurus izin kepada *payroll*, proses nya adalah sebagai berikut :

1. Karyawan membuat *form* izin

Pada proses ini karyawan yang akan melakukan izin atau mengurus izin atas ketidakhadiran nya harus mengisi *form* izin yang disediakan dari setiap yayasan *outsourcing* masing-masing , yang dimana terdapat 5 (lima) yayasan *outsourcing*, yaitu:

- PT. Reza Karya Abadi
- PT. Damarindo Mandiri
- PT. Solutama Lestari
- PT. Rusli Jaya
- PT. Mitra Raya Setiakawan

Form izin yang disediakan oleh yayasan *outsourcing* ini wajib diisi oleh karyawan sesuai dengan izin yang mereka akan atau sudah dilakukan, izin yang akan dilakukan seperti lamaran, menikah, wisuda, dan lain lain, sedangkan izin yang sudah dilakukan seperti sakit, saudara meninggal, dan lain sebagainya. Surat izin harus sudah di tanda tangani oleh *Section Head* masing-masing.

2. Approvel dari *Section Head* karyawan yang melakukan izin

Pada proses ini karyawan *outsourcing* meminta tanda tangan kepada setiap atasannya masing masing yang dimana bertujuan agar atasan mengetahui bahwa ada tim nya yang akan melakukan izin. Karyawan *outsourcing* hanya perlu memberikan *form* surat izin yang sudah lengkap dengan lampiran dari izin yang akan dilakukan.

3. Karyawan *outsourcing* memberikan surat izin ke tim *Payroll*

Di proses ini surat izin dari setiap yayasan yang sudah diisi oleh karyawan, harus di serahkan kepada tim *payroll* agar bisa dilakukan proses penginputan oleh tim *payroll*. Setiap surat izin yang diberikan, harus memiliki lampiran pendukung yang menjadi bukti dari izin yang dilakukan, contohnya :

- Menikah (Surat Undangan)
- Wisuda (Surat keterangan wisuda)
- Ujian Kuliah (Jadwal ujian)
- Sakit (Surat keterangan dokter atau surat keterangan berobat)
- Keluarga meninggal (Surat Kuning atau surat kematian)
- Dan lain sebagainya

4. Peng-input-an surat izin oleh tim *payroll*

Surat izin yang sudah diterima oleh tim *payroll* harus di tanda tangani terlebih dahulu oleh *Dept. Head IRGA* sebelum bisa di *input* oleh tim *payroll*, surat izin akan di input melalui aplikasi khusus tim *payroll* yaitu aplikasi “*puzzle*”. Setiap surat izin yang diterima harus di input selambat-lambat nya pada hari senin karena pada hari senin adalah terakhir penarikan data untuk penginputan laporan KPI.

5. Pengarsipan Surat izin yang sudah diterima

Semua surat izin yang sudah diinput oleh tim *payroll* akan diarsip berdasarkan tanggal dan bulan yang bertujuan untuk memudahkan mencari surat izin kembali ketika dibutuhkan

6. Membuat laporan mingguan

Pada proses ini tim *payroll* akan membuat laporan mingguan yang dimana laporan ini akan dikirimkan kepada tim *CI (continuous Improvement)* yang nantinya akan menjadi pencapaian KPI (*Key Performa indeks*) dari *Departement IRGA*.

4.1.3 Proses Rekapian Absensi Oleh Tim *Payroll*

Pada proses ini akan membahas tentang rekapian yang dilakukan oleh tim *payroll* mengenai absensi karyawan *outsourching* sampai ke proses penggajian karyawan, proses nya dianatara lain adalah :

1. *Upload* absen masuk dan pulang

Pada proses ini, tim *payroll* akan menarik data dari mesin *finger* dan akan mengupload data absen yang sudah di tarik tadi melalui *system* yang sudah ada. Data yang di upload adalah log absen dari *finger* masuk, istirahat dan *finger*

pulang karyawan. Data yang di *upload* nanti akan otomatis masuk kedalam system *Puzzle*. Hal ini dilakukan rutin setiap hari agar tidak ada kesalahan dalam data absensi karyawan, beberapa kesalahan yang pernah terjadi adalah hilangnya absensi karyawan atau ada absensi yang turun akibat adanya karyawan yang masuk tidak sesuai dengan *shift* nya.

2. *Closing-an* gaji karyawan *outsourcing*

PT. Mayora Indah Group disemua cabang nya menerapkan untuk *cut off* periode gaji itu dari tanggal 16-15 (contoh : 16 Mei – 15 Juni untuk gaji periode bulan Juni). Tetapi untuk karyawan *outsourcing closing-an* sampai di tanggal 20, yang dimana sebelum tanggal 20 tersebut semua absensi karyawan *outsourcing* sudah harus benar baik dari absensi masuk dan pulang, pemotongan dari izin yang dilakukan, dan pemotongan BPJS sebelum di tarik data oleh bagian *Payroll* di kantor pusat.

3. Penggajian

Pada proses penggajian ini, yang mengatur adalah tim *Payroll* pusat, yang dimana gaji karyawan akan di transfer terlebih dahulu ke rekening Yayasan *Outsourcing* sebelum dikirim ke rekening para karyawan *outsourcing*. Penggajian akan di kirim setiap akhir bulan ke rekening karyawan.

4.1.4 Kelebihan dan kekurangan *system* yang berjalan

Dari *system* yang sedang berjalan hingga saat ini, terdapat beberapa kelebihan dan kekurangan, diantaranya :

Kelebihan :

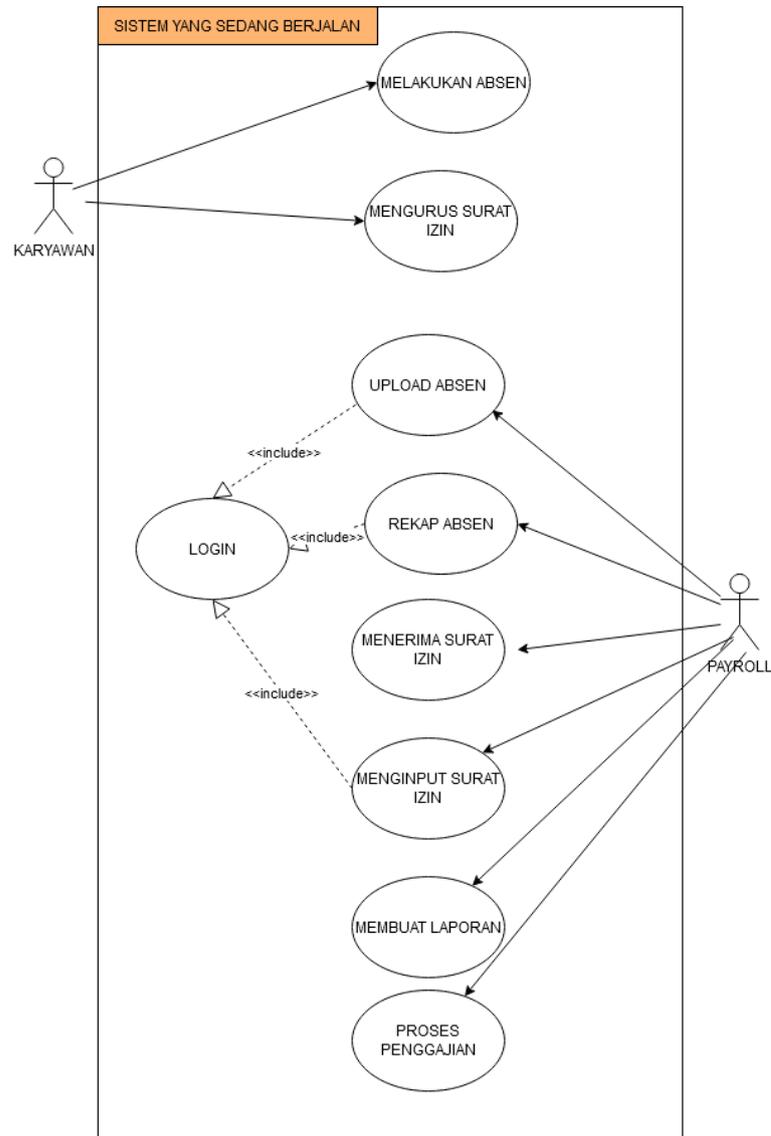
- Absen karyawan sudah menggunakan mesin finger, bukan dengan absen manual kertas
- Proses upload absensi karyawan *outsourcing* sudah bagus dengan system yang sudah ada.
- Data absen sudah muncul 1 hari setelah karyawan melakukan absen
- Penggajian lembur sudah sesuai dengan aturan yang ada dari pemerintah

Kekurangan :

- Lama nya proses karyawan mendapatkan form izin dari pihak yayasan *outsourcing*, karna jam kerja karyawan yang terbagi beberapa shift sedangkan pihak yayasan *outsourcing* tidak mengikuti shift
- Karyawan yang masih dalam proses istirahat semasa sakit tidak bisa mengurus surat izin karena belum bisa ke pabrik yang dimana ketidakhadiran harus di urus paling lambat 1 hari setelah masuk kerja.
- Penyerahan surat izin ke tim *payroll* juga terhalang oleh jam kerja karyawan
- Tempat penyimpanan surat izin yang sudah di input memakan banyak tempat karena masih menggunakan *form* yang memakai kertas

4.2 Use Case Sistem yang Sedang Berjalan

Bentuk diagram *use case* dari sistem yang berjalan saat ini digambarkan dalam Gambar 4.1 ini.



Gambar 4.1 *Use case* yang sedang berjalan

Usecase ini memiliki 2 aktor, diagram *use case* sistem dijelaskan dalam daftar *use case* pada Tabel 4.1.

Tabel 4.1 Daftar *Use Case* Sistem Berjalan

| No | Aktor | Penjelasan |
|----|----------------------|--|
| 1 | Karyawan Outsourcing | Melakukan absen Membuat dan menyerahkan surat izin |
| 2 | Payroll | Merekap absensi karyawan <i>outsourcing</i> Menerima, meng-input surat izin dan membuat laporan |

4.3 Desain Sistem UML (*Unified Modeliing Language*) yang Diusulkan

Dari *system* yang sudah berjalan, hanya bagian yang kurang maksimal saja yang akan di usulkan untuk dibuat *system* yang baru, yaitu *system* pengajuan surat izin. Dalam sistem yang akan diusulkan berikut beberapa urutan prosedur sebagai berikut:

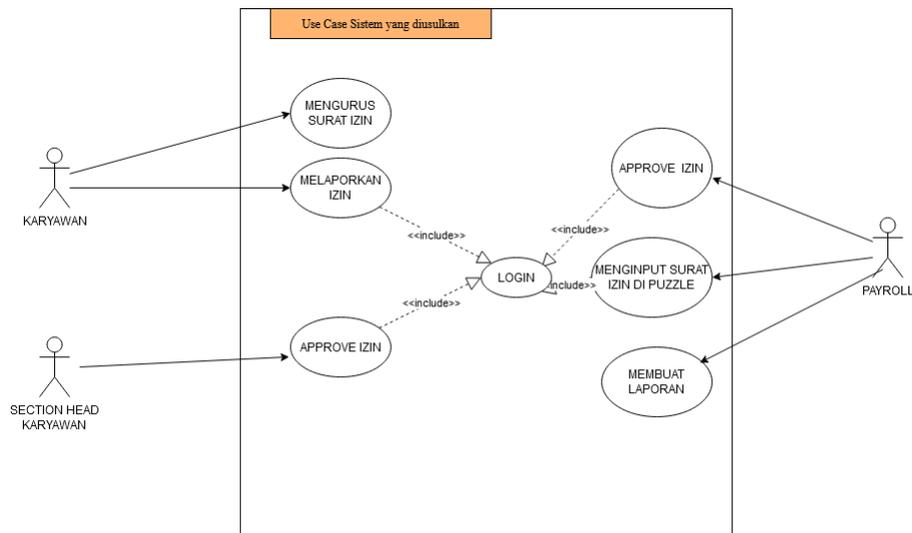
1. Pada halaman utama, karyawan yang ingin mengurus izin harus melakukan login terlebih dahulu dengan memasukan email dan password.
2. Setelah berhasil masuk kedalam *system*, karyawan bisa mengisi data sesuai dengan kebutuhan atau sesuai izin yang akan dilakukan, dan karyawan wajib mengisi data dengan lengkap dan memasukan lampiran pendukung surat izin.
3. Setelah berhasil melakukan pengajuan izin, karyawan hanya perlu menunggu hasil diterima atau tidak nya surat izin melalui email.
4. Setelah karyawan melakukan pengajuan, *Section Head* harus melakukan *approval* izin yang sudah diajukan, *Section Head* harus masuk terlebih dahulu dengan cara login ke dalam *system* dengan memasukan *email* dan *password* yang sudah di tentukan atau dibuat sebelum nya.
5. Dihalaman approve akan muncul nama-nama karyawan yang mengajukan izin, *Section head* akan melakukan *Approve* apabila menyetujui izin karyawan dan melakukan *reject* apabila tidak menyetujui izin tersebut. Data izin yang

sudah di *approve* atau pun di *reject* akan tersampaikan ke karyawan melalui email karyawan yang dimasukkan di data pengajuan izin

6. Berikut nya apabila izin yang diajukan oleh karyawan di *approve* oleh *section head* akan masuk ke data tim payroll untuk melakukan proses *approval* lebih lanjut, untuk bisa melihat data yang sudah di *approve*, payroll masuk ke *system* dengan melakukan *login* dengan memasukkan *username* dan *password*
7. Berikut nya data yang sudah di *approve* akan di cek apakah sesuai dengan ketentuan yang berlaku, apabila data sesuai maka data akan di *approve* kembali oleh admin *payroll* dan di simpan untuk proeses selanjutnya, apabila data tidak sesuai data akan di *reject* oleh admin *payroll*, informasi data yang sudah di *approve* atau pun *reject* akan diterima oleh karyawan melalui email.
8. Diproses berikutnya, admin *payroll* akan menginput surat izin yang sudah di ajukan melalui aplikasi “*puzzle*” dengan melakukan login menggunakan *username* dan *password*, dan menginput berdasarkan data pengajuan yang sudah di *approve* tadi.
9. Di akhir *week payroll* akan mengirimkan laporan jumlah kehadiran karyawan ke bagian Dept. CI, yang terdiri dari jumlah karyawan alpa, datang terlambat, izin, dan lain-lain nya.

4.3.1 Use Case yang diusulkan

Berikut ini adalah *use case* dari sistem yang akan diusulkan



Gambar 4.2 Usecase system yang diusulkan

4.3.2 Tabel deskripsi penjelasan use case yang diusulkan

Berikut ini adalah tabel penjelasan dari use case yang diusulkan dari kekurangan yang ada :

Tabel 4.2 Penjelasan use case diagram Login karyawan

| | |
|----------------|-----------------------|
| Tanggal : 2021 | Author : Aldi Prayoga |
| Versi : 1.0 | |

| | | |
|----------------------------|---|--|
| Use Case Name : | login | Type Use Case |
| Use Case Id : | 001 | <input checked="" type="checkbox"/> Persyaratan Bisnis |
| Priority : | high | |
| Primary Business Actors | Karyawan | |
| Description : | Use case ini digunakan untuk melakukan verifikasi bagi karyawan masuk kedalam system pelaporan surat izin | |
| Pre-Condition : | Karyawan melakukan login dengan memasukan username dan nik diperusahaan | |
| Typical Courses Of Events: | Actor Action | System Responce |
| | 1. Karyawan memilih halaman login untuk karyawan dan memasukan Email dan password perusahaan pada sistem | 2. Menampilkan menu login |
| Alternate Courses: | 3. Jika emial dan password karyawan salah system | |

| | |
|-----------------------|---|
| | akan menampilkan kesalahan |
| <i>Conclusion :</i> | <i>Use Case</i> ini selesai jika <i>actor</i> berhasil <i>login</i> dan Menu utama tampil |
| <i>Post-Condition</i> | - |

Tabel 4.3 Penjelasan *use case* diagram melaporkan surat izin

| | |
|----------------|------------------------------|
| Tanggal : 2021 | <i>Author</i> : Aldi Prayoga |
| Versi : 1.0 | |

| | | |
|--------------------------------|---|--|
| <i>Use Case Name :</i> | Melaporkan surat izin | <i>Tipe Use Case</i> |
| <i>Use Case Id :</i> | 002 | <input checked="" type="checkbox"/> Persyaratan Bisnis |
| <i>Priority :</i> | <i>High</i> | |
| <i>Primary Business Actors</i> | Karyawan | |
| <i>Description :</i> | <i>Use Case</i> ini digunakan karyawan untuk melaporkan ketidakhadirannya pada <i>system</i> yang ada | |
| <i>Pre-Condition :</i> | Karyawan akan mengajukan izin sesuai dengan ketentuan yang ada | |
| <i>Typical Courses</i> | <i>Actor Action</i> | <i>System Responce</i> |
| <i>Of Events:</i> | 1. Karyawan mengisi data pribadi serta memasukan lampiran pendukung surat izin pada sistem | 2. Menampilkan form pengisian data |
| <i>Alternate Courses:</i> | 3. Jika data yang diisi belum lengkap, <i>system</i> tidak akan mau mengirim data surat izin ke dalam <i>system</i> | |
| <i>Conclusion :</i> | <i>Use Case</i> ini selesai jika <i>actor</i> berhasil mengisi data dengan benar | |
| <i>Post-Condition</i> | - | |

Tabel 4.4 Penjelasan *use case* diagram Login Section Head

| | |
|----------------|------------------------------|
| Tanggal : 2021 | <i>Author</i> : Aldi Prayoga |
| Versi : 1.0 | |

| | | |
|--------------------------------|---|--|
| <i>Use Case Name :</i> | <i>Login</i> | <i>Tipe Use Case</i> |
| <i>Use Case Id :</i> | 003 | <input checked="" type="checkbox"/> Persyaratan Bisnis |
| <i>Priority :</i> | <i>high</i> | |
| <i>Primary Business Actors</i> | <i>Section Head</i> | |
| <i>Description :</i> | <i>Use case</i> ini digunakan untuk melakukan <i>verifikasi</i> bagi <i>Section Head</i> masuk kedalam <i>system</i> pelaporan surat izin | |

| | | |
|---------------------------|--|----------------------------------|
| <i>Pre-Condition :</i> | <i>Section Head</i> melakukan <i>login</i> dengan memasukkan <i>email</i> dan <i>password</i> | |
| <i>Typical Courses</i> | <i>Actor Action</i> | <i>System Respone</i> |
| <i>Of Events:</i> | 1. <i>Section Head</i> memilih halaman <i>login</i> untuk <i>Section Head</i> dan memasukkan <i>Email</i> dan <i>password</i> perusahaan pada sistem | 2. Menampilkan menu <i>login</i> |
| <i>Alternate Courses:</i> | 3. Jika <i>emial</i> dan <i>password Section Head</i> salah <i>system</i> akan menampilkan kesalahan | |
| <i>Conclusion :</i> | <i>Use Case</i> ini selesai jika <i>actor</i> berhasil <i>login</i> dan Menu utama tampil | |
| <i>Post-Condition</i> | - | |

Tabel 4.5 Penjelasan use case diagram Approve Section Head

| | |
|----------------|-----------------------|
| Tanggal : 2021 | Author : Aldi Prayoga |
| Versi : 1.0 | |

| | | |
|--------------------------------|---|--|
| <i>Use Case Name :</i> | Approve surat izin | <i>Tipe Use Case</i> |
| <i>Use Case Id :</i> | 004 | <input checked="" type="checkbox"/> Persyaratan Bisnis |
| <i>Priority :</i> | <i>High</i> | |
| <i>Primary Business Actors</i> | <i>Section Head</i> | |
| <i>Description :</i> | <i>Use Case</i> ini digunakan <i>Section Head</i> dari karyawan untuk <i>approve</i> surat izin | |
| <i>Pre-Condition :</i> | <i>Section head</i> melakukan <i>approval</i> pada surat izin yang diajukan oleh karyawan atau bawahannya | |
| <i>Typical Courses</i> | <i>Actor Action</i> | <i>System Respone</i> |
| <i>Of Events:</i> | 1. <i>Section Head</i> membuka data yang akan di <i>approve</i> | 2. Menampilkan halaman <i>approve</i> surat izin |
| <i>Alternate Courses:</i> | 3. Jika <i>Section Head</i> menyetujui pengajuan izin, maka bisa langsung <i>approve</i> surat izin 4. Jika <i>Section Head</i> tidak menyetujui pengajuan izin, maka bisa langsung <i>reject</i> surat izin dan memberikan note alasan di tolaknya surat izin 5. Jika sudah melakukan <i>approval</i> atau <i>reject</i> , karyawan akan menerima notifikasi melalui email masing masing | |
| <i>Conclusion :</i> | <i>Use Case</i> ini selesai jika <i>actor</i> berhasil melakukan <i>approval</i> | |
| <i>Post-Condition</i> | - | |

Tabel 4.6 Penjelasan use case diagram Login payroll

| | | |
|-------------------------|---|--|
| Tanggal : 2021 | Author : Aldi Prayoga | |
| Versi : 1.0 | | |
| Use Case Name : | <i>Login</i> | Tipe Use Case |
| Use Case Id : | 005 | <input checked="" type="checkbox"/> Persyaratan Bisnis |
| Priority : | High | |
| Primary Business Actors | <i>Payroll</i> | |
| Description : | <i>Use case ini digunakan untuk melakukan verifikasi bagi admin payroll masuk kedalam system pelaporan surat izin</i> | |
| Pre-Condition : | <i>Admin Payroll melakukan login dengan memasukan email dan password</i> | |
| Typical Courses | <i>Actor Action</i> | <i>System Respone</i> |
| Of Events: | 1. Admin <i>Payroll</i> memilih halaman login untuk admin dan memasukan <i>emial</i> dan password pada sistem | 2. Menampilkan menu <i>login</i> admin |
| Alternate Courses: | 3. Jika <i>emial</i> dan password admin salah <i>system</i> akan menampilkan kesalahan | |
| Conclusion : | <i>Use Case ini selesai jika actor berhasil login dan Menu utama tampil</i> | |
| Post-Condition | - | |

Tabel 4.7 Penjelasan use case diagram menerima surat izin

| | | |
|-------------------------|---|--|
| Tanggal : 2021 | Author : Aldi Prayoga | |
| Versi : 1.0 | | |
| Use Case Name : | Menerima surat izin | Tipe Use Case |
| Use Case Id : | 006 | <input checked="" type="checkbox"/> Persyaratan Bisnis |
| Priority : | High | |
| Primary Business Actors | <i>Payroll</i> | |
| Description : | <i>Use Case ini digunakan admin payroll untuk menerima dan mengecek surat izin</i> | |
| Pre-Condition : | <i>Admin Payroll menerima dan mengecek semua surat izin yang sudah masuk kedalam system yang sudah dikirimkan oleh karyawan</i> | |
| Typical Courses | <i>Actor Action</i> | <i>System Respone</i> |
| Of Events: | 1. Admin <i>payroll</i> membuka data surat izin yang di terima | 2. Menampilkan data surat izin yang masuk |

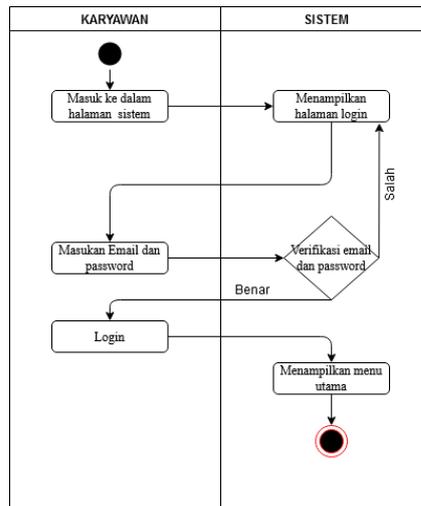
| | | |
|---------------------------|--|--|
| | 3. <i>Admin Payroll</i> mengecek lampiran dari surat izin yang ada | 4. Menampilkan data lampiran surat izin yang masuk |
| <i>Alternate Courses:</i> | 5. Jika data yang dikirim diterima, maka akan mengirimkan notifikasi data di terima ke <i>email</i> karyawan 6. Jika data yang dikirim ditolak, maka akan mengirimkan notifikasi data di tolak ke <i>email</i> karyawan | |
| <i>Conclusion :</i> | <i>Use Case</i> ini selesai jika <i>actor</i> berhasil menerima surat izin | |
| <i>Post-Condition</i> | - | |

4.3.3 Activity diagram system yang diusulkan

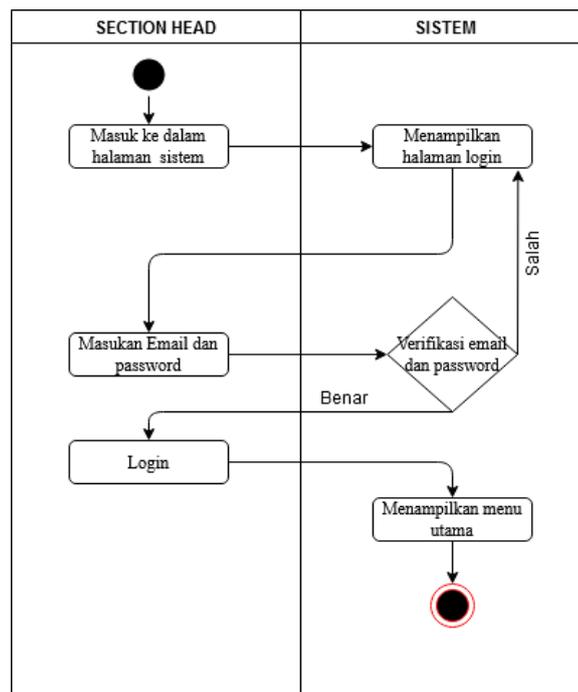
Activity Diagram (diagram aktivitas) adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow*). Dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of events*). Diagram aktivitas di bawah ini adalah penjelasan ke dalam sistem.

a. Activity diagram login

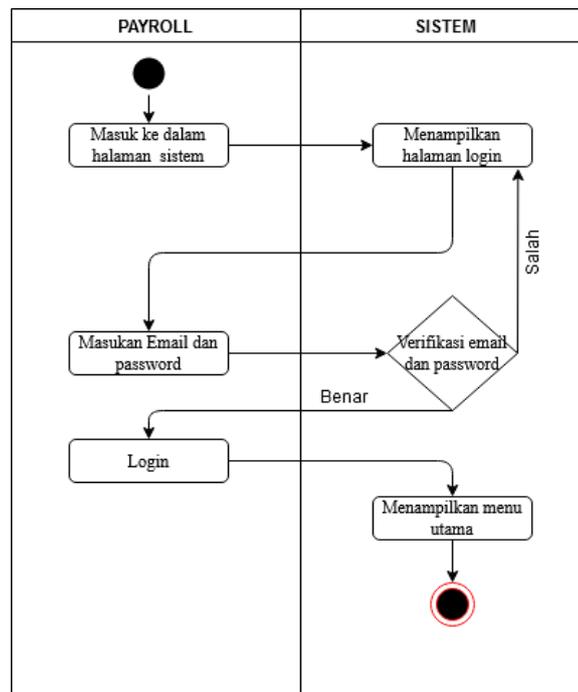
Berikut ini adalah *activity* diagram untuk *login* dari beberapa *actor* :



Gambar 4.3 Activity diagram karyawan masuk system izin



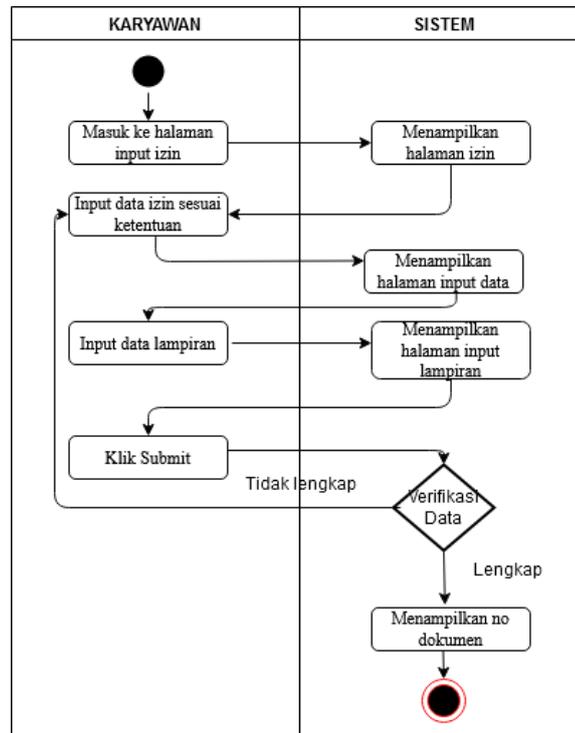
Gambar 4.4 Activity diagram login Section Head



Gambar 4.5 Activity diagram *login Payroll*

b. Activity diagram input izin karyawan

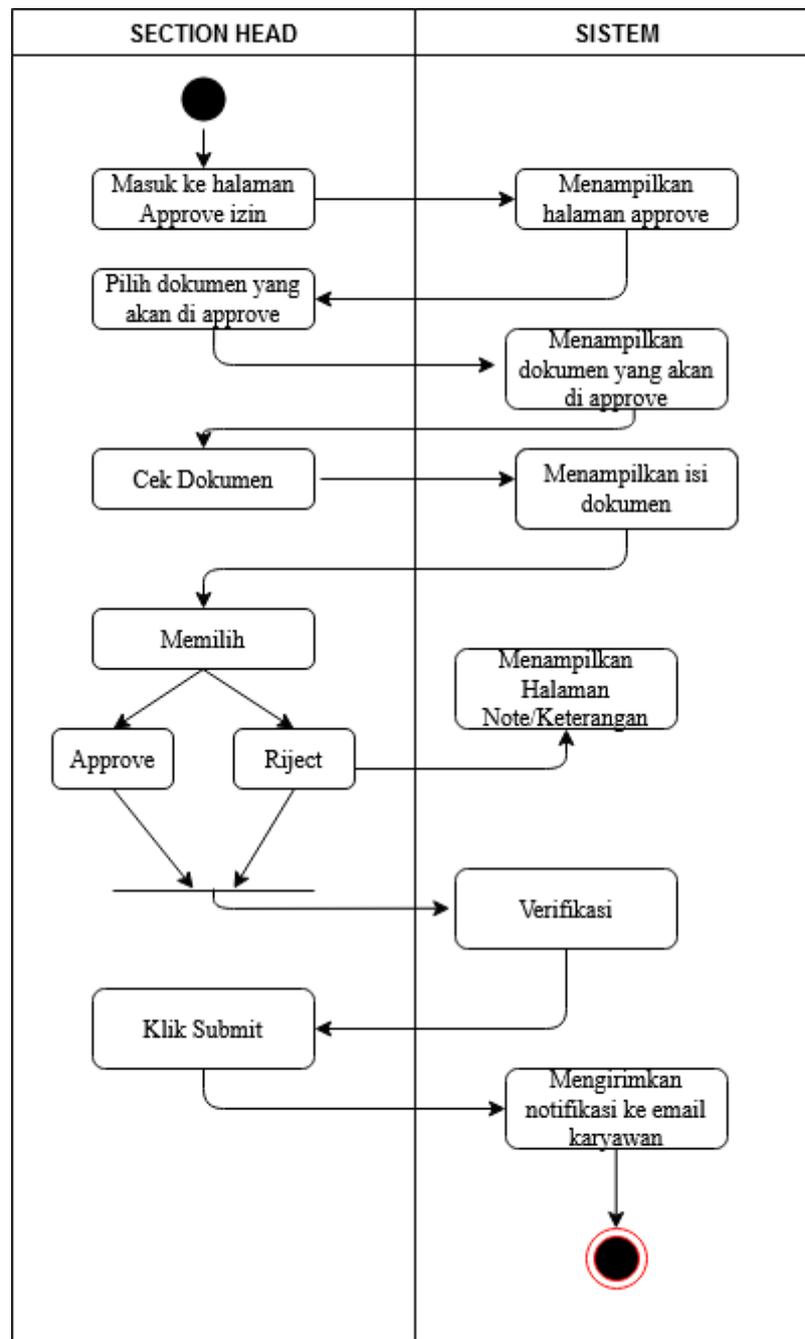
Berikut ini adalah *activity* diagram untuk *input* surat izin :



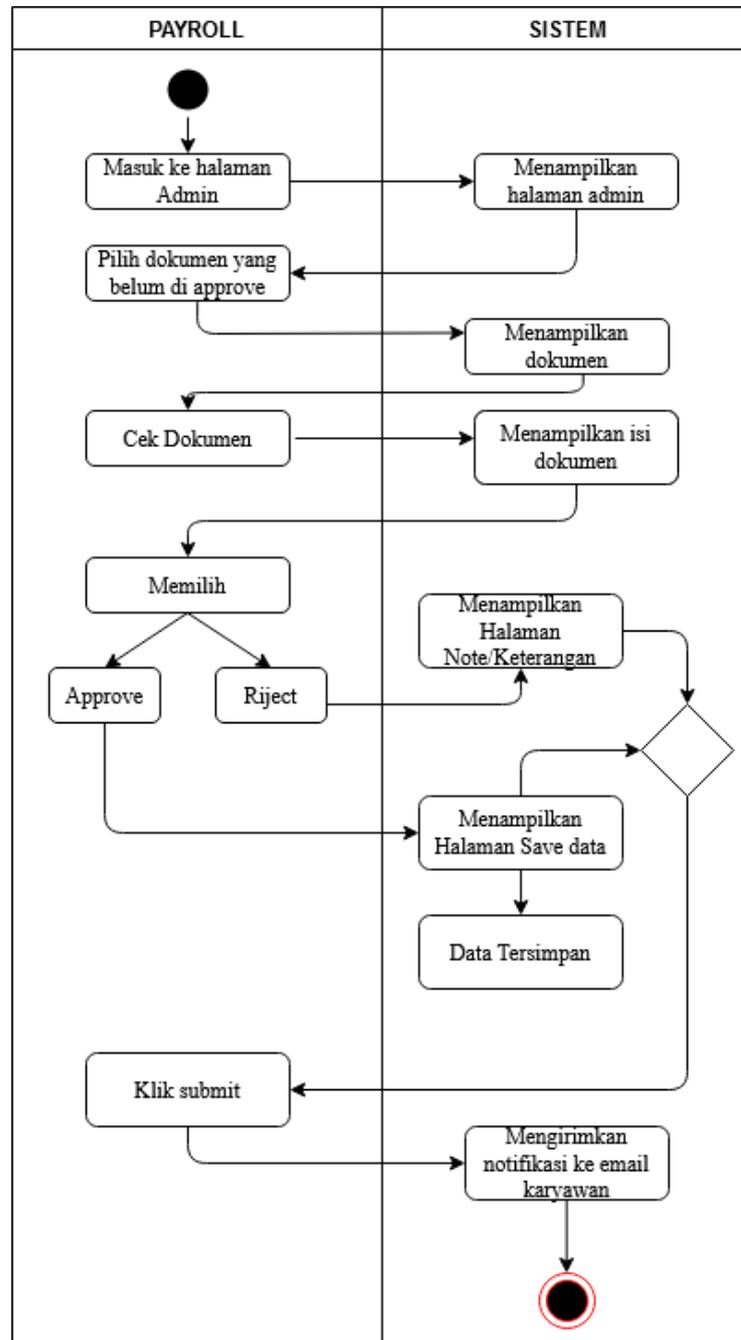
Gambar 4.6 Activity diagram *input* izin karyawan

c. Activity diagram approve izin karyawan

Berikut ini adalah *activity* diagram untuk *approve* izin karyawan oleh *Section head* dan *payroll*:



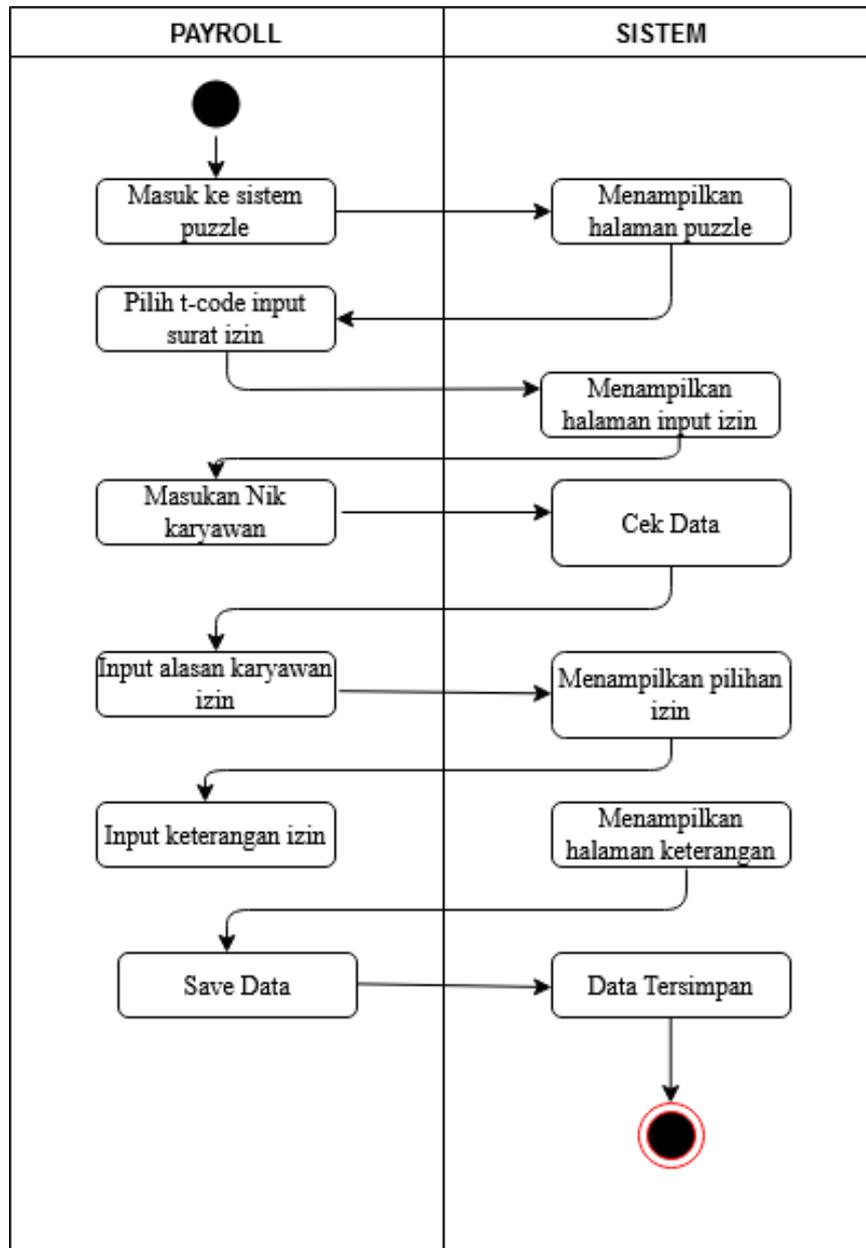
Gambar 4.7 Activity diagram *approve* izin karyawan oleh *Section Head*



Gambar 4.8 Activity diagram *approve* izin karyawan oleh *Payroll*

d. Activity diagram input izin karyawan

Berikut ini adalah *activity* diagram untuk *input* surat izin di *puzzle* oleh tim *payroll* :



Gambar 4.9 Activity diagram *input* izin karyawan di *puzzle*

4.3.4 Sequence diagram *system* yang diusulkan

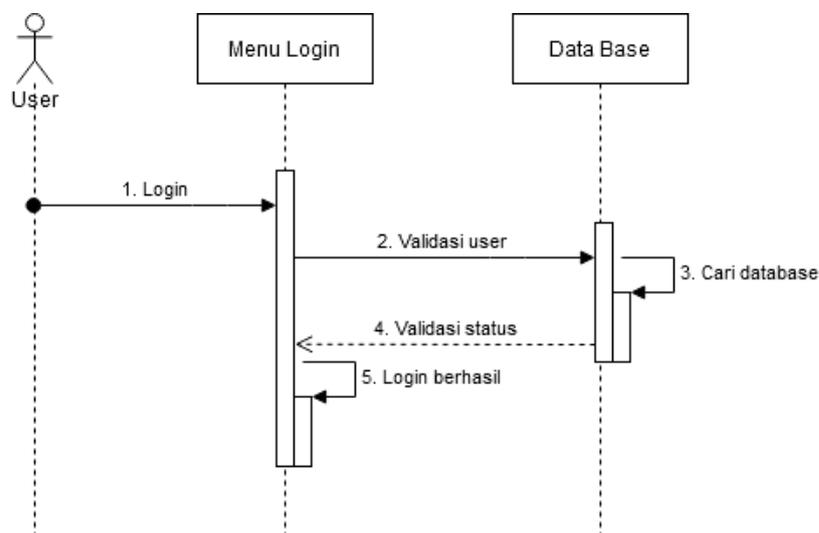
Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event

untuk menghasilkan output tertentu. Diagram ini menunjukkan sejumlah contoh objek dan *message* yang di letakkan diantara objek-objek di dalam *use case*.

Komponen utama *Sequence* diagram terdiri dari objek yang di gambarkan dengan kotak segi empat bernam. *Message* di wakili oleh garis dengan tanda panah dan waktu yang di tunjukkan dengan progress *vertical*. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

a. *Sequense* diagram *login*

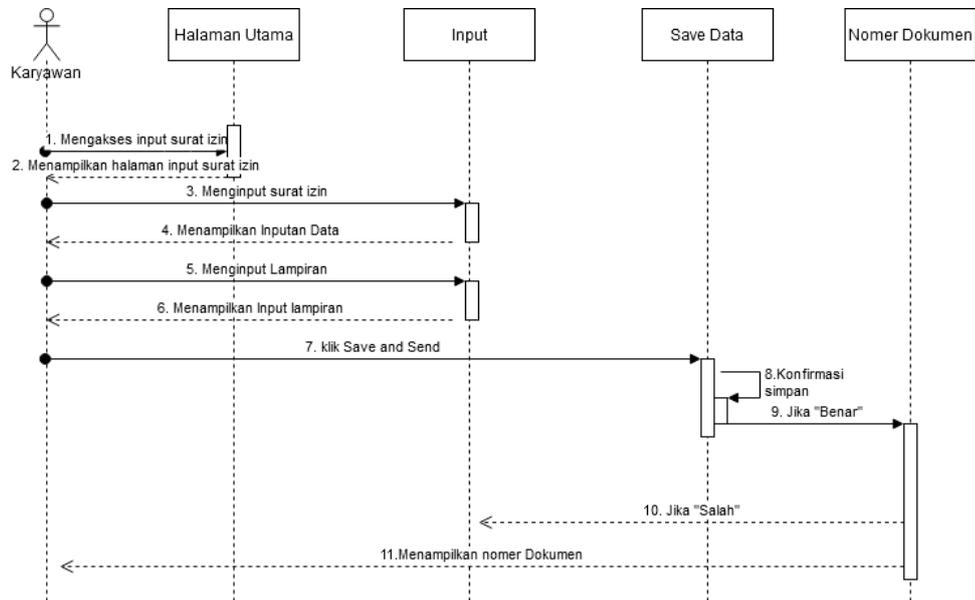
Berikut ini adalah *sequence* diagram *login* yang akan diusulkan :



Gambar 4.10 *Sequense* diagram *login*

b. *Sequense* diagram *input* surat izin karyawan

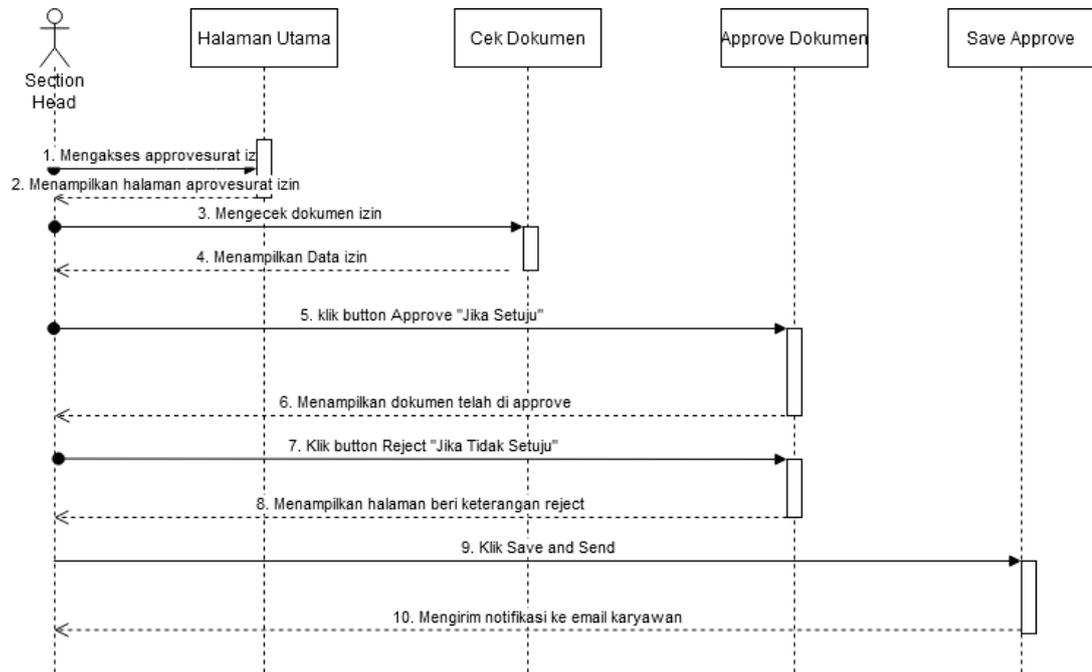
Berikut ini adalah *sequence* diagram surat izin karyawan yang akan diusulkan :



Gambar 4.11 *Sequence* diagram *input* surat izin karyawan

c. *Sequence* diagram *approve* izin oleh *Section Head*

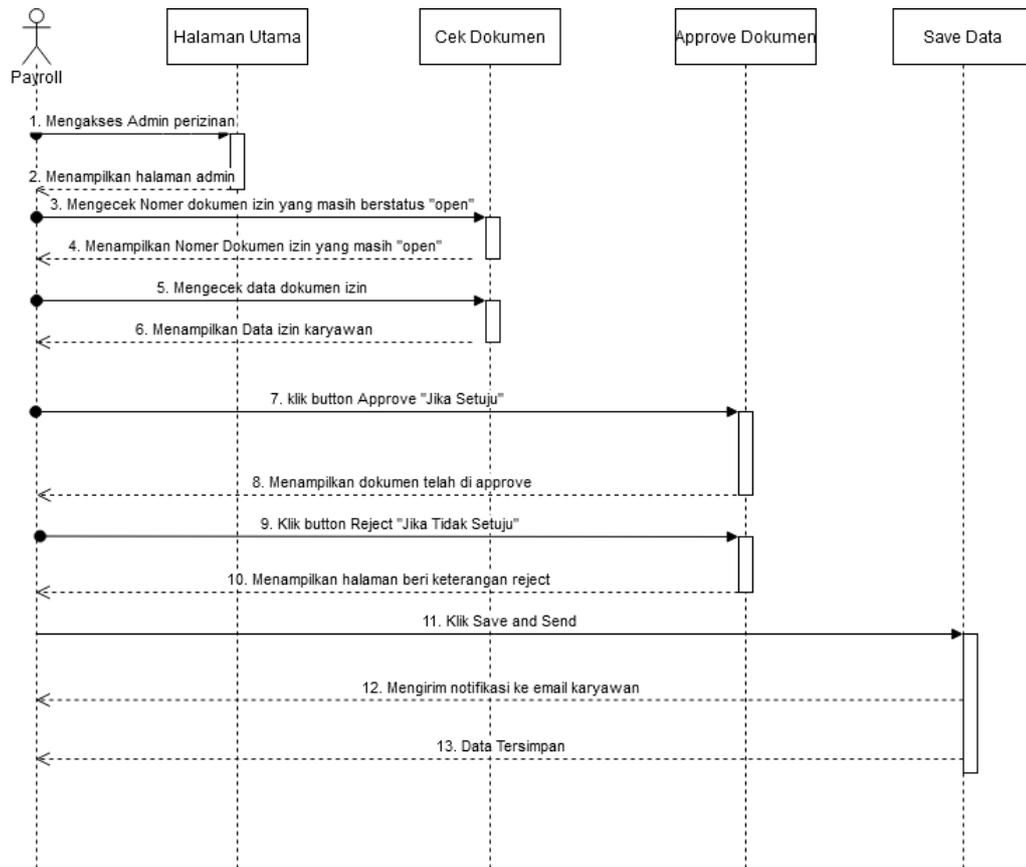
Berikut ini adalah *sequence* diagram *approve* izin oleh *Section Head* yang akan diusulkan :



Gambar 4.12 *Sequence diagram approve izin oleh Section Head*

d. *Sequence diagram approve izin oleh Payroll*

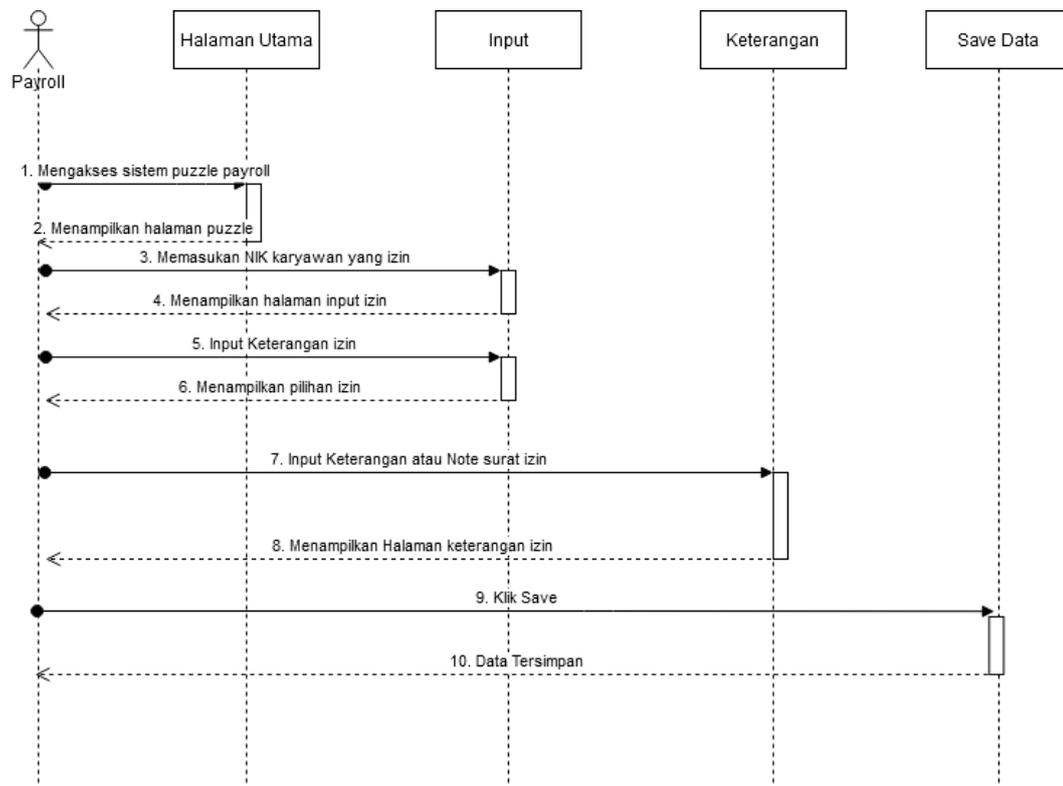
Berikut ini adalah *sequence diagram approve izin oleh Payroll* yang akan diusulkan :



Gambar 4.13 *Sequence diagram approve izin oleh Payroll*

e. *Sequence diagram input izin oleh Payroll di puzzle*

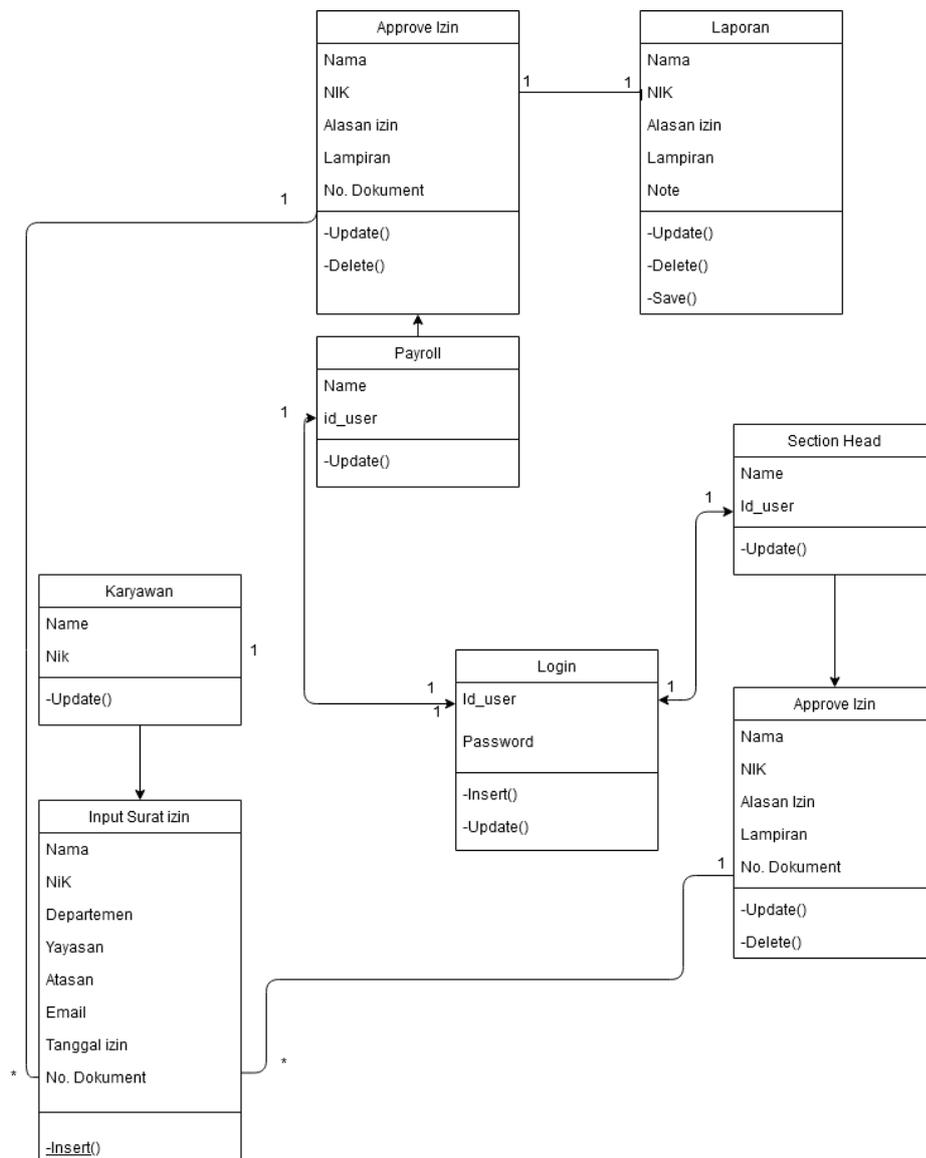
Berikut ini adalah *sequence diagram input izin oleh Payroll di puzzle* yang akan diusulkan :



Gambar 4.14 *Sequence diagram input izin oleh Payroll di puzzle*

4.3.5 *Class diagram system yang diusulkan*

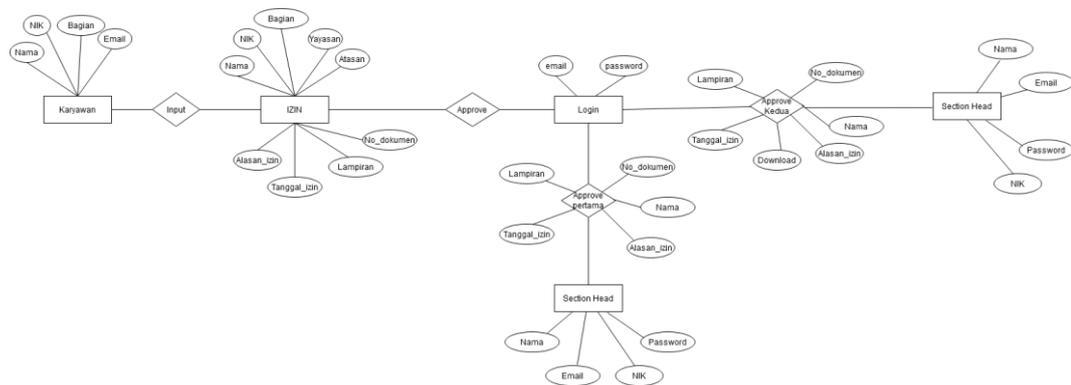
Menggambarakan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. Dengan melihat karakteristik sistem absensi karyawan *outsourcing* sudah berjalan, maka dapat dibuat *class diagram*. Berikut ini adalah penggambaran *class diagram* yang diusulkan.



Gambar 4.15 Class diagram yang di usulkan

4.4 ERD (Entity Relationship Diagram)

Berikut ini tampilan dari ERD (*Entity Relationship Diagram*):



Gambar 4.16 ERD (Entity Relationship Diagram)

4.5 Rancangan Tabel

Berikut ini adalah struktur rancangan tabel yang diusulkan :

a. Tabel Izin Karyawan

Tabel 4.8 Penjelasan Tabel Izin Karyawan

| No | Field | Tipe Data | Value | Keterangan |
|----|--------------------|-----------|-------|-------------|
| 1 | Id | int | 11 | Primary Key |
| 2 | Nama_karyawan | var | 255 | |
| 3 | NIK | int | 10 | |
| 4 | Bagian | var | 255 | |
| 5 | Yayasan | var | 255 | |
| 6 | Atasan | var | 255 | |
| 7 | Alasan_izin | var | 255 | |
| 8 | Tanggal_awal_izin | Date | | |
| 9 | Tanggal_Akhir_izin | Date | | |
| 10 | Keterangan | var | 255 | |
| 11 | Lampiran | var | 255 | |
| 12 | No_dokumen | var | 255 | |

b. Tabel *Approve* Section Head

Tabel 4.9 Penjelasan Tabel *Approve Section Head*

| No | Field | Tipe Data | Value | Keterangan |
|----|--------------------|-----------|-------|-------------|
| 1 | Id | int | 11 | Primary Key |
| 2 | Nama_karyawan | var | 255 | |
| 3 | NIK | int | 10 | |
| 4 | Alasan_izin | var | 255 | |
| 5 | Tanggal_awal_izin | Date | | |
| 6 | Tanggal_Akhir_izin | Date | | |
| 7 | No_dokumen | var | 255 | |

c. Tabel *Approve Payroll*

Tabel 4.10 Penjelasan Tabel *Approve Payroll*

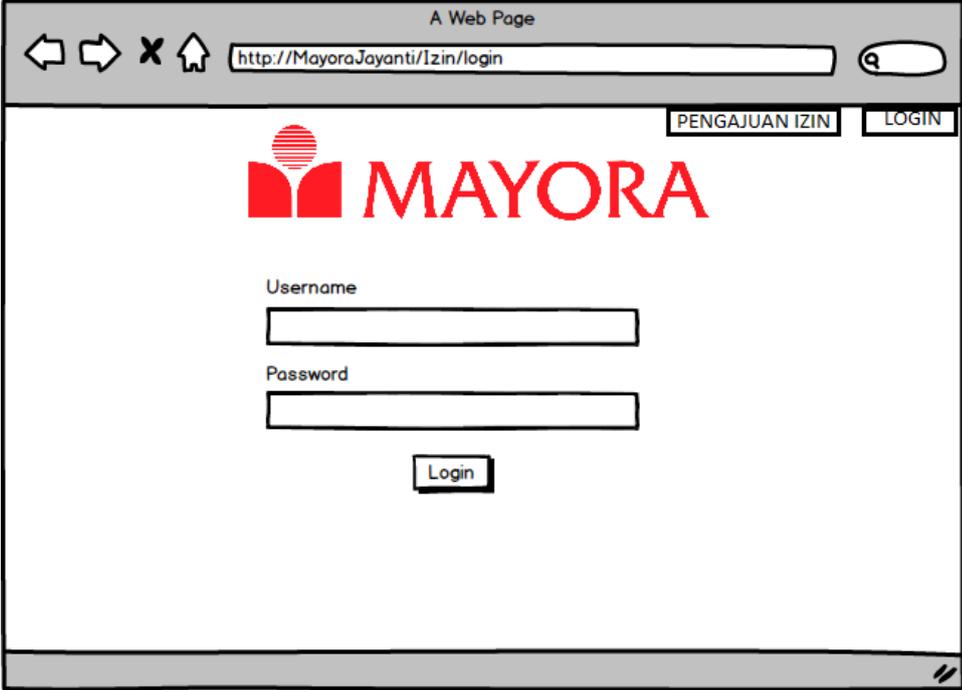
| No | Field | Tipe Data | Value | Keterangan |
|----|--------------------|-----------|-------|-------------|
| 1 | Id | int | 11 | Primary Key |
| 2 | Nama_karyawan | var | 255 | |
| 3 | NIK | int | 10 | |
| 4 | Alasan_izin | var | 255 | |
| 5 | Tanggal_awal_izin | Date | | |
| 6 | Tanggal_Akhir_izin | Date | | |
| 7 | No_dokumen | var | | |

4.6 Perancangan Antar Muka

Tujuan utama dari perancangan antar muka agar interface yang terbentuk bersifat komunikatif dan mudah digunakan oleh pemakai sehingga tujuan dari pembuatan aplikasi tercapai. Perancangan antar muka aplikasi sebagai berikut:

a. Tampilan menu *login*

Berikut ini adalah tampilan menu *login* yang akan diusulkan :



A Web Page

http://MayoraJayanti/Izin/login

PENGAJUAN IZIN LOGIN

MAYORA

Username

Password

Login

Gambar 4.17 Tampilan Menu *Login*

- b. Tampilan menu *Input* Surat izin karyawan

Berikut ini adalah tampilan *Input* Surat izin karyawan yang akan diusulkan :

A Web Page

http://MayoraJayanti/Izin/Pengajuan

Name

Employee id

Departement

Outsourching

Section Head

Email

Unattendance Type

Start date

Reason

Attachment

*max size : 2mb, pdf only

No. Dokument

Gambar 4.18 Tampilan *Input* Surat izin karyawan

c. Tampilan Halaman *Approve* oleh *Section Head*

Berikut ini adalah tampilan halaman *Approve* kedua oleh *Section Head* yang akan diusulkan :

Gambar 4.19 Halaman *Approve* oleh *Section Head*

d. Tampilan Halaman *Approve* oleh *Payroll*

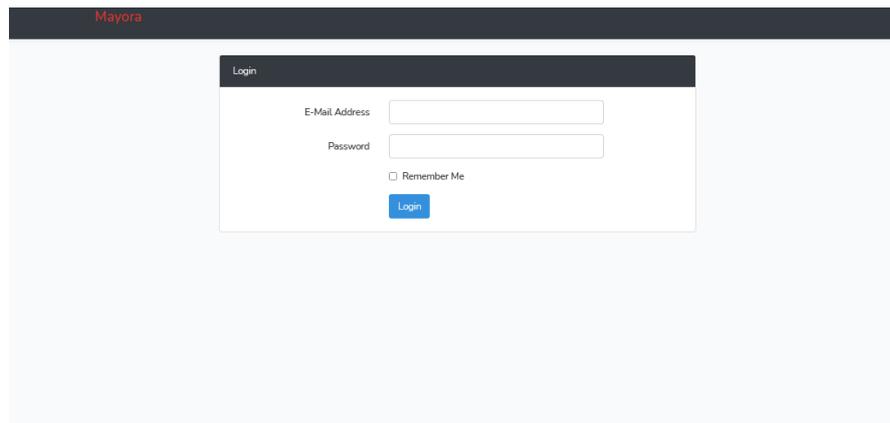
Berikut ini adalah tampilan halaman *Approve* kedua oleh *Payroll* yang akan diusulkan :

Gambar 4.20 Halaman *Approve* oleh *Payroll*

4.7 Implementasi Antar Muka dan Pemrograman

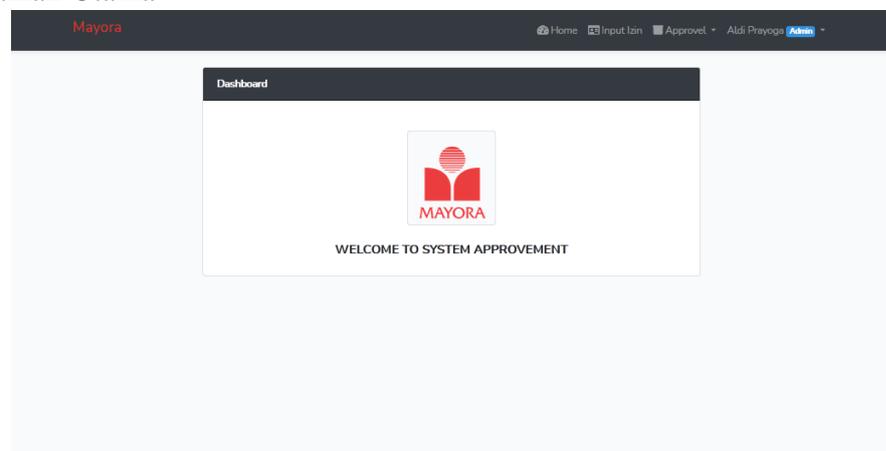
Berikut ini adalah implementasi dari pemrograman yang dibuat :

1. Halaman Login



Gambar 4.21 Halaman Login

2. Halaman Utama



Gambar 4.22 Halamann Utama

3. Halaman *Input Izin*

Mayora Home / Karyawan / Create

Karyawan - Create [Back](#)

NIK:

Nama Karyawan:

Bagian:

Yayasan:

Atasan:

Alasan Izin:

Tanggal Awal:

Tanggal Akhir:

Keterangan:

Upload File:

Gambar 4.23 Halaman *Input Izin*

4. Halaman Approve Section Head

Mayora Home / Approve SH / Index

Approve SH - Index

| NO | NO.DOC | NIK | Nama Karyawan | Alasan Izin | Tanggal Awal | Tanggal Akhir | Status | Action |
|----|---------|----------|---------------|-------------|--------------|---------------|--|---|
| 1 | 06/MYAN | 50084729 | Aldi Prayoga | SAKIT | 2021-10-28 | 2021-10-28 | <input type="button" value="Success"/> | <input type="button" value="Show"/> <input type="button" value="Delete"/> |

Gambar 4.24 Halaman *Approve Section Head*

5. Halaman Approve Payroll

Mayora Home / Approve Payroll / Index

Approve Payroll - Index

| NO | NO.DOC | NIK | Nama Karyawan | Alasan Izin | Tanggal Awal | Tanggal Akhir | Status | Action |
|----|---------|----------|---------------|-------------|--------------|---------------|--|-------------------------------------|
| 1 | 06/MYAN | 50084729 | Aldi Prayoga | SAKIT | 2021-10-28 | 2021-10-28 | <input type="button" value="Success"/> | <input type="button" value="Show"/> |

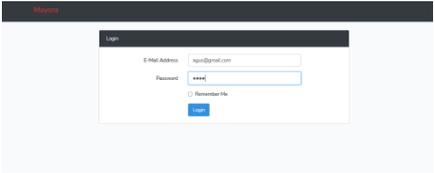
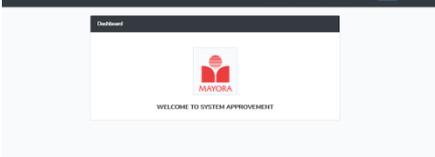
Gambar 4.25 Halaman *Approve Payroll*

4.8 Pengujian Sistem Black Box Testing

Black box testing adalah pengujian yang dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Dianalogikan seperti melihat sebuah kotak hitam, yang hanya bisa dilihat penampilannya saja tanpa mengetahui ada apa dibalik kotaknya. Pengujian *black box* mengevaluasi hanya dari tampilan luarnya (*interface*-nya) dan fungsionalitasnya tanpa mengetahui apa yang sesungguhnya yang terjadi dalam proses detailnya (hanya mengetahui input dan output). Untuk contoh pengujian terhadap beberapa tingkatan dari aplikasi berikut adalah hasil *black box* testing:

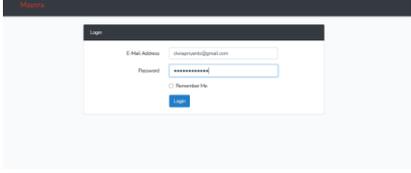
1. Pengujian Black Box Testing pada menu login Karyawan.

Tabel 4.11 Tampilan Black Box Testing menu *login* Karyawan

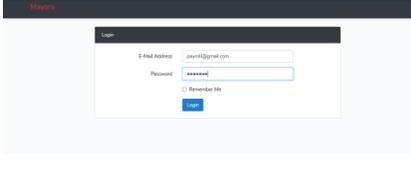
| No. | Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Kesimpulan |
|-----|---------------------------------|---|--|------------|
| 1 | <i>Login</i> aplikasi | Aplikasi menampilkan <i>form login</i> | Menampilkan sebuah <i>form input email</i> dan <i>Password</i>  | Valid |
| 2 | <i>Email dan Password</i> benar | Aplikasi akan menampilkan halaman <i>menu utama</i> | Aplikasi menampilkan halaman menu utama  | Valid |

2. Pengujian Black Box Testing pada menu *login Section Head*.

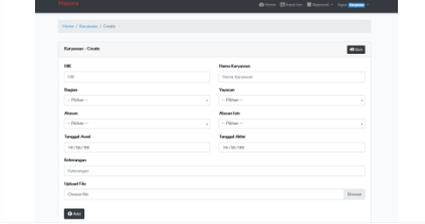
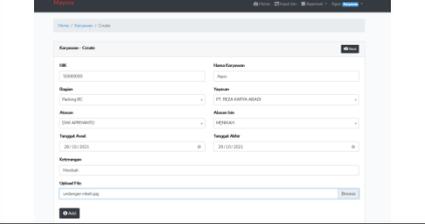
Tabel 4.12 Tampilan Black Box Testing menu *login Section Head*

| No. | Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Kesimpulan |
|-----|---------------------------------|---|---|------------|
| 1 | <i>Login aplikasi</i> | Aplikasi menampilkan <i>form login</i> | Menampilkan sebuah <i>form input email dan Password</i>  | Valid |
| 2 | <i>Email dan Password benar</i> | Aplikasi akan menampilkan halaman <i>menu utama</i> | Aplikasi menampilkan halaman <i>menu utama</i>  | Valid |

3. Pengujian Black Box Testing pada menu *login Payroll*.Tabel 4.13 Tampilan Black Box Testing menu *login Payroll*.

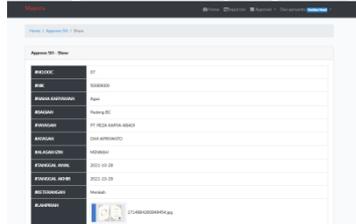
| No. | Pengujian | Hasil yang Diharapkan | Hasil Pengujian | Kesimpulan |
|-----|---------------------------------|---|---|------------|
| 1 | <i>Login aplikasi</i> | Aplikasi menampilkan <i>form login</i> | Menampilkan sebuah <i>form input email dan Password</i>  | Valid |
| 2 | <i>Email dan Password benar</i> | Aplikasi akan menampilkan halaman <i>menu utama</i> | Aplikasi menampilkan halaman <i>menu utama</i>  | Valid |

4. Pengujian Black Box Testing pada menu *Input Izin*.Tabel 4.14 Tampilan Black Box Testing menu *input Izin*

| No. | Pengujian | Hasil Yang Diharapkan | Hasil Pengujian | Kesimpulan |
|-----|---------------------------------|--|--|------------|
| 1 | Menu <i>input izin</i> | Aplikasi menampilkan input izin | Menampilkan <i>input izin</i>  | Valid |
| 2 | <i>Button create input izin</i> | Aplikasi menampilkan <i>form create input izin</i> | Menampilkan <i>form create input izin</i>  | Valid |
| 3 | Mengisi <i>form izin</i> | Aplikasi Mengisi form izin | Menampilkan pengisian form izin.  | Valid |
| 4 | <i>Button Add</i> | Aplikasi menampilkan hasil inputan | Menampilkan form hasil inputan  | Valid |

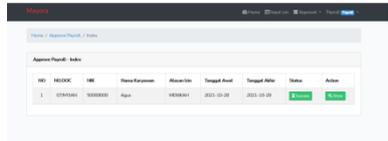
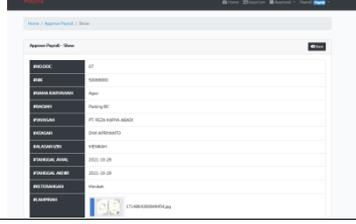
5. Pengujian Black Box Testing pada menu *Approve Section Head*.Tabel 4.15 Tampilan Black Box Testing menu *Approve Section Head*.

| No. | Pengujian | Hasil Yang Diharapkan | Hasil Pengujian | Kesimpulan |
|-----|-----------|-----------------------|-----------------|------------|
|-----|-----------|-----------------------|-----------------|------------|

| | | | | |
|---|----------------------------------|--|---|-------|
| 1 | Menu <i>Approve Section Head</i> | Aplikasi menampilkan <i>Approve Section Head</i> | Menampilkan <i>Approve Section Head</i>  | Valid |
| 2 | <i>Button Show</i> | Aplikasi menampilkan data yang di input | Menampilkan data yang di input  | Valid |

6. Pengujian Black Box Testing pada menu *Approve Payroll*.

Tabel 4.16 Tampilan Black Box Testing menu *Approve Payroll*.

| No. | Pengujian | Hasil Yang Diharapkan | Hasil Pengujian | Kesimpulan |
|-----|-----------------------------|---|---|------------|
| 1 | Menu <i>Approve Payroll</i> | Aplikasi menampilkan <i>Approve Payroll</i> | Menampilkan <i>Approve Section Head</i>  | Valid |
| 2 | <i>Button Show</i> | Aplikasi menampilkan data yang di input | Menampilkan data yang di input  | Valid |

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan uraian pada penelitian yang sudah dilakukan di PT. Mayora Indah Tbk. Jayanti 1, dapat ditarik kesimpulan sebagai akhir dari pembahasan kerja praktek ini sebagai berikut:

1. Sistem absensi karyawan *outsourcing* sudah bagus karena sudah menggunakan mesin finger scan, tidak memakai absensi manual. Proses upload dan penggajian juga sudah berjalan dengan efisien.
2. Sistem pelaporan izin karyawan *outsourcing* di PT. Mayora Indah Tbk. Jayanti 1 belum sepenuhnya menggunakan *system* yang benar, khususnya pada proses pengajuan dan penyerahan surat izin kepada tim *payroll* yang masih *manual*. Pengajuan yang dilakukan dengan menyerahkan kertas *form manual* yang mana kurang efisien serta berjalan dengan lama karena kurangnya akses yang ada. Seharusnya disediakan *system* yang lebih memudahkan karyawan melakukan pengajuan surat izin baik dari akses mengajukan dan akses untuk menyerahkan surat izin.
3. Kendala yang ditemukan di *system* pelaporan izin karyawan *outsourcing* di PT. Mayora Indah Tbk. Jayanti 1 adalah sebagai berikut :
 - Pengajuan surat izin masih memakai kertas *form manual*
 - Karyawan yang masih berhalangan hadir belum bisa melaporkan surat izin

- Jadwal kerja yang terbagi menjadi 3 shift menyebabkan lamanya proses pelaporan izin karyawan

5.2 Saran

Berdasarkan dari proses analisa maka penulis menguraikan saran sebagai berikut:

1. Sistem pelaporan izin karyawan *outsourcing* di PT. Mayora Indah Tbk. Jayanti 1 diharapkan menjadi sistem yang terintegrasi sehingga memudahkan karyawan untuk melaporkan ketidakhadirannya dan memudahkan juga tim *payroll* untuk membuat laporan.
2. Kedepannya perusahaan diharapkan mengembangkan *system* pelaporan izin karyawan *outsourcing* yang berbasis web atau android sehingga lebih efisien.

DAFTAR PUSTAKA

- Al Husain, Abdul Haqy Aji Prastian, Andre Ramadhan (2017). Perancangan Sistem Absensi Online Menggunakan Android Guna Mempercepat Proses Kehadiran Karyawan Pada PT. Sintech Berkah Abadi. *Technomedia Journal*, Vol.2 No.1. E-ISSN: 2528 – 6544.
- Albertus L. Setyabudhi (2017) Perancangan Sistem Informasi Pengolahan Data Absensi dan Pengambilan Surat Cuti Kerja Berbasis Web. *Jurnal Responsive*. Vol. 1 No.1. ISSN : 2614-7602.
- Achmad Rifai, Yasinta Prabawati Yuniar (2019). Penerapan Metode *waterfall* Dalam Perancangan Sistem Informasi Ujian Pada SMK Indonesia Global Berbasis Web. *Jurnal Khatulistiwa Informatika*. VOL. VII, NO. 1. ISSN: 2339-192.
- Hakim, L (2020). Konsep dan Implementasi Pemrograman LARAVEL 7 : Penerbit Informatika.
- Lukman Arif Sanjani, Sulis Janu Hartati, Pantjawati Sudarmaningtyas (2014). Rancang Bangun Sistem Penggajian Pegawai Dan Remunerasi Jasa Medis Pada Rumah Sakit Bedah Surabaya. *Jurnal system informasi* Vol 3, No 1 (2014) ISSN: 2338-137X.
- Pengertian dan definisi system informasi ((Rusdiana & Irfan, 2014: 76). *From website* <https://www.digilib.uinsgd.ac.id>.
- Pengertian dan definisi dari UML (Rendi Juliarto, 2021). *From website* <https://www.dicoding.com/blog/apa-itu-uml/>
- Qurotul Aini, Yuliana Isma Graha, Siti Ria Zuliana (2017). Penerapan Absensi QRCode Mahasiswa Bimbingan Belajar pada Website berbasis YII Framework. *Jurnal Ilmiah SISFOTENIK*. Vol. 7, No. 2, Juli 2017.
- Subiantoro, Sardiarinto (2018). PERANCANGAN SISTEM ABSENSI PEGAWAI BERBASIS WEB Studi Kasus : Kantor Kecamatan Purwodadi. *Jurnal SWABUMI*. Vol.6 No.2. E-ISSN: 2549-5178.
- Sejarah, dan profil lengkap dari PT. Mayora Indah, 2018. *From website* <https://www.mayoraindah.co.id/content/Riwayat-Singkat-Perusahaan-33>

LAMPIRAN

6. Contoh surat izin

| | |
|---|--|
|  | PT. REZA KARYA ABADI Jl. Raya Serang KM 13.8 No.35 RT. 002/01 Kel. Pasir Gadung Kec. Cikupa Tangerang - Banten Telp. 021 - 5940 4912, 0828 1705 4167, 0828 1705 4161 NO : |
| | FORMULIR PERMOHONAN IJIN Yang bertanda tangan di bawah ini: |
| Nama : <u>Agus Daryna</u> Nik/Bagian : <u>logoo</u> | |
| Mengajukan permohonan ijin : | |
| <input type="checkbox"/> Tidak Masuk Kerja (Dispensasi) <input type="checkbox"/> Datang Terlambat <input type="checkbox"/> Pulang Cepat <input type="checkbox"/> Keluar Lingkungan Pabrik <input checked="" type="checkbox"/> Tidak Masuk Kerja, Tanpa Dibayar Upah (Beri tanda x pada kolom yang sesuai dengan kebutuhan) | |
| Untuktanggal : <u>21 agustus 2021</u> | |
| Shift : <u>3</u> | |
| Alasan : <u>menunggu anak berdarah (sakit)</u> | |
| Tangerang, <u>25/08</u> 20 <u>21</u> | |
| Menyetujui, <u>[Signature]</u> UH/SH/DH : SPV.OS | Mengetahui, <u>[Signature]</u> IRGA/FM |
| Pemohon, <u>[Signature]</u> Kary, Ybs | |
| Note : | |

| | |
|--|--|
| Yang bertanda tangan dibawah ini menerangkan bahwa : Nama : <u>Agus Daryna</u> Umur : <u>21</u> Pekerjaan : <u>Logoo</u> Alamat : <u>Jl. Raya Serang KM 13.8 No.35 RT. 002/01 Kel. Pasir Gadung Kec. Cikupa Tangerang - Banten</u> Benar nama tersebut diatas, dibuktikan tsitrah/cuti hamil selama <u>30</u> hari Dari tanggal <u>21/08/2021</u> s/d tanggal <u>25/08/2021</u> Demikian agar yang berkeperntingan maklum. Dik: Purwana Tangerang, <u>25/08/2021</u> Dokter <u>[Signature]</u> | KLINIK GAMMA Jl. Raya Kresak No. 03, Balaraja Tangerang, Telp. (021) 595 0660. SURAT KETERANGAN SAKIT |
|--|--|

| | |
|--|---|
|  | PT. DAMARINDO MANDIRI OUTSOURCING SERVICES Sakira Plaza Blok A No. 06, Jl. Raya Gatosi Subotjo Km. 4.5 Jatiwangi, Tangerang - Banten 15134 Telp. 021 - 59638088 Fax 021-59698089 |
| FORMULIR PERMOHONAN IJIN | SEKOLAH TINGGI KEGURUAN DAN ILMU PENDIDIKAN BANTEN Jl. Ciruas walantaka KM.01 kp. Citerep ds. Kiara kec. Walantaka serang Banten Serang, 4 Agustus 2021 |
| Nama : <u>NUR DILA</u> Nik/Bagian : <u>50100525 / Packing Nrc U</u> | |
| Mengajukan Permohonan Ijin | |
| <input type="checkbox"/> Tidak Masuk Kerja (Dispensasi) <input type="checkbox"/> Datang Terlambat <input type="checkbox"/> Pulang Cepat <input type="checkbox"/> Keluar Lingkungan Pabrik <input checked="" type="checkbox"/> Tidak Masuk Kerja, Tanpa dibayar Upah (beri tanda x pada kolom yang sesuai kebutuhan) | |
| Untuk Tanggal : <u>04 Agustus 2021</u> | |
| Shift : <u>2</u> | |
| Alasan : <u>Pengumpulan Tugas Akhir</u> | |
| Tangerang, <u>05 Agustus</u> 20 <u>21</u> | |
| Menyetujui, <u>[Signature]</u> UH/SH/DH : IRGA/FM | Mengetahui, <u>[Signature]</u> Pengarah DM |
| Pemohon, <u>[Signature]</u> Kary. YBS | |
| Note : | |

| |
|--|
| Nomor : 340/AKADEMIK/STKIP.B/XI/2021 Perihal : permohonan izin Lampiran : 1 lembar Yth, kepada IRGA PT. MAYORA INDAH TBK, Dengan hormat Teriring salam dan doa semoga kita dalam beraktivitas dan menjalankan tugas sehari-hari selalu dalam lindungan Allah SWT. Sesuai dengan kebijakan dari kampus bahwa penyerahan Tugas Akhir Proposal dan KKM maka mohon kiranya mahasiswa/i yang Bernama : NURDILA PRODI : PBI (Pendidikan Bahasa Inggris) diberikan izin untuk tidak bekerja dikarenakan pengumpulan Tugas Akhir Proposal dan KKM terakhir pada Hari/Tanggal : RABU, 04 Agustus 2021 Pukul : 11:00 s/d selesai Demikian permohonan ini kami sampaikan atas perhatian dan perkenaanannya kami ucapkan terima kasih. |
|--|

7. Contoh absensi di Puzzle

Applications Places Sen Sep 6 2021 15:46 User

Puzzle

Function Navigation System Help Window

dance Transaction : Display

Company Code: GRP1 GROUP 1 Organization Unit: A300
 Personnel Area: MYR PT Mayora Indah Tbk HIERARCHY
 Payroll Area: AL03 Factory Mayora Jayanti (OS)
 Employee Group: EGKO Karyawan Outsourcing
 Employee Sub Group: KO20 KO PT. Reza Karya Abadi
 Transaction Period: 16-08-2021 To 15-09-2021
 Jumlah hari kerja: EXEC...

| Transaction Date | Employee Number | Employee Name | Work Group | Sub Work Group | Time In | Time Out | Work Day | Date In | Time In | Date Out | Time Out | Multiple Attendance |
|------------------|-----------------|---------------|------------|----------------|---------|----------|----------|------------|---------|------------|----------|---------------------|
| 16-08-2021 | 50069000 | Agus Sanjaya | GSWGSHD2 | GSSH00 | | | | 16-08-2021 | | 16-08-2021 | | |
| 17-08-2021 | 50069000 | Agus Sanjaya | 4DWGNSD1 | NSABLR01 | | | | 17-08-2021 | | 17-08-2021 | | |
| 18-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 07:00 | 15:00 | ✓ | 18-08-2021 | 06:41 | 18-08-2021 | 15:55 | |
| 19-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 07:00 | 15:00 | ✓ | 19-08-2021 | 06:53 | 19-08-2021 | 15:34 | |
| 20-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 07:00 | 15:00 | ✓ | 20-08-2021 | 06:55 | 20-08-2021 | 15:27 | |
| 21-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 07:00 | 12:00 | ✓ | 21-08-2021 | 06:55 | 21-08-2021 | 15:34 | |
| 22-08-2021 | 50069000 | Agus Sanjaya | 4DWGNSD1 | NSABLR01 | | | | 22-08-2021 | | 22-08-2021 | | |
| 23-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 23:00 | 07:00 | ✓ | 23-08-2021 | 22:50 | 24-08-2021 | 07:32 | |
| 24-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 23:00 | 07:00 | ✓ | 24-08-2021 | 22:52 | 25-08-2021 | 07:45 | |
| 25-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 23:00 | 07:00 | ✓ | 25-08-2021 | | 26-08-2021 | | |
| 26-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 23:00 | 07:00 | ✓ | 26-08-2021 | 22:53 | 27-08-2021 | 07:20 | |
| 27-08-2021 | 50069000 | Agus Sanjaya | GSWGSHD1 | GSSH06 | 17:00 | 22:00 | ✓ | 27-08-2021 | 16:52 | 27-08-2021 | 22:45 | |
| 28-08-2021 | 50069000 | Agus Sanjaya | GSWGSHD2 | GSSH00 | | | | 28-08-2021 | | 28-08-2021 | | |
| 29-08-2021 | 50069000 | Agus Sanjaya | 4DWGNSD1 | NSABLR01 | | | | 29-08-2021 | | 29-08-2021 | | |
| 30-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 15:00 | 23:00 | ✓ | 30-08-2021 | 14:49 | 30-08-2021 | 23:17 | |
| 31-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 15:00 | 23:00 | ✓ | 31-08-2021 | 14:50 | 31-08-2021 | 23:22 | |
| 01-09-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 15:00 | 23:00 | ✓ | 01-09-2021 | 14:51 | 01-09-2021 | 23:26 | |

FRM-40401: No changes to save.
Record: 1/31

8. Contoh izin yang sudah di input

Applications Places Sen Sep 6 2021 15:46 User

Puzzle

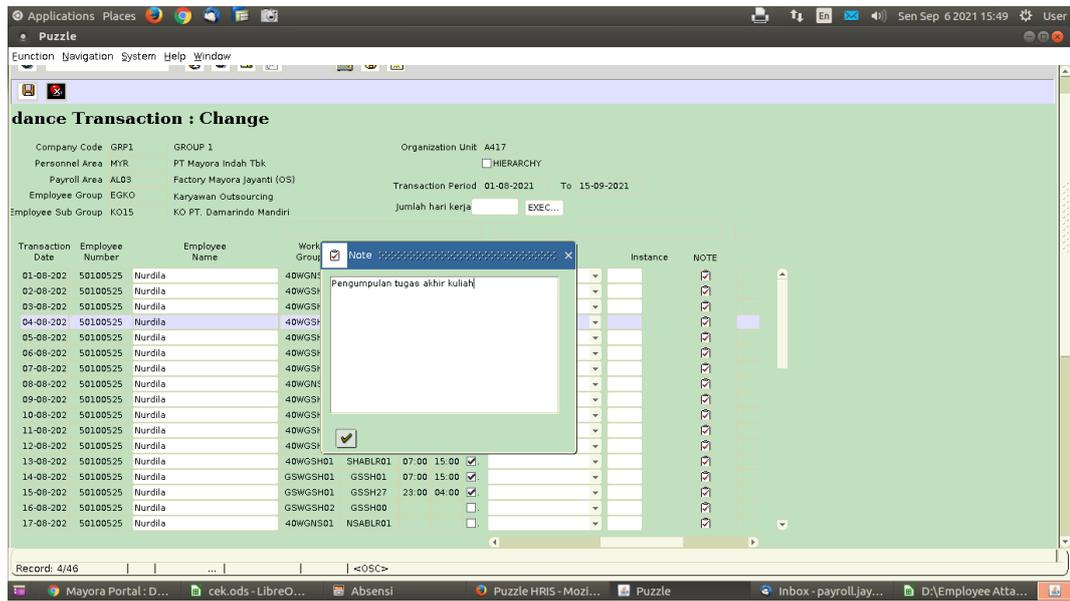
Function Navigation System Help Window

dance Transaction : Display

Company Code: GRP1 GROUP 1 Organization Unit: A300
 Personnel Area: MYR PT Mayora Indah Tbk HIERARCHY
 Payroll Area: AL03 Factory Mayora Jayanti (OS)
 Employee Group: EGKO Karyawan Outsourcing
 Employee Sub Group: KO20 KO PT. Reza Karya Abadi
 Transaction Period: 16-08-2021 To 15-09-2021
 Jumlah hari kerja: EXEC...

| Transaction Date | Employee Number | Employee Name | Work Group | Sub Work Group | Time In | Time Out | Work Day | Unattendance Type | Instance |
|------------------|-----------------|---------------|------------|----------------|---------|----------|----------|--------------------|----------|
| 16-08-2021 | 50069000 | Agus Sanjaya | GSWGSHD2 | GSSH00 | | | | | |
| 17-08-2021 | 50069000 | Agus Sanjaya | 4DWGNSD1 | NSABLR01 | | | | | |
| 18-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 07:00 | 15:00 | ✓ | | |
| 19-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 07:00 | 15:00 | ✓ | | |
| 20-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 07:00 | 15:00 | ✓ | | |
| 21-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 07:00 | 12:00 | ✓ | | |
| 22-08-2021 | 50069000 | Agus Sanjaya | 4DWGNSD1 | NSABLR01 | | | | | |
| 23-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 23:00 | 07:00 | ✓ | | |
| 24-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 23:00 | 07:00 | ✓ | | |
| 25-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 23:00 | 07:00 | ✓ | | |
| 26-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 23:00 | 07:00 | ✓ | IJIN TIDAK DIBAYAR | |
| 27-08-2021 | 50069000 | Agus Sanjaya | GSWGSHD1 | GSSH06 | 17:00 | 22:00 | ✓ | | |
| 28-08-2021 | 50069000 | Agus Sanjaya | GSWGSHD2 | GSSH00 | | | | | |
| 29-08-2021 | 50069000 | Agus Sanjaya | 4DWGNSD1 | NSABLR01 | | | | | |
| 30-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 15:00 | 23:00 | ✓ | | |
| 31-08-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 15:00 | 23:00 | ✓ | | |
| 01-09-2021 | 50069000 | Agus Sanjaya | 4DWGSHD1 | SHBBLR01 | 15:00 | 23:00 | ✓ | | |

FRM-40401: No changes to save.
Record: 1/31



9. Contoh mesin absensi karyawan

