

**LAPORAN
PRAKTIK KERJA LAPANGAN
DI**

PT. TELKOM INDONESIA

Jl. Gegerkalong Hilir No.47, Gegerkalong, Kec. Sukasari, Kota Bandung, 40141

**MEMBANGUN SISTEM ABSENSI BERBASIS *WEBSITE*
MENGUNAKAN *NEXT JS, CHAKRA UI, PRISMA ORM* DAN
POSTGRESQL SEBAGAI *DATABASE* DI PT. TELKOM INDONESIA**

Diajukan untuk memenuhi salah satu syarat kelulusan dari SMK Negeri 1 Cimahi

Disusun Oleh :

NAMA	:	FAVIAN DWI YULIANSYAH
NO. INDUK SISWA	:	201115314
TINGKAT	:	IV (EMPAT)
KOMPETENSI KEAHLIAN	:	SISTEM INFORMATIKA JARINGAN DAN APLIKASI
BIDANG KEAHLIAN	:	TEKNOLOGI INFORMASI DAN KOMUNIKASI



SEKOLAH MENENGAH KEJURUAN NEGERI 1 CIMAHI

2024

PENGESAHAN DARI PIHAK INDUSTRI

**MEMBANGUN SISTEM ABSENSI BERBASIS *WEBSITE*
MENGUNAKAN *NEXT JS, CHAKRA UI, PRISMA ORM* DAN
POSTGRESQL SEBAGAI *DATABASE* DI PT. TELKOM INDONESIA**

Laporan ini telah disetujui oleh :

Pembimbing,

PEPEN SUPRIATNA, S.T., M.M.

NIK. -

Mengetahui

Head of Engineering TCO PT. Telkom Indonesia

PEPEN SUPRIATNA, S.T., M.M.

NIK. -

PT. TELKOM INDONESIA

2024

PENGESAHAN DARI PIHAK SEKOLAH

**MEMBANGUN SISTEM ABSENSI BERBASIS *WEBSITE*
MENGUNAKAN *NEXT JS, CHAKRA UI, PRISMA ORM* DAN
POSTGRESQL SEBAGAI *DATABASE* DI PT. TELKOM INDONESIA**

Laporan ini telah disetujui oleh :

Ketua Kompetensi Keahlian

Pembimbing

ANTONI BUDIMAN, M.Pd
NIP. 19800626 201001 1 005

DENDI MOCHAMAD JABAR, S.Pd
NIP. -

Mengetahui,

Kepala SMK Negeri 1 Cimahi

AGUS PROYATMONO NUGROHO, S.Pd, M.Si.

NIP. 19670831 199003 1 003

SEKOLAH MENENGAH KEJURUAN NEGERI 1

CIMAHI

2024

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Allah SWT. karena dengan rahmat dan karunia-Nya penulis dapat menyelesaikan praktik kerja industri yang dilaksanakan di PT. Telkom Indonesia. Tak lupa sholawat serta salam selalu tercurah kepada Nabi Muhammad SAW., keluarga, para sahabat, dan umatnya hingga akhir zaman.

Adapun tujuan dari penulisan laporan ini adalah untuk memenuhi salah satu syarat kelulusan dalam menempuh pembelajaran di SMK Negeri 1 Cimahi sekaligus sebagai laporan pertanggungjawaban tertulis mengenai pelaksanaan Praktik Kerja Lapangan (PKL) di PT. Telkom Indonesia selama kurang lebih 6 (enam) bulan yang dimulai dari tanggal 3 Juli 2023 sampai dengan 31 Desember 2023.

Dalam karya tulis ini penulis mengambil judul **“MEMBANGUN SISTEM ABSENSI BERBASIS *WEBSITE* MENGGUNAKAN *NEXT JS*, *CHAKRA UI*, *PRISMA ORM* DAN *POSTGRESQL* SEBAGAI DATABASE DI PT. TELKOM INDONESIA”**.

Aplikasi yang penulis bangun merupakan aplikasi yang dapat mempermudah manajemen dalam merekap absensi pegawai dan juga meningkatkan efisiensi kerja manajemen dalam merekap absensi pegawai.

Penulis menyadari dalam penyusunan laporan ini banyak mendapat dukungan, bimbingan, bantuan dan kemudahan dari berbagai pihak sehingga laporan ini dapat terselesaikan. Dengan ketulusan hati, penulis menyampaikan ucapan terima kasih kepada :

1. Kedua orang tua dan seluruh keluarga yang senantiasa mendoakan dan memberi dukungan moral, spiritual, serta materil.
2. Bapak Agus Priyatmono Nugroho, S.Pd., M.Si., selaku Kepala Sekolah Kejuruan Negeri 1 Cimahi.
3. Bapak Antoni Budiman, M.Pd., selaku Ketua Kompetensi Keahlian Sistem Informatika Jaringan dan Aplikasi.

4. Bapak Diky Ridwan, S.Kom., selaku Wali Kelas XIII Sistem Informatika Jaringan dan Aplikasi A.
5. Bapak Dendi Mochamad Jabar, S.Pd., selaku pembimbing dari pihak sekolah sekaligus sebagai Ketua Bengkel Jurusan yang telah banyak memberikan ilmu, nasihat, dan semangat untuk kelancaran pelaksanaan PKL dan penyusunan laporan PKL ini.
6. Seluruh staf Hubungan Industri SMK Negeri 1 Cimahi.
7. Bapak Pepen Supriatna, S.T., M.M., selaku *Head of Engineering Tribe* TCO dan pembimbing dari pihak industri yang telah membina dan memberikan ilmu dan masukan selama Praktik Kerja Lapangan.
8. Bapak Gegy Faskal Nur Hamdani, selaku *Lead Frontend Tribe* TCO dari pihak industri yang telah membimbing dan memberikan ilmu dan masukan mengenai *frontend developer* selama Praktik Kerja Lapangan.
9. Seluruh staf karyawan dan rekan di Tribe TCO PT. Telkom Indonesia yang bersedia memberikan pengarahan selama Praktik Kerja Lapangan.
10. Teman-teman SIJA Angkatan 47 yang telah membantu dan memberikan dukungan serta saran kepada penulis dalam penyusunan laporan ini.
11. Semua pihak yang tidak dapat penulis sebutkan satu-persatu yang telah membantu dalam penyelesaian penulisan laporan ini.

Akhir kata penulis berharap laporan ini dapat bermanfaat dan menjadi sesuatu yang dapat menambah wawasan pengetahuan bagi penulis dan setiap orang yang membacanya.

Bandung, Februari 2024

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	x
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Tujuan	2
1.3. Pembatasan Masalah	2
1.4. Sistematika Penulisan	3
BAB II PROFIL PERUSAHAAN	5
2.1. Profil PT. Telkom Indonesia	5
2.2. Sejarah dan Perkembangan PT. Telkom Indonesia.....	5
2.3. Visi dan Misi PT. Telkom Indonesia.....	10
2.3.1. Visi PT. TELKOM.....	10
2.3.2. Misi PT. TELKOM.....	10
2.4. Logo PT. Telkom Indonesia	10
2.4.1. Keterangan logo PT. TELKOM Indonesia :	11
2.4.2. Warna–warna yang digunakan pada logo adalah :.....	11
2.5. Struktur Grup Perusahaan Telkom Indonesia.....	12
2.6. Produk dan Layanan Perusahaan	12
2.6.1. Layanan Teknologi Informasi (“TI”)	13
2.6.2. <i>E-Payment/Payment Service</i>	13
2.6.3. <i>IT enabler Services (“ITeS”)</i>	14
2.6.4. <i>Network Centric VAS</i>	14

2.6.5.	Jasa Integrasi	15
2.7.	Penghargaan Perusahaan	15
2.8.	Disiplin Kerja	16
BAB III LANDASAN TEORI.....		17
3.1.	Sistem Informasi.....	17
3.2.	Jaringan Komputer	17
3.3.	Aplikasi.....	18
3.4.	Website	18
3.4.1.	Secara Umum	18
3.4.2.	Jenis Website.....	18
3.4.3.	Fungsi Website	19
3.5.	Bahasa Pemrograman	20
3.5.1.	HTML.....	21
3.5.2.	CSS.....	21
3.5.3.	JavaScript	22
3.5.4.	TypeScript.....	22
3.6.	Framework	23
3.6.1.	React.....	23
3.6.2.	Next.JS.....	24
3.6.3.	Node.JS.....	24
3.6.4.	Node Package Manager	24
3.7.	API.....	25
3.7.1.	Jenis API	26
3.7.2.	Arsitektur API	27
3.8.	Database	28
3.8.1.	PostgreSQL	29

3.9.	<i>Code Editor</i>	30
3.10.	<i>Library</i>	31
3.10.1.	Chakra UI	31
3.10.2.	Sweetalert2	31
3.10.3.	React Chart.JS 2	32
3.11.	<i>Unified Modeling Language</i>	32
3.11.1.	Use Case Diagram	33
3.11.2.	Entity Relationship Diagram	34
3.11.3.	Flowchart	35
3.11.4.	Data Flow Diagram	36
3.12.	<i>Object Relational Mapping</i>	38
BAB IV PEMBAHASAN		40
4.1.	Analisa Sistem Absensi Pegawai	40
4.2.	Analisa Kebutuhan Alat & Perangkat	40
4.2.1.	Kebutuhan Perangkat Keras (<i>Hardware</i>)	41
4.2.2.	Kebutuhan Perangkat Lunak (<i>Software</i>)	41
4.2.3.	Kebutuhan <i>Brainware</i>	42
4.3.	Perancangan Sistem Absensi Karyawan	42
4.4.	Persiapan Pembuatan Aplikasi	55
4.5.	Pembuatan Basis Data	59
4.6.	Pembuatan Antarmuka <i>Website</i>	62
4.5.1.	Halaman Absensi	62
4.5.2.	Halaman <i>Login</i>	65
4.5.3.	Halaman <i>Dashboard</i>	66
4.5.4.	Halaman <i>Daily Summary</i>	68
4.5.5.	Halaman <i>Monthly Summary</i>	69

4.7. Pengujian <i>Website Absensi Karyawan</i>	71
BAB V PENUTUP.....	85
5.1. Kesimpulan	85
5.2. Saran	85
DAFTAR PUSTAKA	87

DAFTAR GAMBAR

Gambar 2. 1. Logo Perusahaan Telkom Indonesia	10
Gambar 2. 2. Struktur Grup Perusahaan Telkom Indonesia	12
Gambar 3. 1. Gambaran Website Statis dan Website Dinamis	18
Gambar 3. 2. <i>HTML 5</i>	21
Gambar 3. 3. <i>CSS 3</i>	21
Gambar 3. 4. <i>JavaScript</i>	22
Gambar 3. 5. <i>TypeScript</i>	22
Gambar 3. 6. <i>Next.JS</i>	24
Gambar 3. 7. <i>Node.JS</i>	24
Gambar 3. 8. <i>NPM</i>	25
Gambar 3. 9. Cara kerja <i>API</i>	26
Gambar 3. 10. Skema <i>REST API</i>	28
Gambar 3. 11. <i>Visual Studio Code</i>	30
Gambar 3. 12. <i>Chakra UI</i>	31
Gambar 3. 13. <i>Sweetalert2</i>	32
Gambar 3. 14. Simbol <i>Use Case Diagram</i>	34
Gambar 3. 15. Simbol <i>Flowchart</i>	35
Gambar 3. 16. Simbol <i>DFD</i>	36
Gambar 4. 1 <i>Use Case Diagram</i> Sistem Absensi Karyawan	43
Gambar 4. 2. <i>Entitiy Relationship Diagram</i> Sistem Absensi Karyawan	45
Gambar 4. 3. <i>Flowchart</i> Absensi	46
Gambar 4. 4. <i>Flowchart Login</i>	47
Gambar 4. 5. <i>Flowchart Dialy Summary</i>	48

Gambar 4. 6. <i>Flowmap Monthly Summary</i>	49
Gambar 4. 7. <i>DFD Level 0</i>	50
Gambar 4. 8. <i>DFD level 1</i>	51
Gambar 4. 9. <i>Layout Halaman Absensi</i>	53
Gambar 4. 10. <i>Layout Halaman Login</i>	53
Gambar 4. 11. <i>Layout Halaman Dashboard Admin</i>	54
Gambar 4. 12. <i>Layout Halaman Daily Summary</i>	54
Gambar 4. 13. <i>Layout Halaman Monthly Summary</i>	55
Gambar 4. 14. Instalasi Node.JS	55
Gambar 4. 15. <i>Setup Project</i>	56
Gambar 4. 16. Instalasi <i>library postgres</i>	56
Gambar 4. 17. Instalasi <i>library axios</i>	57
Gambar 4. 18. Instalasi <i>library prisma</i> dan <i>prisma client</i>	57
Gambar 4. 19. Instalasi <i>library Chakra UI</i> dan <i>Sweetalert2</i>	58
Gambar 4. 20. Instalasi <i>library React Chart JS 2</i>	58
Gambar 4. 21. Mengunduh <i>PostgreSQL</i>	59
Gambar 4. 22. Instalasi <i>PostgreSQL</i>	59
Gambar 4. 23. Instalasi <i>PostgreSQL</i>	60
Gambar 4. 24. Inisialisasi <i>prisma</i>	60
Gambar 4. 25. Isi <i>file schema.prisma</i>	61
Gambar 4. 26. Isi <i>file .env</i>	61
Gambar 4. 27. Proses migrasi <i>database</i> menggunakan <i>prisma</i>	61
Gambar 4. 28. <i>Database</i> pada <i>PostgreSQL</i>	62
Gambar 4. 29. Isi <i>code</i> halaman absensi	63
Gambar 4. 30. Halaman <i>form</i> absensi	64
Gambar 4. 31. Halaman sukses ketika berhasil absen.....	65

Gambar 4. 32. Isi <i>code</i> untuk halaman <i>login</i>	65
Gambar 4. 33. Halaman <i>login</i>	66
Gambar 4. 34. Isi <i>code</i> halaman <i>dashboard</i>	66
Gambar 4. 35. Tampilan halaman <i>dashboard</i>	67
Gambar 4. 36. Isi <i>code</i> halaman dan fitur <i>daily summary</i>	68
Gambar 4. 37. Tampilan halaman <i>daily summary</i>	69
Gambar 4. 38. Isi <i>code</i> dari halaman dan fitur <i>monthly summary</i>	70
Gambar 4. 39. tampilan halaman <i>monthly summary</i>	70
Gambar 4. 40. Pengujian halaman absensi 1	73
Gambar 4. 41. Pengujian halaman absensi 2	73
Gambar 4. 42. Pengujian halaman absensi 3	74
Gambar 4. 43. Pengujian halaman absensi 4.....	74
Gambar 4. 44. Pengujian halaman absensi 5	74
Gambar 4. 45. Pengujian halaman <i>login</i> 1	76
Gambar 4. 46. Pengujian halaman <i>login</i> 2	76
Gambar 4. 47. Pengujian halaman <i>login</i> 3	77
Gambar 4. 48. Pengujian halaman <i>login</i> 4	77
Gambar 4. 49. Pengujian halaman <i>dashboard</i> 1	79
Gambar 4. 50. Pengujian halaman <i>dashboard</i> 2	79
Gambar 4. 51. Pengujian halaman <i>dashboard</i> 3	80
Gambar 4. 52. Pengujian halaman <i>daily summary</i> 1	82
Gambar 4. 53. Pengujian halaman <i>monthly summary</i> 1	84

DAFTAR TABEL

Tabel 2. 1. Disiplin Jam Kerja.....	16
Tabel 4. 1. Analisa kebutuhan perangkat keras (<i>Hardware</i>).....	41
Tabel 4. 2. Identifikasi User Sistem Absensi Karyawan.....	44
Tabel 4. 3. Identifikasi <i>use case diagram</i> Sistem Absensi Karyawan	44
Tabel 4. 4. Perancangan basis data tabel absensi	52
Tabel 4. 5. Pengujian <i>black box</i> pada halaman absensi	71
Tabel 4. 6. Pengujian <i>black box</i> pada <i>login</i>	75
Tabel 4. 7. Pengujian <i>black box</i> sistem pada halaman <i>dashboard</i>	78
Tabel 4. 8. Pengujian <i>black box</i> pada fitur <i>daily summary</i>	80
Tabel 4. 9. Pengujian <i>black box</i> pada <i>monthly summary</i>	83

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Dalam era perkembangan teknologi yang pesat, penggunaan teknologi sebagai alat bantu untuk mempermudah berbagai aktivitas menjadi suatu keharusan. Pencatatan absensi pegawai, yang pada awalnya dilakukan secara konvensional, kini membutuhkan perubahan untuk beralih ke sistem yang lebih *modern* dan efisien. PT. Telkom Indonesia, khususnya *Tribe Technology Commercialization (Tribe TCO)*, saat ini masih mengandalkan *Google Form* sebagai sistem absensi dan rekap absensinya yang masih membutuhkan input manual ke *Spreadsheets*. Hal ini memunculkan kebutuhan untuk memperbarui sistem tersebut agar pekerjaan menjadi lebih efisien.

Pencatatan data secara konvensional seringkali berpotensi menyebabkan kesalahan dan kehilangan data, yang pada akhirnya dapat merugikan perusahaan. Melihat potensi masalah ini, Tribe TCO di PT. Telkom Indonesia merencanakan perubahan pada sistem absensinya. Tujuan utamanya adalah untuk menggantikan penggunaan *Google Form* dengan sistem aplikasi website yang dikembangkan secara internal. Selain itu, rekap absensi juga akan diotomatisasi, menghilangkan kebutuhan untuk melakukan input manual ke dalam *Spreadsheets*.

Penggunaan aplikasi *web* diusulkan sebagai solusi yang lebih efisien. Keuntungan utamanya adalah penghematan waktu yang signifikan, di mana karyawan yang bertugas merekap tidak perlu lagi menghabiskan waktu untuk melakukan *input* data absensi secara manual. Dengan demikian, perubahan ini tidak hanya akan memberikan manfaat dalam hal keakuratan data, tetapi juga akan meningkatkan efisiensi pengelolaan data absensi karyawan.

1.2. Tujuan

Tujuan dibuatnya aplikasi *website* absensi kepegawaian dalam penelitian kali ini adalah untuk:

1. Membangun sistem *website* absensi yang efisien dengan menggunakan *framework Javascript, Next.JS* juga menggunakan *Prisma ORM* dan *PostgreSQL* sebagai *database*.
2. Menggantikan sistem rekapitulasi dari manual menjadi layanan otomatis berbasis *website* yang terintegrasi dengan *database*.
3. Meningkatkan efisiensi kerja dalam merekap absensi karyawan untuk memperbaiki pengelolaan data dan mengurangi kesalahan dalam rekapitulasi.

1.3. Pembatasan Masalah

Adapun penulisan ini dibatasi dengan hal sebagai berikut:

1. Aplikasi berbasis *web* yang dibuat hanya mencakup *website* sistem absensi karyawan serta sebuah *dashboard admin* yang menyajikan *daily summary* dan *monthly summary* secara sederhana.
2. *Tools* yang digunakan dalam membangun *website* tersebut menggunakan *framework Next.JS 14.1.0*, *Chakra UI 2.8.0*, dan aplikasi yang digunakan untuk menyimpan *database* adalah *Prisma ORM* dan *PostgreSQL* dengan koneksi *Next.JS* menggunakan *Axios*.
3. Penggunaan *website* ini terbatas pada pihak manajemen *Tribe TCO* yang merekap data sebagai *admin* dan karyawan yang melakukan absensi sebagai *user*.
4. Aplikasi *website* tersebut dilengkapi dengan beberapa fitur umum yang sederhana. Pada akun *admin*, terdapat fitur *daily summary* untuk merekam absensi karyawan berdasarkan tanggal, serta fitur *monthly summary* untuk merekam absensi karyawan berdasarkan bulan. Setiap fitur tersebut memungkinkan pengguna untuk mengekspor data dalam format *.xls* melalui akun *admin*. Serta terdapat fitur autentikasi yaitu *login*.

5. Tampilan *user* dan *admin website* absensi karyawan sepenuhnya menggunakan *component* dari *Chakra UI*.

1.4. Sistematika Penulisan

Sistematika penulisan laporan ini terdiri dari beberapa bab disertai dengan kata pengantar, daftar isi, daftar gambar, daftar tabel, serta daftar Pustaka. Berikut pemaparan dari sistematika laporan ini.

BAB I PENDAHULUAN

Pada bab ini berisi tentang latar belakang masalah yaitu membahas hal-hal yang menjadi persoalan dalam judul laporan, tujuan yaitu membahas tentang tujuan dari judul yang penulis ambil, pembatasan masalah yaitu membatasi pembahasan yang diangkat oleh penulis dalam laporan, dan diakhiri dengan sistematika penulisan yang dapat memudahkan penulis dalam menyusun laporan.

BAB II PROFIL PERUSAHAAN

Bab ini berisi tinjauan umum perusahaan PT. Telkom Indonesia, membahas mengenai sejarah perusahaan, profil perusahaan, lokasi perusahaan, struktur organisasi perusahaan, visi dan misi perusahaan, layanan produk dan jasa perusahaan serta dan disiplin kerja PT. Telkom Indonesia.

BAB III LANDASAN TEORI

Bab ini berisi tentang konsep dan kajian teori-teori penunjang yang penulis rangkum untuk menjadi acuan dalam memahami permasalahan yang dibahas.

BAB IV PEMBAHASAN

Bab ini berisi tentang penjelasan mengenai skenario perancangan sistem dan proses *script*, meliputi analisa masalah, analisis kebutuhan perangkat, ERD, Use Case Diagram, *flowchart*, *Data flow Diagram*, langkah pembuatan dan pengujian aplikasi dengan judul laporan MEMBANGUN SISTEM ABSENSI BERBASIS WEBSITE MENGGUNAKAN NEXT JS, CHAKRA UI, PRISMA ORM DAN POSTGRESQL SEBAGAI DATABASE DI PT. TELKOM INDONESIA.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dari kegiatan pada bab IV serta terdapat saran-saran yang bersifat membangun terhadap proyek yang telah dibuat.

BAB II

PROFIL PERUSAHAAN

2.1. Profil PT. Telkom Indonesia

PT. Telekomunikasi Indonesia, Tbk. (Telkom Indonesia) adalah perusahaan informasi dan komunikasi serta penyedia jasa dan jaringan telekomunikasi secara lengkap di Indonesia. Telkom Indonesia mengklaim sebagai perusahaan telekomunikasi terbesar di Indonesia, dengan jumlah pelanggan telepon tetap sebanyak 15 juta dan pelanggan telepon seluler sebanyak 50 juta. Telkom Indonesia merupakan salah satu perusahaan BUMN (Badan Usaha Milik Negara).

Profil singkat PT. Telkom Indonesia atau profil perusahaan tempat kerja praktek dilaksanakan dapat dilihat seperti berikut

1. Industri : Informasi dan Komunikasi
2. Didirikan : 23 Oktober 1856 (diluncurkan kembali tanggal 23 Oktober 2009)
3. Kantor : Jl. Gegerkalong Hilir, Sukarasa, Sukasari Kota Bandung, Jawa Barat
4. Produk : *Metanesia, Blockchain, AR dan VR*
5. Pemilik : Pemerintah Indonesia
6. Situs *web* : www.telkom.co.id

Penjelasan tentang sejarah, logo, visi, misi, struktur organisasi PT. Telkom Indonesia akan dijelaskan pada sub bab berikutnya.

2.2. Sejarah dan Perkembangan PT. Telkom Indonesia

PT. Telekomunikasi Indonesia, Tbk (Telkom Indonesia) adalah suatu Badan Usaha Milik Negara (BUMN) yang bergerak dalam bidang jasa telekomunikasi. PT. Telkom Indonesia menyediakan sarana dan jasa layanan telekomunikasi dan informasi kepada masyarakat luas sampai ke pelosok daerah di seluruh Indonesia. Sejarah PT. Telkom Indonesia pertama kali berawal dari sebuah badan usaha swasta penyediaan layanan pos dan telegraf yang didirikan kolonial Belanda pada tahun 1882. Pada tahun 1905 pemerintah kolonial Belanda mendirikan perusahaan telekomunikasi sebanyak tiga puluh delapan perusahaan. Kemudian pada tahun

1906 pemerintah Hindia Belanda membentuk suatu jawatan Pos, Telegraf dan Telepon (Post, *Telegraph en Telephone Dienst* / PTT).

Pada tahun 1961 status jawatan diubah menjadi Perusahaan Negara Pos dan Telekomunikasi (PN Postel). Kemudian pada tahun 1965 pemerintah memisahkannya menjadi Perusahaan Negara Pos dan Giro (PN Pos dan Giro) dan Perusahaan Negara Telekomunikasi (PN Telekomunikasi). Pada tahun 1974 Perusahaan Negara Telekomunikasi disesuaikan menjadi Perusahaan Umum Telekomunikasi (PERUMTEL) yang menyelenggarakan jasa telekomunikasi nasional dan internasional.

Pada tahun 1980 Indonesia mendirikan suatu badan usaha untuk jasa Telekomunikasi Internasional yang bernama PT. *Indonesian Satellite Corporation* (INDOSAT) yang terpisah dari PERUMTEL. Pada tahun 1989 pemerintah Indonesia mengeluarkan UU No.3/ 1989 mengenai Telekomunikasi, yang isinya tentang peran swasta dalam penyelenggaraan Telekomunikasi. Pada tahun 1991 PERUMTEL berubah bentuk menjadi perusahaan perseroan (Persero) Telekomunikasi Indonesia berdasarkan PP No.25/ 1991 sampai sekarang.

Perubahan di lingkungan PT. Telkom Indonesia, Tbk terus berlanjut mulai dari perusahaan jawatan sampai perusahaan publik. Perubahan-perubahan besar terjadi pada tahun 1995 meliputi :

1. Restrukturisasi Internal
2. Kerjasama Internal
3. Intial Publik Offering (IPO)

Jenis usaha PT. Telkom Indonesia, Tbk adalah penyelenggara jasa Telekomunikasi dalam negeri dan bidang usaha terkait seperti jasa Sistem Telepon Bergerak Sirkuit (STBS) pelanggan, teleks, penyewaan transponder satelit, *Very Small Aperture Terminal* (VSAT) dan jasa nilai tambah tertentu.

Pada tanggal 1 Juli 1995 organisasi PT. Telkom Indonesia, Tbk mengubah struktur jenis jasa telekomunikasi menjadi tujuh divisi regional dan satu divisi *network* yang keduanya mengelola bidang usaha utama. Divisi regional sebagai pengganti struktur WITEL yang memiliki daerah teritorial tertentu, namun hanya

menyelenggarakan jasa telepon lokal dan mendapat bagian dari jasa SLJJ dan SLI. Divisi *network* menyelenggarakan jasa telekomunikasi jarak jauh.

Unit-unit bisnis PT. Telkom Indonesia, Tbk terdiri dari Divisi, Pusat, Yayasan dan Anak Perusahaan. Adapun divisi yang tersedia di PT. Telkom Indonesia yaitu:

1. Divisi *Long Distance*
2. *Carrier and Interconnection Service*
3. Divisi Multimedia Divisi
4. *Fixed Wireless Network*
5. *Enterprise Service*
6. Divisi Regional I – Sumatera
7. Divisi Regional II – Jakarta
8. Divisi Regional III – Jawa Barat
9. Divisi Regional IV – Jawa Tengah dan Yogyakarta
10. Divisi Regional VI - Kalimantan
11. Divisi Regional VII – Kawasan Timur Indonesia
12. *Maintenance Service Centre*
13. *Training Centre*
14. *Carrier Development Support Centre*
15. *Management Consulting Centre*
16. *Construction Centre*
17. *I/ S Centre*
18. *R and D Centre*
19. *Community Development Centre (CDC)*

PT. Telkom Indonesia memiliki beberapa yayasan yaitu :

1. Dana Pensiun (Dapentel)
2. Yayasan Pendidikan PT. Telkom Indonesia, Tbk.

3. Yayasan Kesehatan
4. Yayasan Sandhykara Putra telkom (YSPT)

PT. Telkom Indonesia memiliki anak perusahaan berdasarkan besar kepemilikan yaitu:

1. Kepemilikan > 50%
 - a. PT. Telekomunikasi Seluler (Telkomsel) : Telekomunikasi (Seluler GSM)
 - b. PT. Dayamitra Telekomunikasi (Dayamitra) : Telekomunikasi (KSO-VI Kalimantan)
 - c. PT. Infomedia Nusantara (Infomedia) : Layanan Informasi
 - d. PT. *AriWest International* (AriaWest) : Telekomunikasi Telepon Tetap (KSO-III Jawa Barat dan Banten)
 - e. PT. Pramindo Ikat Nusantara (Pramindo) : Telekomunikasi Telepon Tetap (KSO-I Sumatera)
 - f. PT. Multimedia Nusantara (Metra) : Multimedia, pay special TV
 - g. PT. Napsindo Primatel International (Napsindo) : *Network Access Point*
 - h. PT. Indonusa Telemedia (Indonusa) : TV Cable (baru)
 - i. PT. Graha Sarana Duta (GSD) : Properti, Konstruksi dan Jasa (baru)
2. Kepemilikan 20% - 50%
 - a. PT. Patra Komunikasi Indonesia (Patrakom) : Layanan VSAT
 - b. PT. Cita Sari Makmur (CSM) : VSAT dan layanan Telekomunikasi lainnya
 - c. PT. Pasifik Satelit Nusantara (PSN) : Transponder Satelit dan Komunikasi
3. Kepemilikan < 20%

- a. PT. Mandara Seluler Indonesia (MSI) : Layanan NMT – Seluler dan CDMA
- b. PT. Batam Bintan Telekomunikasi (Babintel) Telepon Tetap di Batam dan Pulau Bintan
- c. PT. Pembangunan Telekomunikasi Indonesia (Bangtelindo) : Pengelolaan Jaringan dan Peralatan Telekomunikasi

Divisi Multimedia dan Divisi Pembangunan ditetapkan 31 Desember 1996 berdasarkan keputusan direksi PT. Telkom Indonesia, Tbk. Seiring dengan diberlakukannya pasar bebas maka PT. Telkom Indonesia, Tbk membentuk kerja sama dengan para investor dan operator kelas dunia yang disebut dengan pola Kerja Sama Operasi (KSO). Tujuan dibentuknya KSO adalah :

- 1. Mempercepat pembangunan Telekomunikasi untuk kurun waktu IV, karena pendanaan disediakan oleh mitra KSO.
- 2. Memperoleh ahli teknologi kelas dunia yang bergabung dalam mitra KSO.
- 3. Meningkatkan kemampuan berkompetensi dalam era pasar bebas.

Pada tahun 1999, pemerintah Indonesia menetapkan Undang-Undang Nomor 36 Tahun 1999 mengenai penghapusan monopoli penyelenggaraan telekomunikasi. Memasuki abad ke-21, pemerintah melakukan penyesuaian regulasi mengenai sektor telekomunikasi dengan membuka kompetisi pasar bebas. Dengan demikian, PT. Telkom Indonesia tidak lagi memonopoli industri telekomunikasi di Indonesia. Tahun 2001 Telkom membeli 35% saham Telkomsel dari

PT. Indosat sebagai bagian dari implementasi restrukturisasi industri jasa telekomunikasi di Indonesia yang ditandai dengan penghapusan kepemilikan bersama dan kepemilikan silang antara Telkom dan Indosat. Sejak bulan Agustus 2002 terjadi duopoli penyelenggaraan telekomunikasi lokal. Pada tanggal 23 Oktober 2009, PT. Telkom Indonesia merilis “*New Telkom*” atau “*Telkom Baru*”. Peluncuran “*New Telkom*” diganti dengan pergantian identitas perusahaan.

2.3. Visi dan Misi PT. Telkom Indonesia

2.3.1. Visi PT. TELKOM

“To become a leading Telecommunication, Information, Media & Edutainment (TIME) Player in the Region”. Yang memiliki arti untuk menjadi telekomunikasi terkemuka, informasi, media & edutainment pengguna di daerah.

2.3.2. Misi PT. TELKOM

1. *To Provide TIME Services with Excellent Quality & Competitive Price*
2. *To be the Role Model as the Best Managed Indonesian Corporation.*

Yang memiliki arti untuk menyediakan layanan waktu dengan cepat, kualitas dan harga kompetitif. Menjadi model peran sebagai perusahaan terbaik yang dikelola di Indonesia.

2.4. Logo PT. Telkom Indonesia

Sebuah logo akan menjadi suatu Brand Images dimana dari suatu perusahaan. Logo juga bersifat persepsi kuat terhadap perusahaan. PT. TELKOM sekarang menggunakan logo terbarunya yang diluncurkan pada tahun 2013. Pada Gambar 2.1 merupakan logo PT. Telkom Indonesia



Gambar 2. 1. Logo Perusahaan Telkom Indonesia

(Sumber https://www.telkom.co.id/sites/about-telkom/id_ID/page/brand-guideline-logo-telkom-indonesia-657)

2.4.1. Keterangan logo PT. TELKOM Indonesia :

The World In Your Hand : Sebuah logo akan menjadi suatu *Brand Images* dimana dari suatu perusahaan. Sudah banyak perusahaan-perusahaan yang melakukan transformasi visi dan misi melalui logo contohnya Telkom. Logo juga bersifat persepsi kuat terhadap perusahaan. Adapun arti dari simbol-simbol logo PT. TELKOM Indonesia sebagai berikut :

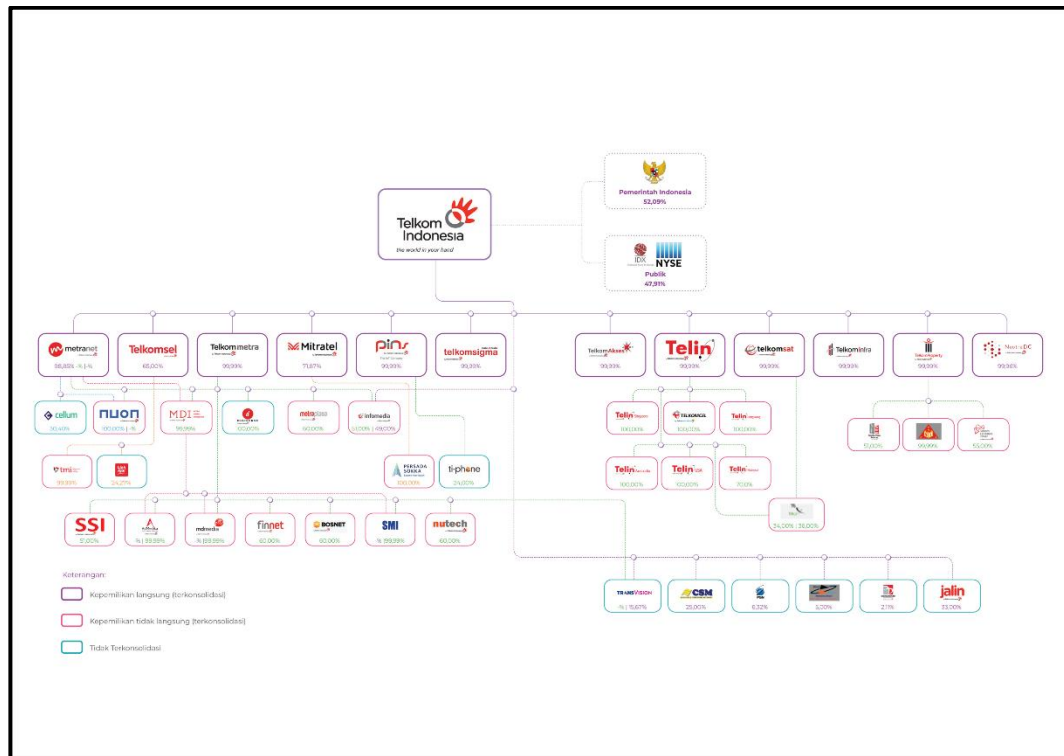
1. Lingkaran: sebagai simbol dari kelengkapan produk dan layanan dalam portofolio bisnis baru TELKOM yaitu *Telecommunication, Information, Media & Edutainment (TIME), Expertise*.
2. Tangan yang meraih ke luar: Simbol ini mencerminkan pertumbuhan dan ekspansi ke luar, *Empowering*.
3. Jemari tangan: Simbol ini memaknai sebuah kecermatan, perhatian, serta kepercayaan dan hubungan yang erat, *Assured*.
4. Kombinasi tangan dan lingkaran: Simbol dari matahari terbit yang maknanya adalah perubahan dan awal yang baru, *Progressive*.
5. Telapak tangan : mencerminkan kehidupan untuk menggapai masa depan, *Heart*.

2.4.2. Warna–warna yang digunakan pada logo adalah :

1. **Merah** Mencerminkan spirit Telkom untuk selalu optimis dan berani dalam menghadapi tantangan dan perusahaan.
2. **Putih** Mencerminkan spirit Telkom untuk memberikan yang terbaik bagi bangsa.
3. **Hitam** Melambangkan kemauan keras.
4. **Abu** Melambangkan teknologi.

2.5. Struktur Grup Perusahaan Telkom Indonesia

Berikut ini struktur grup perusahaan PT. Telkom Indonesia terlihat pada Gambar 2.2. di bawah ini.



Gambar 2. 2. Struktur Grup Perusahaan Telkom Indonesia

(Sumber https://www.telkom.co.id/sites/about-telkom/id_ID/page/struktur-group-perusahaan-208)

2.6. Produk dan Layanan Perusahaan

Sebagai perusahaan penyelenggara layanan *TIME*, Telkom berkomitmen untuk terus melakukan inovasi produk dan layanan di sektor-sektor di luar telekomunikasi. Kami menciptakan produk dan layanan yang mampu mengantisipasi perkembangan konten (*content*) dan perangkat (*device*), baik itu *smartphone*, PC (*Personal Computer*) atau *tablet*, yang sangat pesat. Selain itu, inovasi Kami juga dikembangkan dengan menggabungkan komunikasi suara, layanan data via internet dan IPTV dalam satu paket yang Kami sebut *Triple Play*. Produk dan layanan inovatif ini sangat berbeda dari kompetitor Kami sehingga memberikan keunggulan bagi Telkom dalam hal *Time to Market* dan

memposisikannya sebagai perusahaan yang prestisius di tahun-tahun yang akan datang. Kreativitas ini pula yang mendorong evolusi besar dalam tubuh Telkom yang terefleksi dalam program transformasi yang telah dimulai sejak penghujung tahun 2009.

Namun untuk mengantisipasi dinamika bisnis telekomunikasi yang semakin pesat, Telkom berdasarkan Peraturan Perusahaan Perseroan PT Telekomunikasi Indonesia, Tbk. No.PD.506 tahun 2011 tentang Pohon Produk mensinergikan seluruh produk, layanan dan solusinya, mulai dari produk dan layanan *legacy* hingga *new wave*, dan mengelompokkannya ke dalam sejumlah kegiatan usaha sebagaimana berikut ini:

2.6.1. Layanan Teknologi Informasi (“TI”)

1. *Managed Application & Performance/ITO Cloud Based Managed Services*

Yang berbasis aplikasi *client server* mendukung layanan *cloud computing* yang dapat diakses pelanggan melalui jaringan internet.

2. *Server Based Managed Services*

Yang berbasis *server* yang diakses oleh pelanggan melalui layanan internet atau aplikasi khusus lain sebagai *client*.

3. *IT Consulting*

Merupakan layanan konsultasi TI yang terdiri dari pekerjaan pengembangan sistem/aplikasi, *Blueprint/MasterPlan* TI atau perancangan dokumen strategis lainnya

2.6.2. *E-Payment/Payment Service*

1. *Billing Payment*

Adalah layanan yang memudahkan proses transaksi pembayaran bagi pelanggan jasa atau barang kepada perusahaan penyedia jasa seperti PLN, Telkom, PDAM, PJKA, dan lain-lain melalui jasa yang disediakan *collection agent* di antaranya bank, koperasi, BPR, *convenience store*, dan lain-lain sehingga tidak harus mendatangi perusahaan bersangkutan.

2. *Remittance*

Adalah layanan pengiriman uang di mana pengirim dan penerima dana tidak harus memiliki rekening di bank namun harus memiliki perangkat seluler untuk menerima notifikasi.

3. *E-Money*

Melayani penyimpanan uang pelanggan/penggunaannya secara elektronik dalam suatu media (handphone, kartu prabayar, atau suatu rekening virtual yang dapat diakses melalui media internet) untuk melakukan transaksi secara elektronik.

4. *E-Voucher*

Atau *Telkom Voucher* merupakan *single voucher* yang diterbitkan Telkom yang berfungsi untuk membeli layanan atau mengisi pulsa layanan milik *Telkom Group*, seperti *Kartu As*, *simPATI* dan *Flexi Trendy*, layanan prabayar *TelkomVision* dan *Speedy Hotspot*.

2.6.3. *IT enabler Services (“ITeS”)*

1. *Business Process Outsourcing (“BPO”)*

Merupakan layanan berbasis kontrak untuk mengelola/menjalankan beberapa proses perusahaan pelanggan.

2. *Knowledge Process Outsourcing (“KPO”)*

Merupakan layanan BPO untuk proses bisnis yang memerlukan kompetensi khusus untuk pengelolaannya.

2.6.4. *Network Centric VAS*

1. *Data & Telephony*

Merupakan layanan data dan telepon berbasis TI yang menawarkan nilai tambah bagi pelanggan layanan konektivitas.

2. *Security*

Merupakan layanan *security* berbasis TI yang bernilai tambah bagi pelanggan layanan konektivitas.

3. *Server & Storage*

Merupakan layanan *server & storage* berbasis TI yang bernilai tambah bagi pelanggan layanan konektivitas.

2.6.5. Jasa Integrasi

1. Jasa Integrasi Jaringan & Piranti Keras

Memadukan dan menyediakan perangkat konektivitas di lokasi pelanggan (CPE).

2. Jasa Integrasi Aplikasi dan Piranti Lunak

Memadukan dan menyediakan seluruh sistem yang diperlukan untuk mengintegrasikan dua atau lebih aplikasi/piranti lunak/sistem.

3. Jasa Integrasi *Computing Hardware*

Memadukan sistem yang menggabungkan seluruh perangkat komputasi yang diperlukan untuk membantu sistem informasi bagi pelanggan.

2.7. Penghargaan Perusahaan

1. Indonesia Most Reputable Companies 2023, Predikat Excellent, Peringkat 1 (Champion), kategori Telekomunikasi
2. Bisnis Indonesia Corporate Social Responsibility Award (BISRA) 2023, Gold di Kategori Pilar Ekonomi
3. The 8th PR INDONESIA Awards, kategori Terpopuler - Media Mass
4. Best Digital Transformation Company

BAB III

LANDASAN TEORI

3.1. Sistem Informasi

Berikut terdapat beberapa penjelasan mengenai pengertian atau definisi sistem informasi menurut para ahli, diantaranya yaitu:

1. Menurut (Cegielski, 2015) Sistem informasi adalah proses mengumpulkan, memproses, menganalisis, dan menyebarkan informasi untuk tujuan tertentu.
2. Menurut (Leitch, 2011) Sistem informasi adalah suatu sistem yang terdapat di dalam sebuah organisasi yang mempertemukan kebutuhan pengelola transaksi harian, mendukung operasi, bersifat managerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.
3. Menurut (O'brien, 2011) Sistem informasi adalah suatu kombinasi teratur apapun baik dari *people*, *hardware*, *software*, maupun *database* yang mengumpulkan, mengubah, dan menyebarkan informasi di dalam suatu bentuk organisasi.

(Sano, n.d.)

3.2. Jaringan Komputer

Jaringan komputer adalah sebuah sistem yang terdiri atas komputer-komputer yang didesain untuk dapat berbagi sumber daya (*printer*, *CPU*), berkomunikasi (surel, pesan instan), dan dapat mengakses informasi (peramban web). Tujuan dari jaringan komputer adalah agar dapat mencapai tujuannya, setiap bagian dari jaringan komputer dapat meminta dan memberikan layanan (*service*). Pihak yang meminta/menerima layanan disebut klien (*client*) dan yang memberikan/mengirim layanan disebut peladen (*server*). Desain ini disebut dengan sistem *client-server*, dan digunakan pada hampir seluruh aplikasi jaringan komputer (Yudianto, 2003).

3.3. Aplikasi

Menurut Kamus Besar Bahasa Indonesia, aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

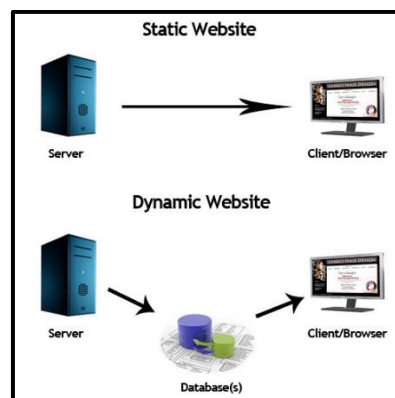
3.4. Website

3.4.1. Secara Umum

Website disebut juga *site*, situs, situs *web* atau portal. Merupakan kumpulan halaman *web* yang berhubungan antara satu dengan lainnya, halaman pertama sebuah *website* adalah *home page*, sedangkan halaman demi halamannya secara mandiri disebut *web page*, dengan kata lain *website* adalah situs yang dapat diakses dan dilihat oleh para pengguna internet diseluruh dunia. (Santoso, 2014)

3.4.2. Jenis Website

Menurut (Santoso, 2014) “Membagi *website* menjadi golongan kanan dan golongan kiri. Dalam *website* dikenal dengan sebutan *website dinamis* dan *website statis*.”



Gambar 3. 1. Gambaran Website Statis dan Website Dinamis

(Sumber <https://www.exabytes.co.id/blog/apa-itu-website-dinamis/>)

1. *Website Statis*

Website Dinamis adalah *website* yang mempunyai halaman konten yang tidak berubah-ubah.

2. *Website Dinamis*

Website dinamis adalah *website* yang secara struktur ditunjukkan untuk *update* sesring mungkin.

3.4.3. Fungsi Website

Berdasarkan fungsinya *website* dibagi menjadi beberapa jenis, diantaranya:

1. *Website Pribadi*

Website pribadi biasanya dibuat oleh seseorang untuk kebutuhan profesionalitas atau menyampaikan opini pribadi. Misalnya saja untuk portofolio *online* atau *blog* pribadi.

Saat ini tersedia berbagai layanan untuk membuat *website* pribadi, mulai dari yang gratis sampai layanan berbayar. Untuk yang gratis, misalnya saja Blogspot.com dan juga WordPress.com (Waryanto, 2021).

2. *Website Toko Online/e-Commerce*

Jenis *website* selanjutnya adalah *website* toko *online*. Selayaknya toko fisik di dunia nyata, *website* jenis ini difungsikan sebagai toko, namun bedanya terdapat di ranah *online*.

Dengan *website* toko *online* dapat menjangkau lebih banyak konsumen dari berbagai wilayah, buka toko 24 jam, hingga transaksi yang cepat dan mudah (Waryanto, 2021).

3. *Blog*

Blog biasanya berisi artikel yang bertujuan untuk berbagi pengetahuan hingga strategi pemasaran online.

Setiap blog biasanya memiliki *niche blog* yang jadi spesifikasinya, baik itu tentang *travel*, kecantikan, teknologi, finansial, dan lainnya. (Waryanto, 2021).

Sehingga berdasarkan dari uraian teori diatas dapat ditarik kesimpulan bahwa *website* merupakan kumpulan halaman- halaman yang dapat menampilkan teks, gambar, animasi, video, ataupun suara yang masing-masing dihubungkan dengan jaringan- jaringan halaman. Berdasarkan sifatnya, *website* dibagi menjadi dua golongan yaitu *website* statis dan *website* dinamis. Sedangkan jika dilihat berdasarkan fungsinya, *website* terbagi menjadi tiga yaitu *website* personal (pribadi), *website* toko online (*e- Commerce*), dan *blog* (Waryanto, 2021).

3.5. Bahasa Pemrograman

Bahasa pemrograman adalah sejumlah kata kode, dan simbol yang membuat programmer dapat menyampaikan perintah kepada komputer. Programmer menggunakan sebuah perangkat pemrograman untuk menciptakan program. Program yang memudahkan pengguna dalam membangun program yang disebut perangkat pemrograman. Sama halnya dengan manusia yang mengucapkan berbagai bahasa (Inggris, Spanyol, Perancis, dan lain-lain). Programmer menggunakan bahasa pemrograman dan perangkat untuk menciptakan program. (HP, 2018).

3.5.1. HTML

HTML (Hypertext Markup Language) adalah bahasa dasar untuk *web scripting* bersifat *client side* yang memungkinkan untuk menampilkan informasi dalam bentuk teks, grafik, serta multimedia dan juga untuk menghubungkan antar tampilan *web page (hyperlink)* (Febio, 2011).



Gambar 3. 2. *HTML 5*

(Sumber https://en.m.wikipedia.org/wiki/File:HTML5_logo_and_wordmark.svg)

3.5.2. CSS

CSS kepanjangan dari *Cascading Style Sheet* adalah bahasa yang merepresentasikan halaman *web*. Seperti warna, *layout*, dan *font*. Dengan menggunakan *CSS*, seorang *web developer* dapat membuat halaman *web* yang dapat beradaptasi dengan berbagai macam ukuran layar. Pembuatan *CiSS* biasanya terpisah dengan halaman *HTML*. Meskipun *CSS* dapat disisipkan di dalam halaman *HTML*. Hal ini ditujukan untuk memudahkan pengaturan halaman *HTML* yang memiliki rancangan yang sama (Pahlevi, 2018).



Gambar 3. 3. *CSS 3*

(Sumber https://en.m.wikipedia.org/wiki/File:CSS3_logo_and_wordmark.svg)

3.5.3. *JavaScript*



Gambar 3. 4. *JavaScript*

(Sumber <https://en.wikipedia.org/wiki/File:JavaScript-logo.png>)

JavaScript adalah bahasa skrip berorientasi objek yang dapat digunakan secara lintas *platform* untuk membuat halaman web menjadi interaktif. Terdapat juga versi *JavaScript* yang lebih canggih untuk sisi *server*, seperti *Node.js*, yang memungkinkan penambahan lebih banyak fungsionalitas ke situs web daripada sekadar mengunduh *file*, seperti kolaborasi *real-time* antara beberapa komputer. Dalam sebuah lingkungan *host*, misalnya pada *browser web*, *JavaScript* dapat berinteraksi dengan objek-objek di sekitarnya untuk memberikan kontrol programatis terhadap mereka. (Sunyoto, 2010)

3.5.4. *TypeScript*



Gambar 3. 5. *TypeScript*

(Sumber <https://nl.wikipedia.org/wiki/TypeScript>)

TypeScript merupakan bahasa pemrograman turunan *JavaScript*, memberikan peningkatan fungsionalitas melalui penambahan fitur-fitur seperti pengetikan statis, kelas, antarmuka, dan modul. Penelitian ini bertujuan untuk memahami dan mengimplementasikan fitur-fitur tersebut dalam konteks pengembangan perangkat lunak. (Nugroho, 2023)

3.6. Framework

Framework merupakan kerangka kerja untuk mengembangkan aplikasi berbasis *website* maupun *desktop*. Kerangka kerja disini sangat membantu *developer* dalam menuliskan sebuah dengan lebih terstruktur dan tersusun rapi.

Kerangka kerja diciptakan untuk mempermudah kinerja dari *programmer*. Sehingga, seorang *programmer* tidak perlu untuk menuliskan kode secara berulang-ulang. Karena di dalamnya sendiri anda hanya perlu menyusun komponen-komponen pemrograman saja.

Fungsi *framework* adalah membuat *source code* menjadi lebih terstruktur. Terstruktur disini, berarti program yang dibuat akan dimasukkan ke dalam setiap komponen sesuai dengan fungsinya masing-masing. Salah satu contoh dari kode program terstruktur dapat dilihat dari *framework Javascript*, yaitu *React* dan *Next.JS*.

Apabila ingin menambahkan atau mengurangi beberapa fitur dari *website* yang telah dibuat, maka dengan menggunakan *framework* dapat memudahkan anda untuk melakukan hal tersebut. Dengan melakukan *maintenance*, anda dapat mengubah versi *website* tersebut dan menambahkan beberapa fitur dengan lebih mudah dan aman. Sehingga, ketika ada perbaikan pada *website* maka terdapat notifikasi atau pesan bagi pengguna bahwa *website* masih dalam perbaikan. Selanjutnya, dari segi dokumentasi juga lebih terstruktur (Adani, 2020).

3.6.1. React

React merupakan sebuah *library JavaScript* yang memungkinkan pengembang untuk mendeskripsikan tampilan (*view*) dalam halaman web, khususnya dalam format *HTML*, berdasarkan beberapa keadaan (*state*), yang umumnya terkait dengan data. Pendekatan ini memudahkan pengelolaan dan pembaruan tampilan berdasarkan perubahan data atau interaksi pengguna, menjadikan *React* pilihan yang kuat untuk membangun antarmuka pengguna yang dinamis dan responsif (Musyaffa et al, 2024).

3.6.2. *Next.JS*

Next.JS merupakan sebuah *framework React* yang handal dan memiliki kinerja tinggi untuk pengembangan *web modern*. *Framework* ini menyajikan berbagai fitur canggih, termasuk *rendering hibrid*, *prefetching rute*, optimasi gambar otomatis, dan dukungan internasionalisasi secara bawaan (Musyaffa et al, 2024).



Gambar 3. 6. *Next.JS*

(Sumber <https://gorzelinski.com/blog/next-js-overview-in-1000-words/>)

3.6.3. *Node.JS*

Node.js adalah *runtime environment* untuk JavaScript yang bersifat *open-source* dan *cross-platform*. Dengan *Node.js* kita dapat menjalankan kode JavaScript di mana pun, tidak hanya terbatas pada lingkungan *browser*. *Node.js* juga menyediakan banyak *library/module* JavaScript yang membantu menyederhanakan pengembangan aplikasi web. (Nugroho, 2023)



Gambar 3. 7. *Node.JS*

(Sumber https://en.wikipedia.org/wiki/File:Node.js_logo.svg)

3.6.4. *Node Package Manager*

Node Package Manager atau NPM adalah *package manager* dan *repository software* dalam bahasa pemrograman *JavaScript* yang digunakan untuk

memasang, menghapus, dan mengelola *library* atau *package software* dalam proyek pengembangan aplikasi.

NPM memungkinkan *developer* membagikan kode yang mereka tulis dan menggunakan kode yang ditulis oleh orang lain, memfasilitasi kolaborasi dan mempercepat proses pengembangan. *Repository* NPM berisi lebih dari satu juta *package* perangkat lunak yang dapat dipakai dan dikontribusikan oleh siapa saja.

Salah satu fitur utama NPM adalah *file package.json* yang digunakan untuk menentukan dependensi dan *metadata* proyek. Fungsi ini mempermudah manajemen versi dan distribusi kode. (Revoupedia, 2023)



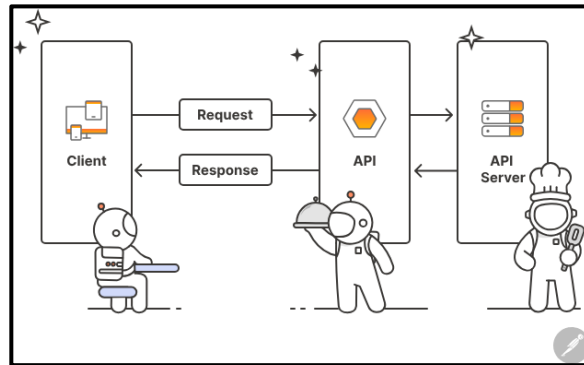
Gambar 3. 8. NPM

(Sumber <https://id.m.wikipedia.org/wiki/Berkas:Npm-logo.svg>)

3.7. API

Application Programming Interface (API) adalah sebuah teknologi yang memfasilitasi pertukaran informasi atau data antara dua atau lebih aplikasi perangkat lunak. *API* adalah antarmuka virtual antara dua fungsi perangkat lunak yang saling bekerja sama. Sebuah *API* mendefinisikan bagaimana cara programmer memanfaatkan suatu fitur tertentu dari sebuah komputer. (Jagoan Hosting, 2023)

Berikut pada gambar 3.9 di bawah ini, diilustrasikan cara kerja *API* dengan analogi pelanggan sebagai klien, pelayan sebagai *API*, dan koki atau pembuat makanan sebagai *API Server*.



Gambar 3. 9. Cara kerja *API*

(Sumber <https://www.postman.com/what-is-an-api/>)

3.7.1. Jenis *API*

Berdasarkan hak aksesnya terdapat empat jenis *API*, yaitu sebagai berikut.

1. *Public API*

Public API merupakan antarmuka aplikasi yang dapat diakses secara terbuka oleh berbagai pengembang. Ketersediaan sumber daya ini secara luas membuatnya menjadi pilihan utama dalam pengembangan aplikasi. Penelitian ini bertujuan untuk menganalisis penggunaan *API* publik sebagai sumber informasi terbuka dalam konteks aplikasi perangkat lunak. (Jagoan Hosting, 2023)

2. *Private API*

Private API adalah antarmuka aplikasi yang tidak terbuka secara umum dan cenderung terbatas. Pada umumnya *Private API* dibuat untuk kebutuhan internal dalam pengembangan aplikasi tertentu. Sebagai contoh *API* dari *front-end* yang digunakan untuk mengakses data di *back-end* dari sebuah *website* atau dalam pengembangan aplikasi *mobile*. (Jagoan Hosting, 2023)

3. *Partner API*

Partner API merujuk pada antarmuka pemrograman aplikasi yang dapat diakses oleh mitra bisnis atau pihak yang telah memperoleh izin khusus. Sama seperti *public API*, akses ke *Partner API* memerlukan proses pendaftaran dengan penyedia layanan *API*. Setelah mendapatkan izin, mitra bisnis dapat mengakses data dan sumber informasi yang disediakan oleh *platform* tersebut. (Jagoan Hosting, 2023)

4. *Composite API*

Composite API mengacu pada antarmuka pemrograman aplikasi yang memungkinkan pengguna untuk mengakses berbagai jenis data dari berbagai *server* dan *hosting* melalui satu titik akses tunggal. Dengan kata lain, *Composite API* menggabungkan atau menyatukan data dari sumber-sumber yang berbeda menjadi satu tampilan yang terpadu. (Jagoan Hosting, 2023)

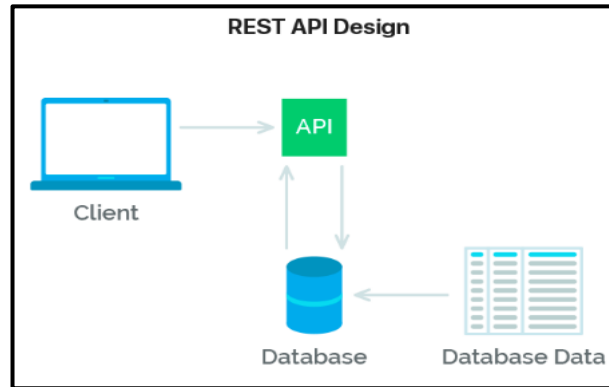
Kelebihan utama dari *Composite API* adalah kemampuannya untuk memberikan pengguna akses terpusat terhadap data yang tersebar di berbagai lokasi. Dengan menggunakan *Composite API*, pengguna tidak perlu berinteraksi dengan *multiple-endpoint* atau melakukan banyak permintaan *API* terpisah untuk mendapatkan informasi yang diinginkan. Sebaliknya, mereka dapat mengakses dan menggabungkan berbagai jenis data hanya dengan satu permintaan ke *Composite API*. (Jagoan Hosting, 2023)

3.7.2. **Arsitektur API**

Ada tiga macam arsitektur yang sering dipakai pada API, di antaranya adalah REST, SOAP, dan RPC. Berikut penjelasannya.

1. *Representational State Transfer (REST)* adalah arsitektur perangkat lunak yang memberlakukan syarat mengenai cara *API* bekerja. *REST* pada awalnya dibuat sebagai panduan untuk mengelola komunikasi pada jaringan kompleks seperti internet. *REST* dapat digunakan untuk

mendukung komunikasi berperforma tinggi dan andal sesuai skala. *REST* juga dapat dengan mudah diterapkan, membawa visibilitas dan portabilitas lintas *platform* ke semua sistem *API*. (Jagoan Hosting, 2023)



Gambar 3. 10. Skema *REST API*

(Sumber <https://medium.com/wripolinema/memahami-api-rest-api-dan-restful-api-5fd2327edd3c>)

2. *Simple Object Access Protocol (SOAP)*, dimana bentuk data yang digunakan adalah *XML*. *SOAP* memungkinkan semua data disimpan dalam dokumen tertentu. (Jagoan Hosting, 2023)
3. *RPC (Remote Procedure Call)* merupakan teknologi yang ditujukan membuat komunikasi antara client side dan server side lebih sederhana sederhana. (Jagoan Hosting, 2023)

3.8. Database

Database atau basis data merupakan kumpulan dari data yang memiliki hubungan antara satu dengan yang lainnya, tersimpan pada perangkat keras komputer dan dapat digunakan perangkat lunak untuk memanipulasinya (Sauerwein, 1964-1984).

Menurut (Bratha, 2021) dari definisi ini, terdapat tiga hal yang berhubungan dengan basis data, yaitu sebagai berikut:

1. Data yang terdapat dalam komputer itu sendiri yang diorganisasikan dalam bentuk basis data.

2. Simpanan permanen (*storage*) digunakan untuk menyimpan basis data tersebut. Simpanan ini merupakan salah satu bagian dari teknologi perangkat keras yang digunakan pada sistem informasi. Simpanan permanen pada umumnya berupa sebuah *hard disk*.
3. Perangkat lunak untuk memanipulasi data. Perangkat lunak ini dapat dibuat sendiri dengan menggunakan bahasa pemrograman komputer atau dibeli dalam bentuk suatu paket. Banyak paket perangkat lunak yang disediakan untuk memanipulasi basis data. Paket perangkat lunak ini disebut dengan *database* manajemen sistem.

3.8.1. *PostgreSQL*

Menurut situs resminya *PostgreSQL* adalah sistem manajemen basis data relasional berbasis objek yang berasal dari *POSTGRES* yang dikembangkan di Departemen Ilmu Komputer Universitas *California* di *Berkeley*. Inovasi yang ditawarkan oleh *POSTGRES* telah menjadi cikal bakal bagi banyak konsep yang kemudian baru tersedia dalam beberapa sistem basis data komersial. Sebagai turunan *open-source*, *PostgreSQL* mendukung sebagian besar standar *SQL* dan menawarkan beragam fitur modern seperti kueri kompleks, kunci asing, pemicu, tampilan yang dapat diperbarui, integritas transaksional, dan kontrol konkurensi multiversi. Selain itu, pengguna dapat memperluas fungsionalitas *PostgreSQL* dengan menambahkan jenis data, fungsi, operator, fungsi agregat, metode indeks, dan bahasa prosedural. Keunggulan lainnya adalah lisensi liberal yang memungkinkan siapa pun menggunakan, memodifikasi, dan mendistribusikan *PostgreSQL* secara gratis untuk berbagai tujuan, baik itu kepentingan pribadi, komersial, atau akademis.

3.9. Code Editor

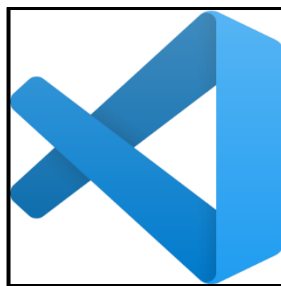
Code editor pada dasarnya adalah editor teks di mana pengembang dapat menulis kode untuk mengembangkan perangkat lunak apapun. *Code* editor juga memungkinkan pengembang untuk menyimpan *file* teks kecil untuk kode tersebut. *Code* editor juga sangat cepat dalam pengoperasian dan memiliki ukuran yang kecil jika dibandingkan dengan *IDE* atau *Integrated development environment*. Bahkan *Code* editor memiliki kemampuan mengeksekusi dan *debug* kode (Setiawan, 2022).

3.9.1. Visual Studio Code

Visual Studio Code (atau disingkat dengan VS Code) adalah *code editor* yang dapat diekstensikan dan perangkat lunak ini adalah perangkat lunak *open source* di bawah lisensi *MIT*.

Cukup unduh dan instal *plugin* yang sesuai untuk menyesuaikannya dengan lingkungan. *VS Code* ditingkatkan dengan fitur seperti integrasi mesin penyelesaian otomatis kode yang kuat (*IntelliSense*), konsol *debugging* dan terminal untuk meluncurkan perintah *server*.

VS Code dirancang dengan sangat baik secara keseluruhan dan keuntungan utamanya adalah mereka menawarkan arsitektur berbasis ekstensi. Karena *IDE* dari *Microsoft* ini cukup ringan, ia pun dapat diperluas dengan menambahkan komponen berurutan sesuai kebutuhan (Setiawan, 2022).



Gambar 3. 11. *Visual Studio Code*

(Sumber <https://github.com/topics/visual-studio-logo>)

3.10. *Library*

Dalam pemrograman, *library* adalah kumpulan kode yang telah ditulis sebelumnya (berisi *function*, *class*, *modul*) dan dapat digunakan kembali untuk menyelesaikan tugas-tugas tertentu.

Tujuan utama dari *library* yaitu menyediakan solusi siap pakai, sehingga *developer* tidak perlu membangun segalanya dari awal. *Library* membantu meningkatkan efisiensi dalam pengembangan *software* dengan mengurangi waktu dan usaha yang diperlukan untuk mengembangkan aspek-aspek dasar sebuah aplikasi.

Penggunaan *library* membantu *developer* untuk lebih fokus pada aspek unik dan inovatif dari proyek, sambil memanfaatkan kode yang telah teruji dan dioptimalkan oleh komunitas *developer* yang lebih luas (Revoupedia, Apa itu Library dalam Pemrograman, n.d.).

3.10.1. *Chakra UI*

Chakra UI merupakan sebuah *library* komponen yang simpel, modular dan mudah diakses untuk memudahkan dan mempercepat pengembang aplikasi untuk membangun tampilan perangkat lunak berbasis *React.js* (Famy, 2022).



Gambar 3. 12. Chakra UI

(Sumber <https://github.com/chakra-ui/chakra-ui/blob/main/README.md>)

3.10.2. *Sweetalert2*

Berdasarkan dari situs resminya *sweetalert2.github.io*. *sweetalert2* adalah penggantian kotak *popup javascript* yang indah responsif dan mudah diakses. *Developer* menyediakan banyak templat disitus resminya yang bisa digunakan secara langsung. Bukan hanya itu pengembang juga menyediakan dokumentasi

secara lengkap untuk melakukan pengeditan. *Tag sweetalert2* di letakan didalam skrip *javascript* (Sosanto et al, 2021).



Gambar 3. 13. Sweetalert2

(Sumber <https://sweetalert2.github.io/>)

3.10.3. *React Chart.JS 2*

Berdasarkan situs resminya *React Chart.js 2* adalah sebuah *library* yang memungkinkan pengembang menggunakan komponen *React* untuk membuat grafik dan visualisasi data dengan mudah menggunakan *Chart.js* versi 2. *Chart.js* sendiri adalah sebuah *library JavaScript* yang populer untuk membuat grafik dan visualisasi data di *web*.

React Chart.js 2 menyediakan komponen-komponen *React* yang memungkinkan penggunaan *Chart.js* menjadi lebih mudah dalam proyek-proyek yang menggunakan *React*. Dengan menggunakan *React Chart.js 2*, pengembang dapat dengan cepat membuat berbagai jenis grafik seperti grafik garis, grafik batang, grafik lingkaran, dan lain-lain, serta mengintegrasikannya dengan aplikasi *web* yang dibangun menggunakan *React*.

3.11. *Unified Modeling Language*

Unified Modeling Language (UML) merupakan sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Adapun konsep analisa dan perancangan berorientasi obyek (Humisar Hasugian, 2012), antara lain :

3.11.1. *Use Case Diagram*

Use case diagram adalah jenis diagram UML (*Unified Modeling Language*) yang menggambarkan fungsi, ruang lingkup, dan interaksi pengguna dengan sistem tersebut. *Diagram use case* memvisualisasikan interaksi antara pengguna (aktor) dan sistem (*use case*), serta tindakan apa saja yang dapat dilakukan aktor terhadap *use case* secara rinci.

Dalam proses pengembangan *website*, aplikasi *mobile*, atau sistem lainnya, *use case diagram* memiliki peranan penting. Sebab, diagram ini mampu mengidentifikasi kebutuhan pengguna, memperjelas persyaratan sistem, dan merancang fungsionalitas *website*.

Dengan adanya *use case diagram*, komunikasi antar tim *web developer*, *project manager*, dan *stakeholder* lain dapat difasilitasi dengan baik. Sehingga, langkah-langkah pengembangan berikutnya, seperti *coding*, *testing*, dan *launching website* ke *web hosting* dapat berjalan lancar.

Jadi, penggunaan *use case diagram* bisa membantu merencanakan fungsionalitas *website*, serta memastikan bahwa pengembangan *website* berjalan sesuai dengan kebutuhan pengguna dan *stakeholder* yang terlibat. (Ariffud, 2023)

Berikut pada gambar 3.14. di bawah ini merupakan simbol *Use case diagram* beserta penjelasannya.

Simbol	Nama	Keterangan
	Actor	Entitas yang berinteraksi dengan sistem
	User case	Aktivitas yang dapat dilakukan actor pada sistem
	Association	Hubungan antara actor dengan use case
	System	Sistem yang sedang dikembangkan
	Include	Suatu use case termasuk bagian dari use case lain
	Extend	Satu use case dapat diperluas dengan use case lain
	Dependency	Ketergantungan antara elemen-elemen diagram
	Generalization	Satu actor atau use case merupakan generalisasi dari yang lain
	Realization	Implementasi dari satu use case oleh yang lain
	Collaboration	Dua atau lebih actor dan use case yang terhubung
	Note	Penjelasan tambahan terkait elemen-elemen diagram
	Anchor	Hubungan teks note dengan elemen-elemen diagram lain

Gambar 3. 14. Simbol *Use Case Diagram*

(Sumber <https://www.niagahoster.co.id/blog/use-case-diagram-adalah/>)

3.11.2. *Entity Relationship Diagram*

Entity Relationship Diagram atau diagram hubungan entitas adalah sebuah diagram yang digunakan untuk perancangan suatu *database* dan menunjukkan relasi atau hubungan antar objek atau entitas beserta atribut-atributnya secara detail. Dengan menggunakan ERD, sistem *database* yang sedang dibentuk dapat digambarkan dengan lebih terstruktur dan terlihat rapi.

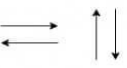











Selain digunakan dalam perancangan *database*, ERD sendiri sering digunakan untuk *debugging database* jika terjadi masalah pada *database*. Untuk melakukan *debug* pada *database* bukanlah hal yang mudah, terlebih lagi jika *database* yang mengalami masalah memiliki banyak tabel dan memerlukan penulisan *SQL* yang kompleks. Dengan menggambarkan skema *database* menggunakan ERD, kamu menjadi lebih mudah untuk menemukan permasalahan yang terjadi dalam *database* dan menyelesaikan masalah dengan mudah. (Setiawan R. , 2021)

3.11.3. Flowchart

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah.

Flowchart berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan *flowchart* dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis..

Berikut pada gambar 3.15. di bawah ini merupakan simbol *flowchart* beserta penjelasannya.

	Flow Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.		Input/output Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	On-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.		Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	Off-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.		Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	Terminator Simbol yang menyatakan awal atau akhir suatu program.		Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	Process Simbol yang menyatakan suatu proses yang dilakukan komputer.		Display Simbol yang menyatakan peralatan output yang digunakan.
	Decision Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.		Preparation Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

Gambar 3. 15. Simbol *Flowchart*

(Sumber <https://www.dicoding.com/blog/wp-content/uploads/2021/07/Simbol-flowchart-1024x685.jpg>)

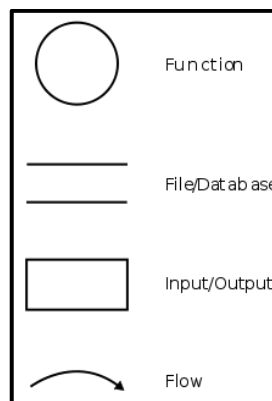
3.11.4. Data Flow Diagram

Data flow diagram atau disingkat *DFD* adalah suatu diagram yang menggambarkan aliran data dari sebuah proses yang sering disebut dengan sistem informasi. Di dalam *data flow* diagram juga menyediakan informasi mengenai input dan output dari tiap entitas dan proses itu sendiri (Adani, Data Flow Diagram (DFD): Pengertian, Jenis, Fungsi & Contoh, 2021).

Dalam diagram alir data juga tidak mempunyai kontrol terhadap *flownya*, sehingga tidak adanya aturan terkait keputusan atau pengulangan. Bentuk penggambaran berupa *data flowchart* dengan skema yang lebih spesifik. Tujuan dari adanya DFD sendiri adalah sebagai penyedia atau menjembatani antara pengguna dengan sistem (Adani, Data Flow Diagram (DFD): Pengertian, Jenis, Fungsi & Contoh, 2021).

1. Notasi Diagram Alir Data

Berikut pada gambar 3.16. di bawah ini merupakan simbol atau notasi dari diagram alir data:



Gambar 3. 16. Simbol DFD

(Sumber <https://www.sekawanmedia.co.id/blog/dfd-adalah/>)

a. Data Store

Merupakan file untuk menyimpan data yang digunakan untuk proses selanjutnya. Dapat dikatakan juga, sama seperti basis data (database). Pada umumnya, data store berupa tabel yang dapat diolah, serta mampu terhubung dengan setidaknya satu masukan

dan satu keluaran. Penggambaran atau simbol data store berupa dua garis sejajar.

b. Data Flow

Merupakan arus data yang mengalir antara terminator, proses, dan data store. Data flow digambarkan dengan simbol tanda panah, dan fungsi utamanya adalah untuk mengalirkan informasi dari satu sistem ke sistem yang lain.

c. External Entity

Lebih sering disebut dengan terminator merupakan pihak di luar sistem, dapat berupa individu, divisi, perusahaan, atau sistem yang lainnya. Terminator dapat memberikan masukan atau keluaran terhadap sistem. Simbol dari external entity dilambangkan dengan persegi panjang atau kotak.

d. Process

Process dilakukan oleh mesin dengan mengubah input menjadi output dengan format yang berbeda. Simbol proses digambarkan dalam bentuk lingkaran, oval, atau persegi panjang dengan tambahan sudut bundar.

(Adani, Data Flow Diagram (DFD): Pengertian, Jenis, Fungsi & Contoh, 2021).

2. Jenis-jenis DFD

Berikut di bawah ini merupakan penjelasan mengenai 3 (tiga) jenis flow diagram:

a. Diagram Level 0 (Diagram Konteks)

Merupakan diagram dengan tingkatan paling rendah, dimana menggambarkan sistem berinteraksi dengan entitas eksternal. Pada diagram konteks akan diberi nomor untuk setiap proses yang berjalan, dimulai dari angka 0 terlebih dahulu.

Jadi, untuk setiap aliran data akan langsung diarahkan menuju sistem. Dan ciri dari diagram level 0 terletak pada tidak adanya informasi yang terkait data yang tersimpan pada data store.

b. Diagram Level 1

Merupakan lanjutan dari diagram konteks, dimana setiap proses yang berjalan akan diperinci pada tingkatan ini. Sehingga, proses utama akan dipecah menjadi sub - sub proses yang lebih kecil lagi.

c. Diagram Level 2

Merupakan tingkat lanjutan dari level yang sebelumnya, dimana pada fase ini akan dijelaskan lebih detail terkait tiap prosesnya. Namun, untuk tingkatan ini jarang sekali dikerjakan dan lebih banyak hanya menerapkan dua level di bawahnya saja.

(Adani, Data Flow Diagram (DFD): Pengertian, Jenis, Fungsi & Contoh, 2021).

3.12. *Object Relational Mapping*

Object Relational Mapping atau ORM adalah sebuah teknik pemrograman untuk mengkonversi data antara sistem *database* relasional menjadi objek-objek dalam bahasa pemrograman yang bersifat *object-oriented*. ORM menciptakan sebuah objek *database* yang bersifat virtual yang dapat dikenali dan digunakan dalam bahasa pemrograman (Adi Nugroho, 2009).

3.12.1. *Prisma ORM*

Menurut situs resminya *Prisma* adalah ORM atau *Object-relational mapping* yang bersifat *open-source*. Komponen-komponennya meliputi *Prisma Client*, *Prisma Migrate*, dan *Prisma Studio*. *Prisma Client* adalah pembangun query yang otomatis dibuat dan tipe aman untuk *Node.js* & *TypeScript*. *Prisma Migrate* adalah sistem migrasi, sedangkan *Prisma Studio* adalah antarmuka grafis untuk melihat dan mengedit data di *database*. *Prisma Studio* adalah satu-satunya

bagian dari *Prisma ORM* yang tidak *open source*, dan hanya dapat dijalankan secara lokal. *Prisma Client* dapat digunakan dalam aplikasi *backend* apa pun, termasuk aplikasi *serverless* dan *microservices*. Ini bisa menjadi *REST API* atau apa pun yang membutuhkan *database*. Setiap proyek yang menggunakan alat dari *toolkit Prisma* dimulai dengan *file Schema Prisma*. *Schema Prisma* memungkinkan pengembang untuk mendefinisikan model aplikasi mereka dalam bahasa pemodelan data yang intuitif.

BAB IV

PEMBAHASAN

4.1. Analisa Sistem Absensi Pegawai

Sistem absensi karyawan di *Tribe Technology Commercialization (TCO) PT. Telkom Indonesia* masih mengandalkan *Google Form*. Namun, penggunaan *form* tersebut menjadi kurang efektif karena keterbatasan fitur seperti input teks, checkbox, pilihan ganda, dan tanggal. Keterbatasan ini membuat kustomisasi *form* absensi menjadi kurang optimal.

Proses rekapitulasi absensi karyawan masih dilakukan secara manual dengan menginputkan data ke *Spreadsheet*. Pendekatan manual ini tidak hanya kurang efisien bagi pihak manajemen tetapi juga meningkatkan risiko kesalahan dalam rekapitulasi data absensi.

Untuk meningkatkan efisiensi, diperlukan sebuah aplikasi berbasis *website* untuk sistem absensi karyawan di *Tribe TCO PT. Telkom Indonesia*. Aplikasi ini akan memungkinkan pihak manajemen dan karyawan untuk melakukan absensi serta rekapitulasi secara efektif. Diharapkan dengan adanya aplikasi ini, proses absensi dan rekapitulasi data absensi dapat dilakukan secara fleksibel, tanpa harus secara manual memasukkan data ke dalam sistem.

Dengan demikian, aplikasi ini diharapkan dapat mempermudah proses absensi dan rekapitulasi bagi semua pihak yang terlibat, serta mengurangi potensi kesalahan dalam pengelolaan data absensi karyawan.

4.2. Analisa Kebutuhan Alat & Perangkat

Setelah menganalisa masalah, tentunya perlu dilakukan analisis kebutuhan yaitu menganalisis apa saja yang dibutuhkan dalam pembuatan website ataupun dalam penggunaannya. Analisis kebutuhan ini dibagi menjadi 3 (tiga), yaitu kebutuhan software, kebutuhan hardware, dan kebutuhan brainware.

4.2.1. Kebutuhan Perangkat Keras (*Hardware*)

Pengertian hardware adalah segala piranti atau komponen dari sebuah komputer yang sifatnya bisa dilihat secara kasat mata dan bisa diraba secara langsung. Dengan kata lain hardware merupakan komponen yang memiliki bentuk nyata. Spesifikasi *hardware* yang digunakan dalam pembuatan sistem absensi karyawan dijelaskan pada tabel 4.1. di bawah ini:

Tabel 4. 1. Analisa kebutuhan perangkat keras (*Hardware*)

No.	Jenis	Keterangan
1.	<i>Processor</i>	AMD Ryzen 7 5800H, Up to 4.4GHz
2.	RAM	8GB
3.	<i>Storage</i>	1TB

Seperti itu, spesifikasi yang dibutuhkan sesuai dengan tabel 4.1 diatas. Selain itu, juga dibutuhkan perangkat pendukung lainnya seperti *keyboard*, *mouse*, dan *monitor*.

4.2.2. Kebutuhan Perangkat Lunak (*Software*)

Selain *Hardware* (perangkat keras) dibutuhkan juga *software* (perangkat lunak) yang dapat menunjang penggunaan perangkat keras tadi. Spesifikasi *software* yang digunakan dalam pembuatan *website* ini, diantaranya :

1. Windows 11 sebagai sistem operasi yang digunakan dalam proses pengerjaan aplikasi berbasis *web*.
2. Visual Studio Code versi 1.85.2 merupakan *software* yang digunakan untuk penulisan *script HTML*, *CSS* maupun *Javascript* dalam pembuatan *website* ini.
3. Javascript sebagai bahasa pemrograman yang digunakan untuk menjalankan *framework* Next.JS 14.1.0.
4. Chakra UI sebagai *library* Javascript yang menyediakan berbagai macam komponen *user interface*.
5. Prisma ORM sebagai *Object-relational mapping* yang berfungsi menginisialisasi dan memetakan struktur data.

6. Postgresql yang difungsikan sebagai *database* dan di autentikasi dan dihubungkan melalui *library Axios*.
7. Draw.io digunakan untuk membuat perancangan *Flowmap* dan *Data Flow Diagram*.
8. Canva adalah *platform* desain grafis yang digunakan untuk membuat grafis media sosial, presentasi, poster, dokumen dan konten visual lainnya. Pada laporan ini menggunakan canva untuk membuat *layout (mockup)* untuk desain *website*.
9. *Web browser* digunakan untuk menampilkan apa yang dihasilkan dari script yang telah dibuat. Selain itu, juga digunakan untuk mencari referensi dan dokumentasi terkait pengerjaan *website*.

4.2.3. Kebutuhan *Brainware*

Brainware adalah orang yang menggunakan atau mengoperasikan sebuah perangkat komputer. Dengan kata lain, *brainware* adalah pengguna/*user* yang menggunakan atau memanajemen atau mengelola *website* ini. Pengguna atau *admin website* tentunya harus dapat mengoperasikan komputer dengan baik. *Brainware* pada pembuatan *website* absensi karyawan ini dibagi menjadi dua jenis *user*, berikut di bawah ini penjelasannya:

1. Admin memiliki hak akses untuk mengelola data absensi karyawan. Admin juga dapat melihat data absensi dan *men-download* laporan hasil rekap *daily summary* dan *monthly summary*.
2. User memiliki hak akses hanya untuk melukan absensi, yaitu mengisi *form* yang ada di halaman absensi pada *website*.

4.3. Perancangan Sistem Absensi Karyawan

Perancangan sistem adalah suatu proses menyusun atau mengembangkan sistem informasi yang baru. Perancangan sistem merupakan tahap selanjutnya setelah Analisa sistem. Setelah mendapatkan gambaran dengan jelas tentang apa yang dikerjakan pada Analisa sistem, dilanjutkan dengan memikirkan bagaimana

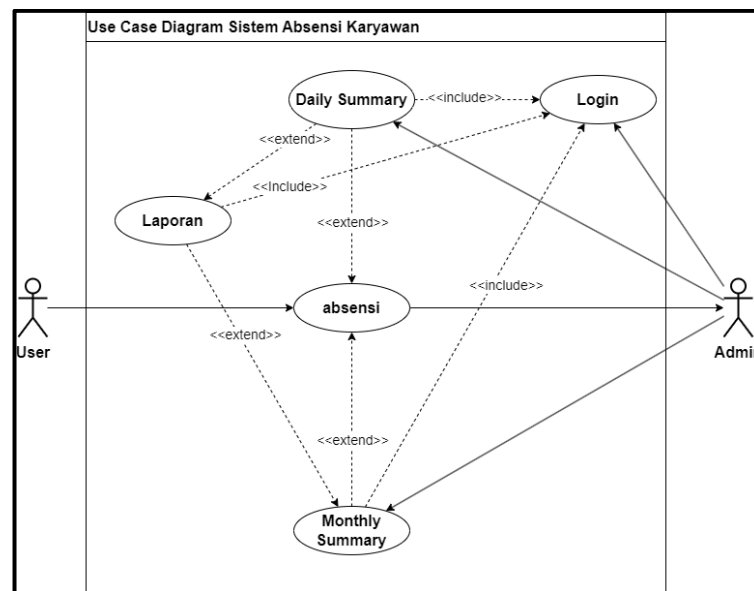
membentuk sistem tersebut agar lebih efektif. Hasil sistem yang dirancang harus sesuai dengan kebutuhan pemakai untuk mendapatkan informasi.

Selanjutnya pada perancangan sistem ini akan dijelaskan mengenai perancangan, Perancangan *Use Case Diagram*, Perancangan *Entity Relationship Diagram*, perancangan *Flowchart*, perancangan *Data Flow Diagram (DFD)*, perancangan *database*, dan perancangan *Layout (Mock Up)*.

4.3.1. Perancangan *Use Case Diagram*

Perancangan *Use Case Diagram* pada pembuatan *website* sistem absensi karyawan digunakan untuk menggambarkan interaksi antar sistem dan aktor (*user*). *Use case diagram* ini berguna untuk pemetaan hak akses aktor dalam pembuatan pengisian absensi, penggunaan fitur *Daily Summary*, *Monthly Summary*, dan fitur cetak laporan. Adapun langkah-langkah dalam pembuatan *use case diagram* sebagai berikut.

Pada gambar 4.1. merupakan *use case diagram* sistem absensi karyawan.



Gambar 4. 1 *Use Case Diagram* Sistem Absensi Karyawan

1. Identifikasi aktor (*user*)

Terdapat beberapa *user* yang terlihat dalam *website* sistem absensi karyawan yaitu Admin sebagai pihak manajemen karyawan, dan Karyawan.

Identifikasi user ini berguna untuk menjelaskan kegiatan/aktivitas yang dilakukan oleh user. Adapun penjelasan dari masing-masing user dapat dilihat pada tabel 4.2.

Tabel 4. 2. Identifikasi User Sistem Absensi Karyawan

No	User	Spesifikasi
1.	Admin	Memiliki hak akses untuk melihat mengedit menggunakan fitur-fitur <i>admin</i> seperti <i>Daily Summary</i> , <i>Monthly Summary</i> dan mencetak laporan.
2.	Karyawan	Memiliki hak akses mengisi <i>form</i> absensi.

2. Identifikasi *use case diagram*

Pada tahap ini dilakukan proses identifikasi use case apa saja yang dilakukan dalam sistem absensi karyawan, setiap use case yang digunakan diberi penjelasan pada spesifikasi kemudian dijelaskan user siapa saja yang terlibat. Tabel 4.3. menunjukan identifikasi use case diagram sistem absensi karyawan.

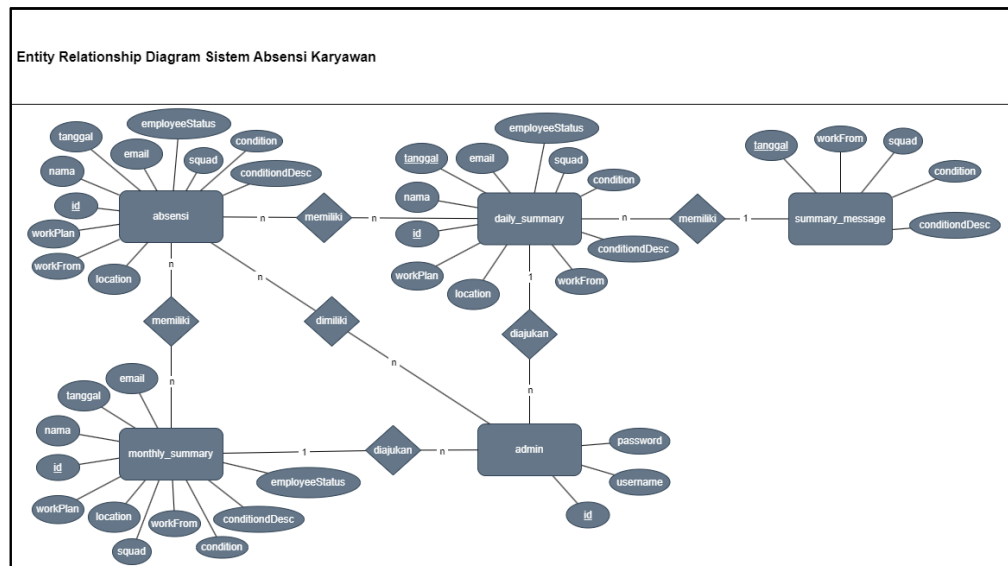
Tabel 4. 3. Identifikasi *use case diagram* Sistem Absensi Karyawan

Use Case	Spesifikasi	User
Absensi	<i>Use case</i> menggambarkan kegiatan mengetikan <i>form</i> absensi yang terdapat pada <i>website</i> .	Karyawan
Login	<i>Use case</i> menggambarkan kegiatan memasukan <i>Username</i> dan <i>password</i> untuk mendapatkan hak akses.	Admin
Daily Summary	<i>Use case</i> menggambarkan kegiatan melakukan rekapitulasi data absensi karyawan yang di filter berdasarkan tanggal absensi dan membuat pesan <i>summary</i> .	Admin
Monthly Summary	<i>Use case</i> menggambarkan kegiatan melakukan rekapitulasi data absensi karyawan per bulan.	Admin
Cetak Laporan	<i>Use case</i> menggambarkan kegiatan melakukan konversi hasil <i>Daily Summary</i> dan <i>Monthly Summary</i> dari database ke sebuah file xls.	Admin

4.3.1. Perancangan Entity Relationship Diagram

Perancangan *Entity Relationship Diagram* (ERD) dibuat untuk menggambarkan bagaimana data saling terkait satu sama lain. Di dalam ERD ini terdapat sebuah entitas, atribut, dan relasinya untuk menghubungkan antar entitas. Maka daripada itu, dalam hal ini peranan ERD yaitu untuk menggambarkan ataupun membuat model suatu basis data yang sederhana. Bukan hanya itu, ERD juga dapat memudahkan orang yang merancang basis data dalam membuat perubahan di dalam basis data.

Berikut gambar perancangan ERD dari aplikasi pengajuan permintaan barang dinjukan pada gambar 4.2. dibawah ini.

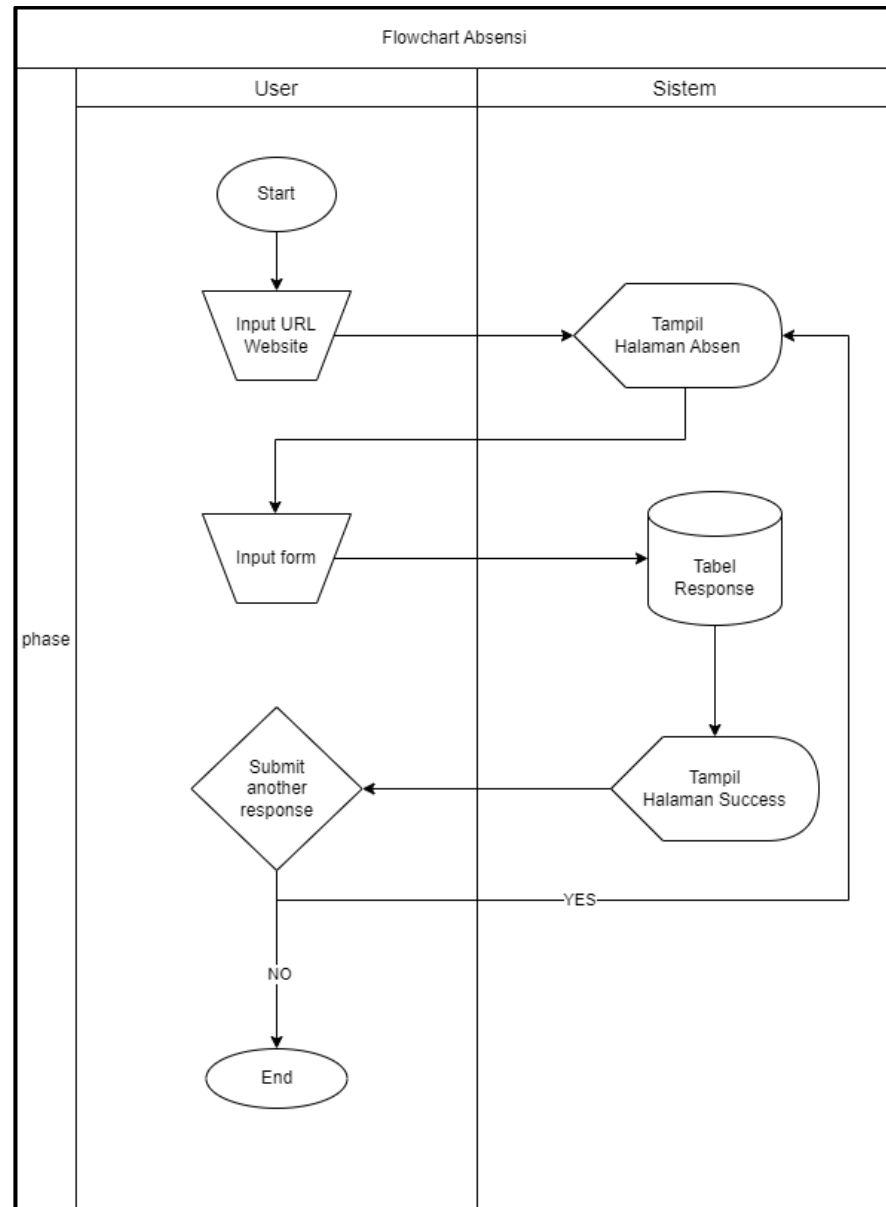


Gambar 4. 2. *Entity Relationship Diagram* Sistem Absensi Karyawan

4.3.2. Perancangan *Flowchart*

1. *Flowchart* absensi

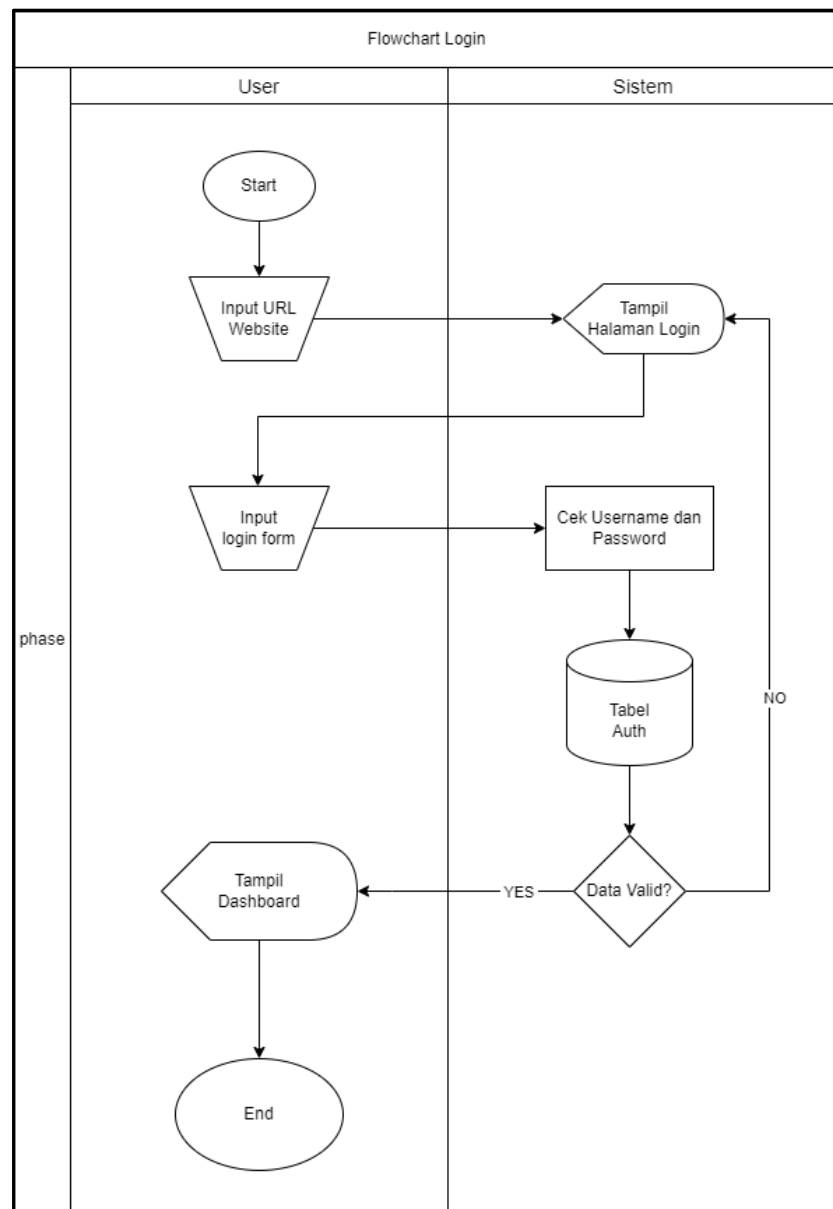
Berikut merupakan *flowschart* dari sistem absensi yang ditunjukkan pada gambar 4.3. di bawah.



Gambar 4. 3. *Flowchart* Absensi

2. Flowchart login

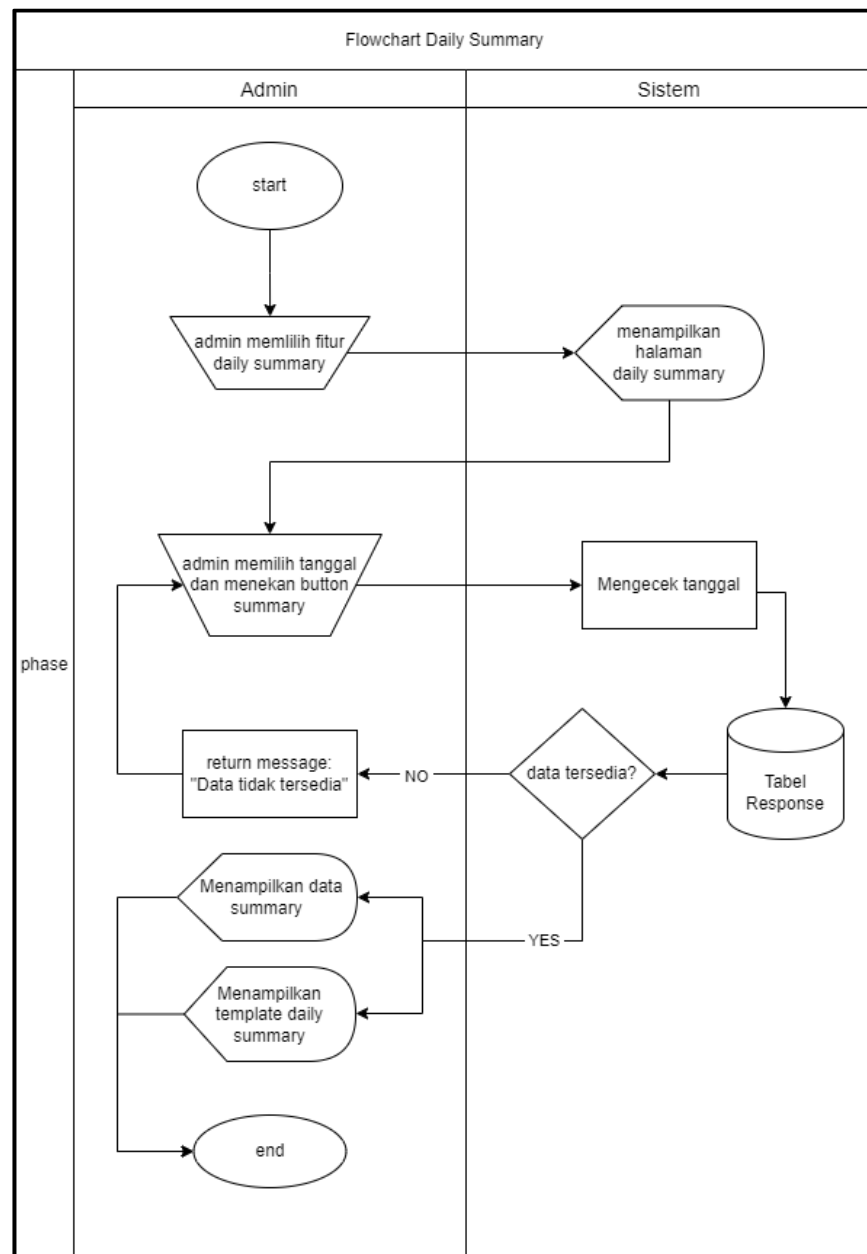
Berikut merupakan *flowchart* dari sistem *login* yang ditunjukkan pada gambar 4.4. di bawah.



Gambar 4. 4. *Flowchart Login*

3. Flowchart daily summary

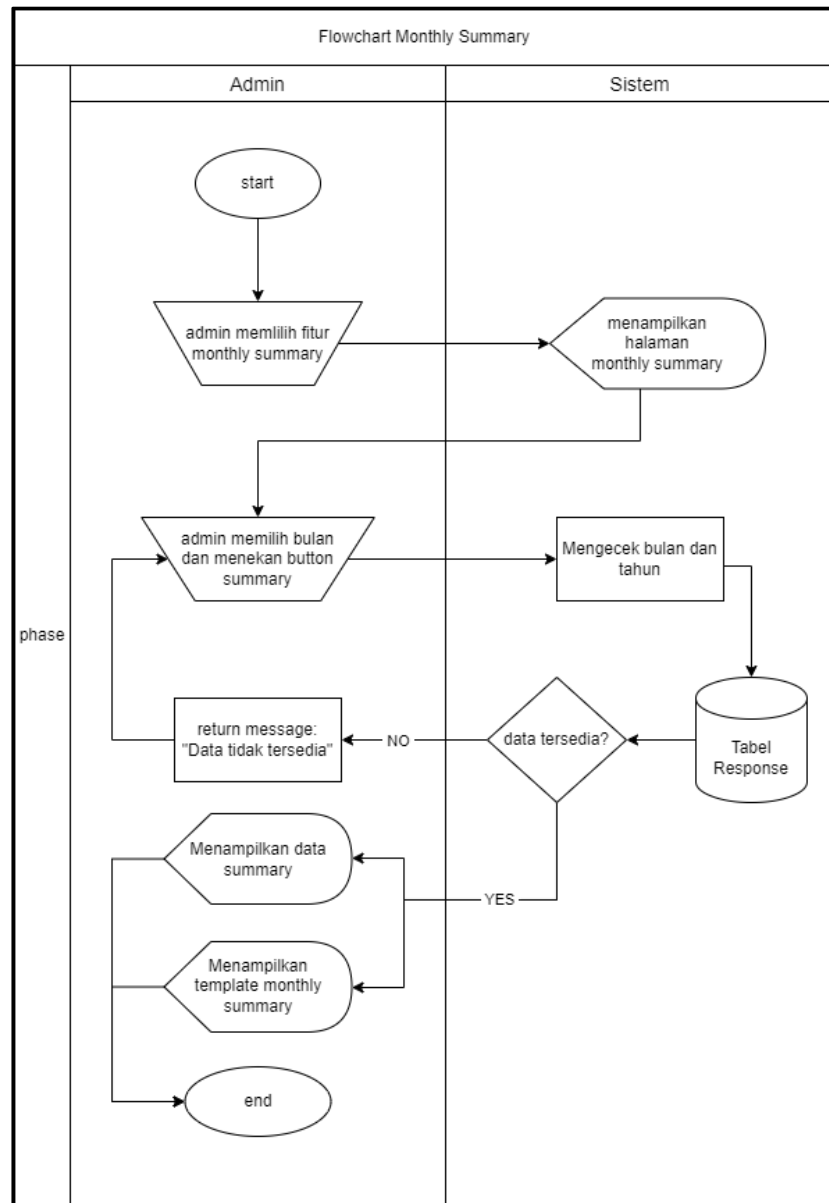
Berikut merupakan *flowchart* dari fitur *daily summary* yang ditunjukkan pada gambar 4.5. di bawah.



Gambar 4. 5. Flowchart Dially Summary

4. Flowchart monthly summary

Berikut merupakan *flowchart* dari fitur *Monthly summary* yang ditunjukkan pada gambar 4.6. di bawah.

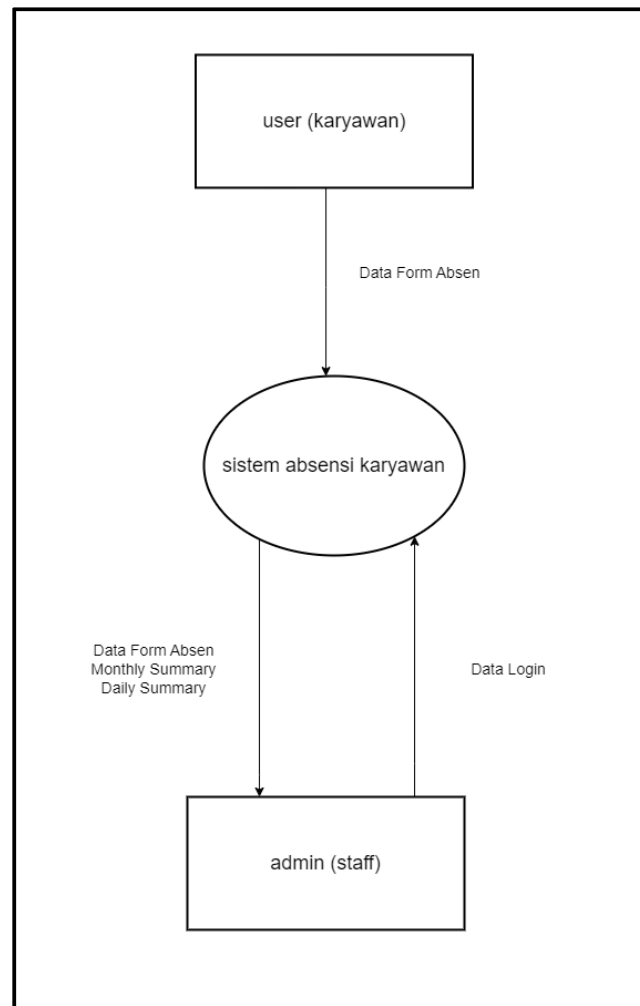


Gambar 4. 6. Flowmap Monthly Summary

4.3.3. Perancangan Data Flow Diagram

Salah satu model dari pembahasan *website* sistem absensi karyawan ini dengan menggunakan pemodelan *Data Flow Diagram* (DFD) level 0 dan DFD level 1.

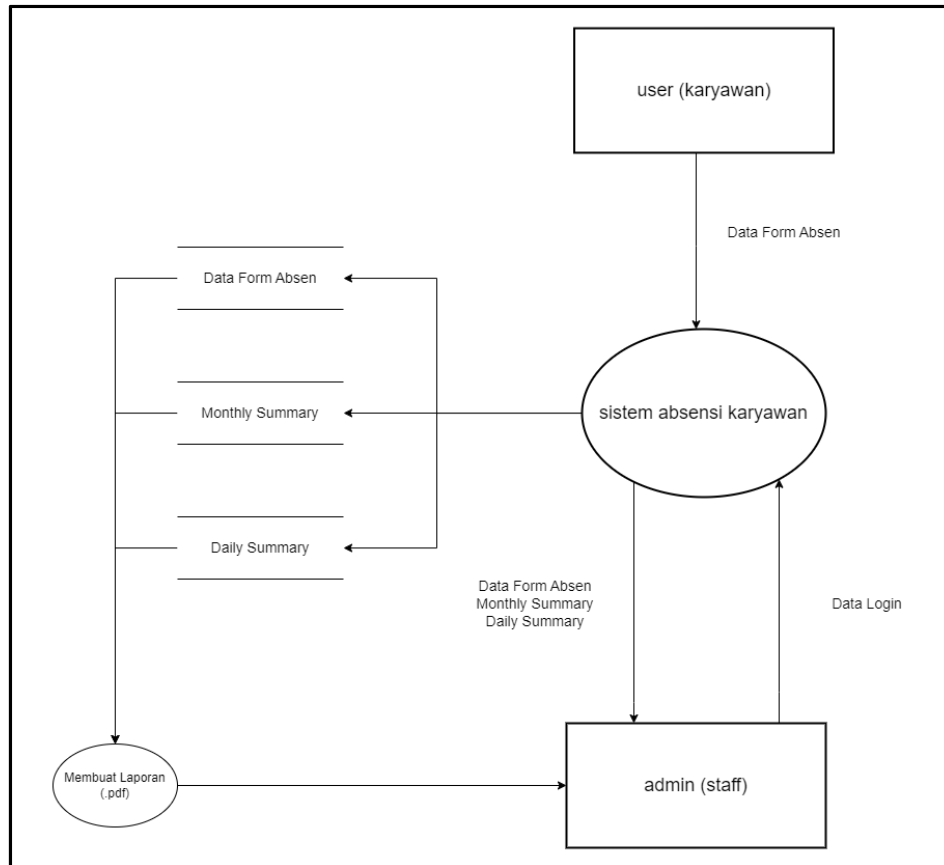
Untuk DFD yang pertama yaitu *context diagram*. Berikut pada gambar 4.7 di bawah ini adalah gambar dari *context diagram*:



Gambar 4. 7. DFD Level 0

Berdasarkan pada gambar 4.7 *context diagram* diatas, menunjukan alur bahwa admin dapat melakukan pengelolaan absensi yaitu *Daily Summary* dan *Monthly Summary*. Untuk user dapat memberikan data form absen pada aplikasi.

Selanjutnya adalah DFD *level 1*, yang dimana semua proses yang ada pada DFD *level 0* dirinci dengan lengkap sehingga lebih lengkap dan detail. Proses-proses utama yang ada akan dipecah menjadi sub proses. Berikut pada gambar 4.8 di bawah ini adalah gambar dari DFD *level 1*:



Gambar 4. 8. DFD *level 1*

Berdasarkan pada gambar 4.8 diatas, menunjukan bahwa DFD *level 1* ini merupakan pengembangan lebih lanjut dari DFD *level 0*. Pada DFD *level 1* ini terlihat bahwa *admin* dapat mengelola data *form* absen, data *daily summary* dan data *monthly summary*.

4.3.4. Perancangan Database

Perancangan database atau basis data merupakan proses yang bertujuan untuk menentukan isi serta pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. Salah satu tahap yang dilakukan dalam perancangan basis data yaitu pembuatan tabel. Berikut merupakan struktur tabel yang digunakan pada pembuatan Website Absensi Karyawan.

1. Tabel Absensi

Tabel ini berfungsi sebagai tempat penyimpanan data dari *user* yang melakukan *input* pada *form* absensi. Tabel absensi terdiri beberapa *field*, seperti *id*, *createdAt*, *email*, *date*, *name*, *employeeStatus*, *squad*, *condition*, *conditionDesc*, *workFrom*, *location*, dan *workPlan*. Berikut merupakan struktur tabel absensi yang ditunjukkan pada tabel 4.4.

Tabel 4. 4. Perancangan basis data tabel absensi

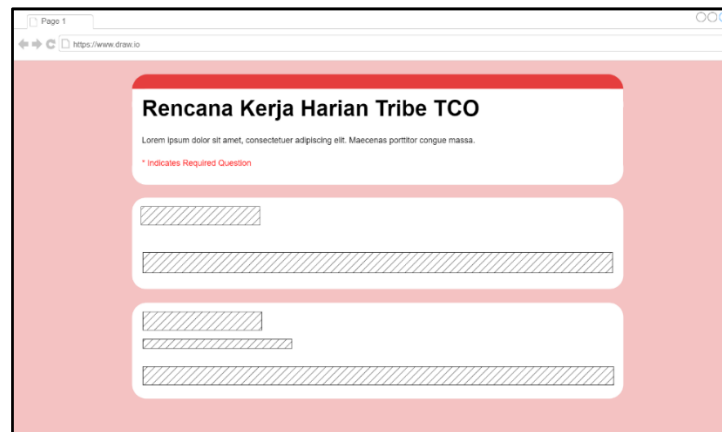
Absensi				
Field	Type	Null	Key	Extra
<i>id</i>	<i>int(11)</i>	<i>NO</i>	<i>PRIMARY KEY</i>	<i>AUTO_INCREMENT</i>
<i>createdAt</i>	<i>Date</i>	<i>NO</i>		<i>DEFAULT(NOW)</i>
<i>email</i>	<i>varchar(128)</i>	<i>NO</i>		
<i>date</i>	<i>Date</i>	<i>NO</i>		
<i>name</i>	<i>varchar(128)</i>	<i>NO</i>		
<i>employeeStatus</i>	<i>varchar(128)</i>	<i>NO</i>		
<i>squad</i>	<i>varchar(128)</i>	<i>NO</i>		
<i>condition</i>	<i>varchar(128)</i>	<i>NO</i>		
<i>conditionDesc</i>	<i>varchar(128)</i>	<i>NO</i>		
<i>workFrom</i>	<i>varchar(128)</i>	<i>NO</i>		
<i>location</i>	<i>varchar(128)</i>	<i>NO</i>		
<i>workPlan</i>	<i>varchar(128)</i>	<i>NO</i>		

4.3.5. Perancangan *Layout (Mock Up)*

Desain layout atau tampilan antarmuka berfungsi untuk menentukan bagaimana tampilan dari website sistem pengajuan permintaan barang.

1. Tampilan Halaman Absensi

Berikut merupakan tampilan perancangan *layout mockup* halaman absensi yang akan ditampilkan padagambar 4.9 di bawah ini.

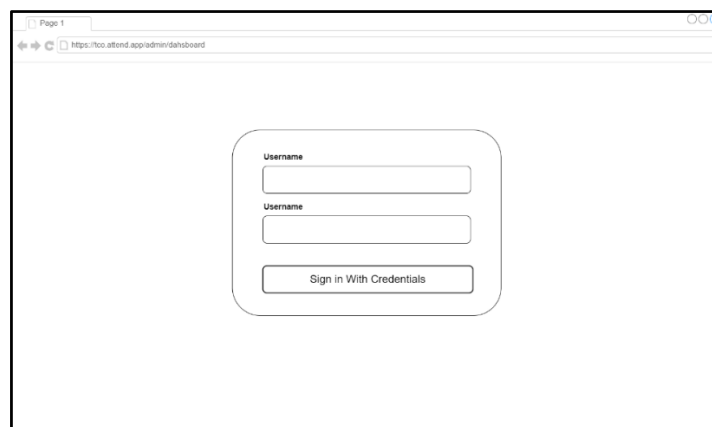


Gambar 4. 9. *Layout* Halaman Absensi

2. Tampilan Halaman *Login*

Tampilan halaman login terdiri dari form username dan password, tombol login yang akan mengalihkan ke halaman dashboard.

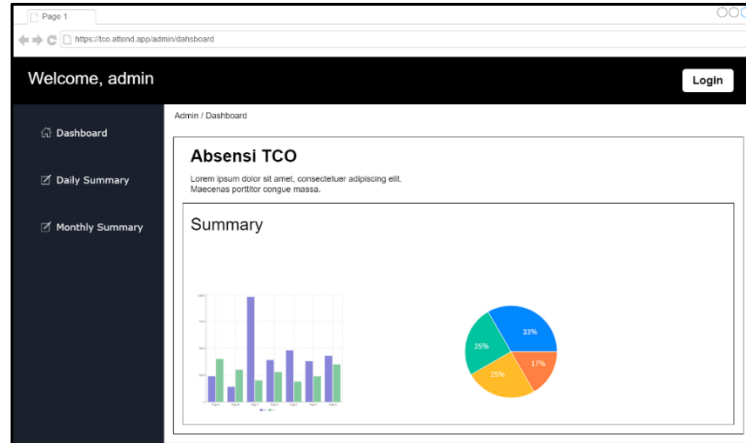
Desain tampilan halaman login dapat dilihat pada gambar 4.10.



Gambar 4. 10. *Layout* Halaman *Login*

3. Tampilan Halaman *Dashboard Admin*

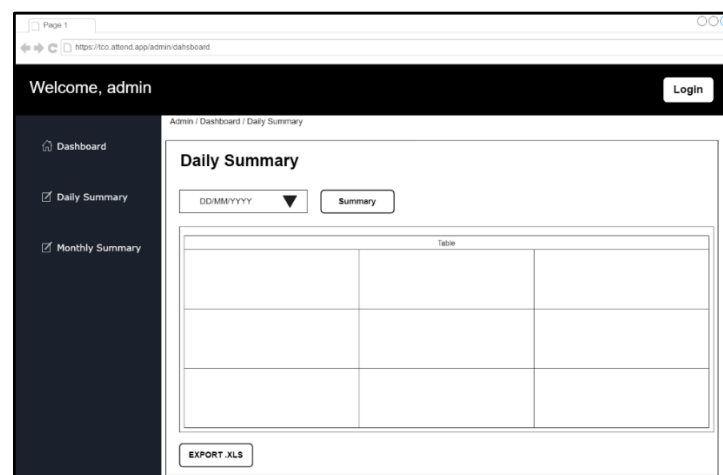
Halaman *dashboard* merupakan halaman yang akan tampil setelah *admin* berhasil *login* ke dalam sistem. Desain tampilan *dashboard admin* dapat dilihat pada gambar 4.11.



Gambar 4. 11. *Layout Halaman Dashboard Admin*

4. Tampilan Halaman *Daily Summary*

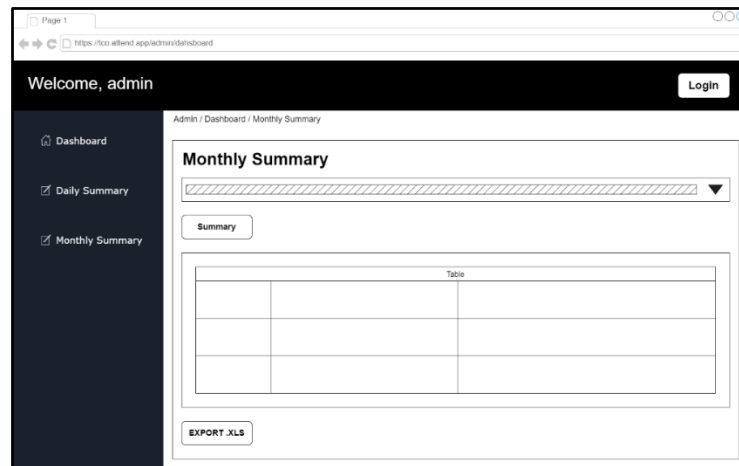
Tampilan halaman *daily summary* adalah halaman yang menampilkan *field data response* yang difilter berdasarkan tanggal, bulan dan tahun. Terdapat fitur *input* untuk menentukan tanggal. Desain tampilan halaman *daily summary* dapat dilihat pada gambar 4.12.



Gambar 4. 12. *Layout Halaman Daily Summary*

5. Tampilan Halaman Monthly Summary

Tampilan halaman monthly summary adalah halaman yang menampilkan field data response yang difilter berdasarkan bulan dan tahun. Terdapat fitur input untuk menentukan bulan. Desain tampilan halaman monthly summary dapat dilihat pada gambar 4.13.

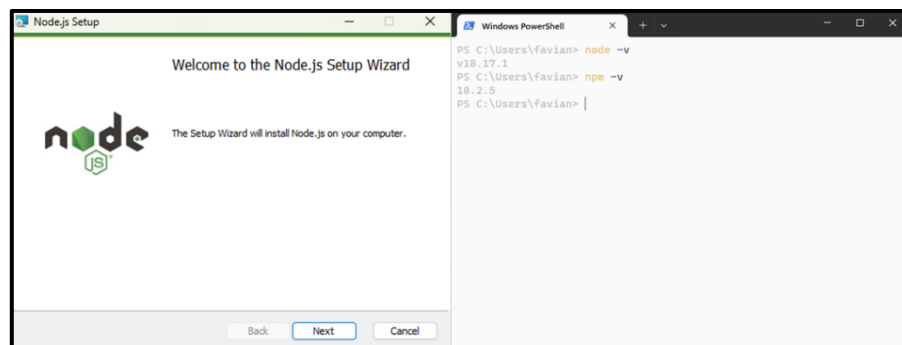


Gambar 4. 13. *Layout Halaman Monthly Summary*

4.4. Persiapan Pembuatan Aplikasi

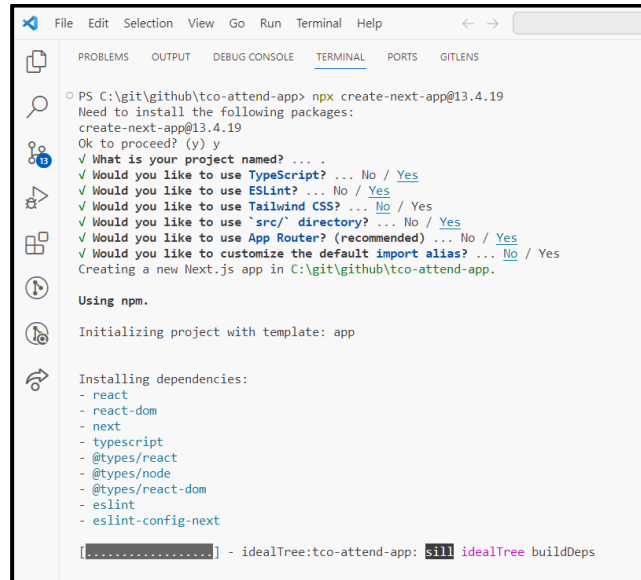
Dalam pembuatan aplikasi sistem absensi karyawan berbasis *website* menggunakan *framework Next.js* 14.1.0. Berikut penjelasan mengenai tahapan yang dilakukan dalam membuat aplikasi berbasis *website* ini:

1. Menginstal terlebih dahulu dan menggunakan Node.JS untuk menginstal beberapa library seperti next.js dan library lainnya. Berikut proses instalasi node.js beserta pengecekan versi Node.JS dapat dilihat di gambar 4.14.



Gambar 4. 14. Instalasi Node.JS

2. *Setup project* dengan membuka *visual studio code* lalu pada *terminal* ketikan “`npx create-next-app@14.1.0`”. Pada *setup project* kali ini penulis memilih menggunakan *Typescript*, *ESLint*, direktori *src* dan menggunakan *App Router*. *Setup project* bisa dilihat pada gambar 4.15. di bawah ini.



```

PS C:\git\github\tco-attend-app> npx create-next-app@13.4.19
Need to install the following packages:
create-next-app@13.4.19
Ok to proceed? (y) y
✓ What is your project named? ... .
✓ Would you like to use TypeScript? ... No / Yes
✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like to use `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the default import alias? ... No / Yes
Creating a new Next.js app in C:\git\github\tco-attend-app.

Using npm.

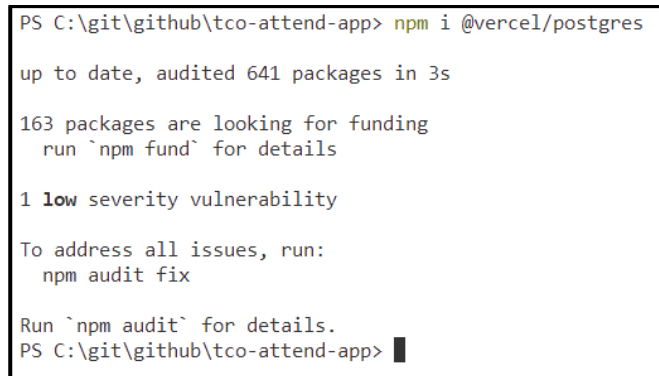
Initializing project with template: app

Installing dependencies:
- react
- react-dom
- next
- typescript
- @types/react
- @types/node
- @types/react-dom
- eslint
- eslint-config-next

[.....] - idealTree:tco-attend-app: sill idealTree buildDeps
  
```

Gambar 4. 15. *Setup Project*

3. Menginstal beberapa *library* dan dependensi melalui *terminal*. Library dan dependensi yang disebut diantaranya:
- Postgres*, hal pertama yang penulis adalah melakukan instalasi *postgres* sebagai *client library*, untuk menghubungkan *postgresql* dengan project seperti pada gambar 4.16.



```

PS C:\git\github\tco-attend-app> npm i @vercel/postgres

up to date, audited 641 packages in 3s

163 packages are looking for funding
  run `npm fund` for details

1 low severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
PS C:\git\github\tco-attend-app>
  
```

Gambar 4. 16. Instalasi *library postgres*

- b. Berikutnya menginstal library Axios. Library ini berfungsi untuk menjembatani permintaan dari website ke *database* untuk melakukan reading ataupun writing. Instalasi library Axios dapat dilihat pada gambar 4.17.

```
PS C:\git\github\tco-attend-app> npm i axios
added 9 packages, and audited 634 packages in 2s

163 packages are looking for funding
  run `npm fund` for details

1 low severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
PS C:\git\github\tco-attend-app>
```

Gambar 4. 17. Instalasi library axios

- c. Setelah instalasi *library* Axios selesai, selanjutnya melakukan instalasi *prisma* dan *prisma client*, untuk melakukan *mapping* pada *database postgresql*. Juga digunakan sebagai alat migrasi *database*. Proses instalasi *prisma* dan *prisma client* terdapat pada gambar 4.18.

```
PS C:\git\github\tco-attend-app> npm i -D prisma
added 6 packages, and audited 640 packages in 31s

163 packages are looking for funding
  run `npm fund` for details

1 low severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
PS C:\git\github\tco-attend-app> npm i @prisma/client
added 1 package, and audited 641 packages in 6s

163 packages are looking for funding
  run `npm fund` for details

1 low severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
PS C:\git\github\tco-attend-app>
```

Gambar 4. 18. Instalasi *library prisma* dan *prisma client*

- d. Selanjutnya, setelah *prisma* berhasil terinstal, dilanjutkan dengan menginstal *library Chakra UI* dan *Sweetalert2* sebagai penyedia komponen *user interface*. Proses instalasi bisa dilihat pada gambar 4.19. di bawah ini.

```
PS C:\git\github\tco-attend-app> npm i @chakra-ui/react @chakra-ui/next-js @emotion/react @emotion/styled framer-motion
up to date, audited 602 packages in 11s
155 packages are looking for funding
  run `npm fund` for details
1 low severity vulnerability
To address all issues, run:
  npm audit fix
Run `npm audit` for details.
PS C:\git\github\tco-attend-app> npm i sweetalert2
up to date, audited 602 packages in 1s
155 packages are looking for funding
  run `npm fund` for details
1 low severity vulnerability
To address all issues (including breaking changes), run:
  npm audit fix --force
Run `npm audit` for details.
PS C:\git\github\tco-attend-app> █
```

Gambar 4. 19. Instalasi *library Chakra UI* dan *Sweetalert2*

- e. Setelah selesai menginstal *Chakra UI* dan *sweetalert*, dilanjutkan dengan menginstal *library React Chart JS 2* sebagai penyedia komponen diagram untuk mempresentasikan data di *website* ini. Proses instalasi bisa dilihat pada gambar 4.20. di bawah ini.

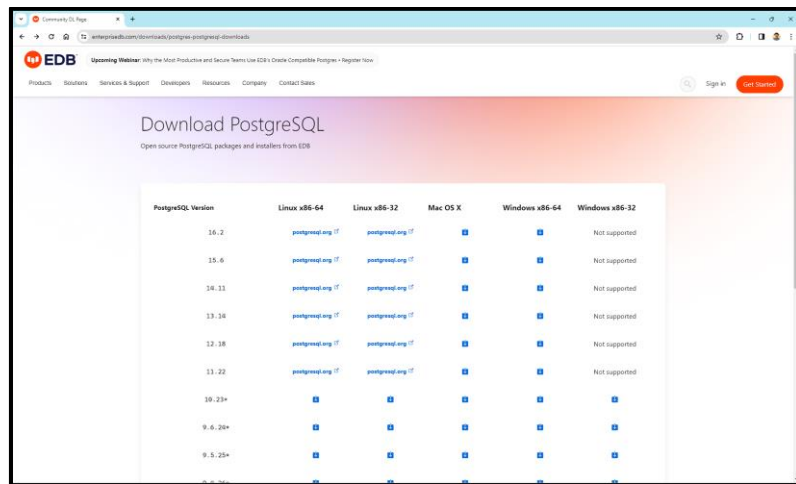
```
PS C:\git\github\tco-attend-app> npm i react-chartjs-2
up to date, audited 641 packages in 2s
163 packages are looking for funding
  run `npm fund` for details
1 low severity vulnerability
To address all issues, run:
  npm audit fix
Run `npm audit` for details.
PS C:\git\github\tco-attend-app> █
```

Gambar 4. 20. Instalasi *library React Chart JS 2*

4.5. Pembuatan Basis Data

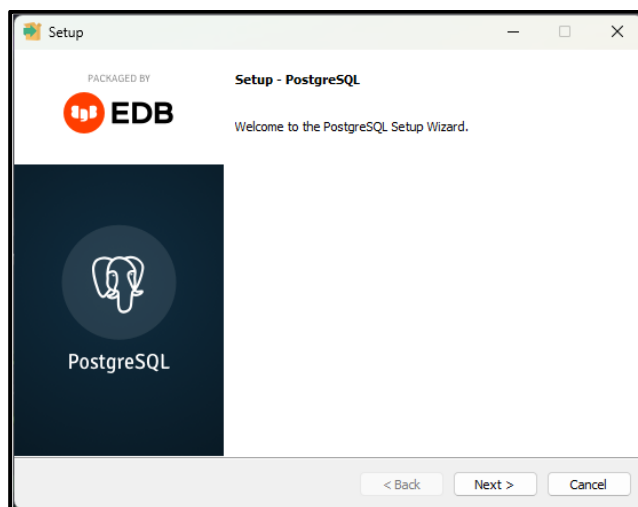
Membuat basis data dengan *PostgreSQL* dan *Prisma* sebagai *ORM* dengan menginstal *PostgreSQL* terlebih dahulu.

Hal yang pertama dilakukan adalah mengunduh aplikasi *PostgreSQL* di *website* resmi, pada proyek kali ini penulis mengunduh *PostgreSQL* versi 15.6. Berikut pada gambar 4.21. merupakan proses mengunduh *PostgreSQL* di *website* resmi.

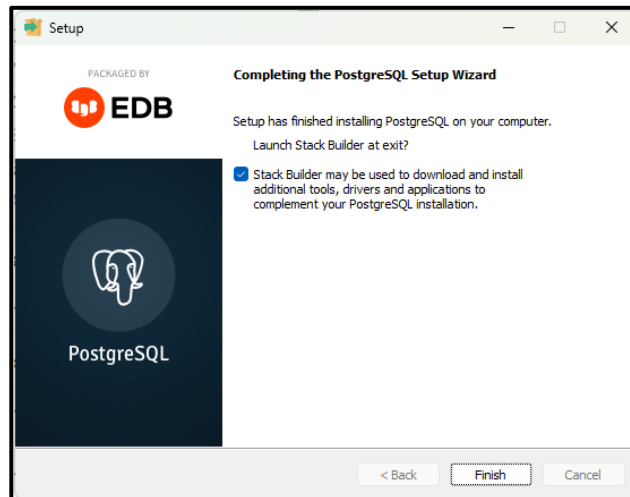


Gambar 4. 21. Mengunduh *PostgreSQL*

Instalasi dilakukan dengan menjalankan aplikasi *setup PostgreSQL* yang sudah di unduh sebelumnya. *Setup PostgreSQL* dapat dilihat pada gambar 4.22. dan gambar 4.23.



Gambar 4. 22. Instalasi *PostgreSQL*



Gambar 4. 23. Instalasi *PostgreSQL*

Setelah selesai melakukan instalasi *PostgreSQL*, dilanjutkan dengan melakukan inisialisasi pada *prisma*. Inisialisasi dilakukan dengan mengetikkan perintah “*npm prisma init*” yang ditampilkan pada gambar 4.24.

```
PS C:\git\github\tco-attend-app> npm prisma init

✓ Your Prisma schema was created at prisma/schema.prisma
  You can now open it in your favorite editor.

warn Prisma would have added DATABASE_URL but it already exists in .env
warn You already have a .gitignore file. Don't forget to add '.env' in it to not commit any private information.

Next steps:
1. Set the DATABASE_URL in the .env file to point to your existing database. If your database has no tables yet,
   read https://pris.ly/d/getting-started
2. Set the provider of the datasource block in schema.prisma to match your database: postgresql, mysql, sqlite,
   sqlserver, mongodb or cockroachdb.
3. Run prisma db pull to turn your database schema into a Prisma schema.
4. Run prisma generate to generate the Prisma Client. You can then start querying your database.

More information in our documentation:
https://pris.ly/d/getting-started

PS C:\git\github\tco-attend-app> █
```

Gambar 4. 24. Inisialisasi *prisma*

Perintah tersebut akan melakukan inisialisasi yang prosesnya menciptakan *file* baru yaitu *schema.prisma* dan *.env*.

Pada *file schema.prisma* berisikan konfigurasi utama dari *prisma*, di dalamnya terdapat konfigurasi *client* dan *database*. Isi dari *file schema.prisma* ditampilkan pada gambar 4.25.

```

prisma > schema.prisma
1 // This is your Prisma schema file,
2 // learn more about it in the docs: https://pris.ly/d/prisma-schema
3
4 generator client {
5   provider = "prisma-client-js"
6 }
7
8 datasource db {
9   provider = "postgresql"
10  url      = env("DATABASE_URL")
11 }
12
13 model Absensi {
14   id Int @id @default(autoincrement())
15   createdAt DateTime @default(now())
16   email String @unique
17   date DateTime
18   name String
19   employeeStatus String
20   squad String
21   condition String
22   conditionDesc String
23   workFrom String
24   location String
25   workPlan String
26 }
27
28 model Auth {
29   id Int @id @default(autoincrement())
30   username String
31   password String
32 }

```

Gambar 4. 25. Isi file *schema.prisma*

Pada file *.env* berisikan *database key* untuk menghubungkan *website* dengan *database PostgreSQL*. Isi dari file *.env* dapat dilihat pada gambar 4.26.

```

1 DATABASE_URL="postgresql://postgres:favian899@localhost:5432/tco-attend-db?schema=public"

```

Gambar 4. 26. Isi file *.env*

Selanjutnya adalah melakukan migrasi *database* menggunakan *prisma*. Migrasi *prisma* ini bertujuan untuk mengeksekusi file *schema.prisma*, membuat *database* baru pada *PostgreSQL* yang berdasarkan konfigurasi dari file *schema.prisma* tersebut. Untuk melakukan migrasi menggunakan *prisma* adalah dengan mengetikkan “*npx prisma migrate dev*” pada *terminal*. Proses migrasi bisa dilihat pada gambar 4.27.

```

PS C:\git\github\tco-attend-app> npx prisma migrate dev
Environment variables loaded from .env
Prisma schema loaded from prisma\schema.prisma
Datasource "db": PostgreSQL database "tco-attend-db", schema "public" at "localhost:5432"

PostgreSQL database tco-attend-db created at localhost:5432

✓ Enter a name for the new migration: ... migrasi 1
Applying migration `20240205000127_migrasi_1`

The following migration(s) have been created and applied from new schema changes:

migrations/
├─ 20240205000127_migrasi_1/
└─ migration.sql

Your database is now in sync with your schema.

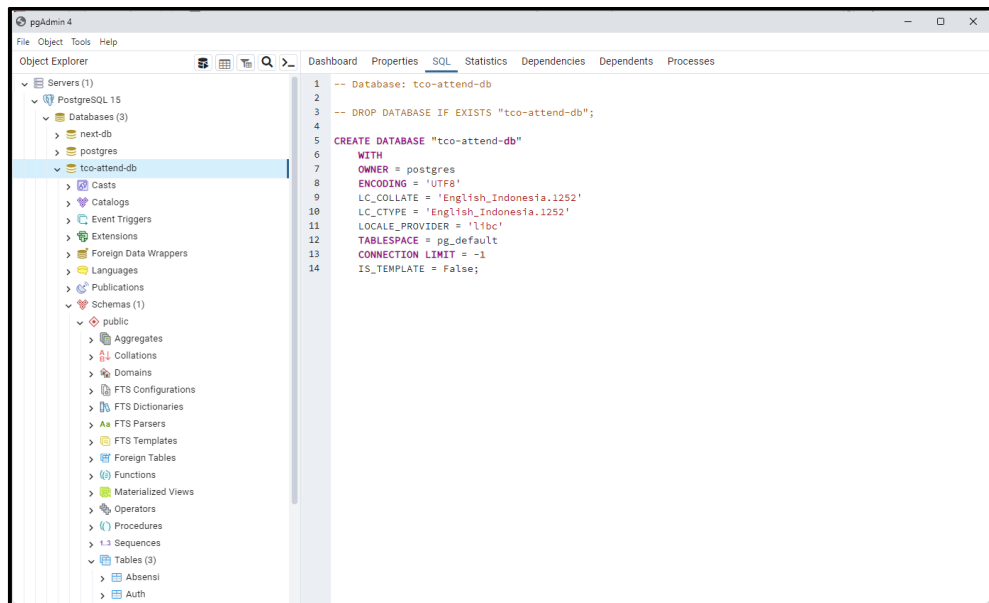
✓ Generated Prisma Client (v5.9.1) to .\node_modules\@prisma\client in 134ms

PS C:\git\github\tco-attend-app>

```

Gambar 4. 27. Proses migrasi *database* menggunakan *prisma*

Pada saat melakukan migrasi *Prisma* akan membuat *database* baru pada *PostgreSQL* dengan nama *database tco-attend-db* yang berjalan pada *localhost:5432*. Berikut pada gambar 4.28. menampilkan *database* pada aplikasi *PostgreSQL*.



Gambar 4. 28. Database pada *PostgreSQL*

4.6. Pembuatan Antarmuka Website

Implementasi antarmuka menggambarkan tampilan dari aplikasi yang dibangun sistem absensi karyawan berbasis *website*. Berikut merupakan implementasi antarmuka pada *web* absensi karyawan.

4.5.1. Halaman Absensi

Halaman absensi merupakan halaman yang menampilkan *form* pengisian absensi. Pada halaman absensi terdapat dua halaman berbeda, yaitu halaman absensi dan halaman sukses.

Berikut pada gambar 4.29. merupakan tampilan *code* untuk halaman absensi.

```

27 const AbsensiPage: React.FC = () => {
28   const [email, setEmail] = useState("");
29   const [date, setDate] = useState("");
30   const [name, setName] = useState("");
31   const [employeeStatus, setEmployeeStatus] = useState("");
32   const [squad, setSquad] = useState("");
33   const [condition, setCondition] = useState("");
34   const [conditionDesc, setConditionDesc] = useState("");
35   const [workFrom, setWorkFrom] = useState("");
36   const [location, setLocation] = useState("");
37   const [workPlan, setWorkPlan] = useState("");
38
39   const router = useRouter();
40
41   const handleSubmit = async (e: SyntheticEvent) => {
42     e.preventDefault();
43     try {
44       // const response = await axios.post("http://localhost:3000/api/submit", {
45       const response = {
46         email: String(email),
47         date: Date(),
48         name: String(name),
49         employeeStatus: String(employeeStatus),
50         squad: String(squad),
51         condition: String(condition),
52         conditionDesc: String(conditionDesc),
53         workFrom: String(workFrom),
54         location: String(location),
55         workPlan: String(workPlan),
56       };
57       const submittedData = {
58         email: email,
59         date: date,
60         name: name,
61         employeeStatus: employeeStatus,
62         squad: squad,
63         condition: condition,
64         conditionDesc: conditionDesc,
65         workFrom: workFrom,
66         location: location,
67         workPlan: workPlan,
68       };
69       setEmail("");
70       setDate("");
71       setName("");
72       setEmployeeStatus("");
73       setSquad("");
74       setCondition("");
75       setConditionDesc("");
76       setWorkFrom("");
77       setLocation("");
78       setWorkPlan("");
79       router.push("/absensi/success");
80     } catch (error) {
81       console.log(error);
82     }
83   };
84 }

```

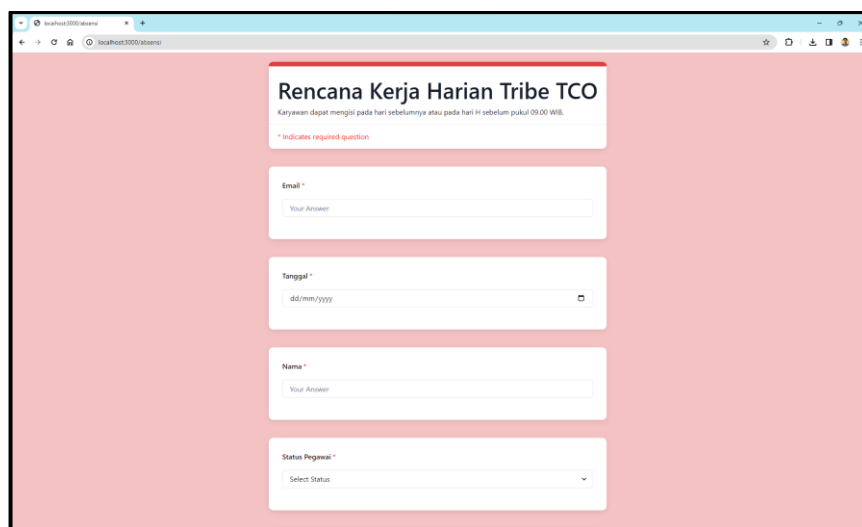
Gambar 4. 29. Isi *code* halaman absensi

Halaman absensi merupakan halaman yang bertujuan untuk karyawan mengisi form-form yang diperlukan. Terdapat beberapa form diantaranya :

1. *Form email* adalah *form* yang nantinya akan diisi oleh pengguna berupa teks *email*. *Form email* ini membutuhkan tanda "@" atau "at" saat diisi.
2. *Form tanggal* adalah *form* yang mengharuskan pengguna memasukkan tanggal sebagai *input*.
3. *Form nama* adalah *form* yang mengharuskan pengguna memasukkan nama lengkap.
4. *Form status pegawai* merupakan *form dropdown* yang mengharuskan pengguna memilih beberapa opsi, antara lain Karyawan Tetap, *Digital Talent* atau *Pro Hire*, dan Tenaga Kerja Penunjang.
5. *Form squad* adalah *form dropdown* yang mengharuskan pengguna memilih beberapa opsi, antara lain *Tribe Leader*, *Business Collaboration*, *Product Onboard Business Operational*, dan *Project & Platform*.

6. *Form* kondisi badan adalah *form dropdown* yang mengharuskan pengguna memilih beberapa opsi, di antaranya Sehat, Kurang Sehat (masih bisa bekerja), dan Sakit (perlu istirahat).
7. *Form* keterangan kondisi merupakan *form* lanjutan dari *form* kondisi yang mengharuskan pengguna untuk menjelaskan kondisinya saat mengisi formulir tersebut. Jika pengguna dalam keadaan sehat, maka pengguna hanya perlu memilih opsi "sehat" yang tertera pada *input radio*.
8. *Form* informasi kerja adalah *form* yang mengharuskan pengguna memilih beberapa opsi, di antaranya *Work From Office* (WFO) atau *Customer Visit*, *Work From Anywhere* (WFA), dan Cuti.
9. *Form* lokasi dan kota adalah *form* yang mengharuskan pengguna mengisi lokasi saat mengakses *website* tersebut.
10. *Form* rencana kerja atau aktivitas adalah *form* yang mengharuskan pengguna mengisi rencana kerja atau aktivitas pada hari itu.

Berikut pada gambar 4.30. merupakan penampilan dari halaman *form* absensi.



Rencana Kerja Harian Tribe TCO

Karyawan dapat mengisi pada hari sebelumnya atau pada hari ini sebelum pukul 09:00 WIB.

* Indicates required question

Email *

Your Answer

Tanggal *

dd/mm/yyyy

Nama *

Your Answer

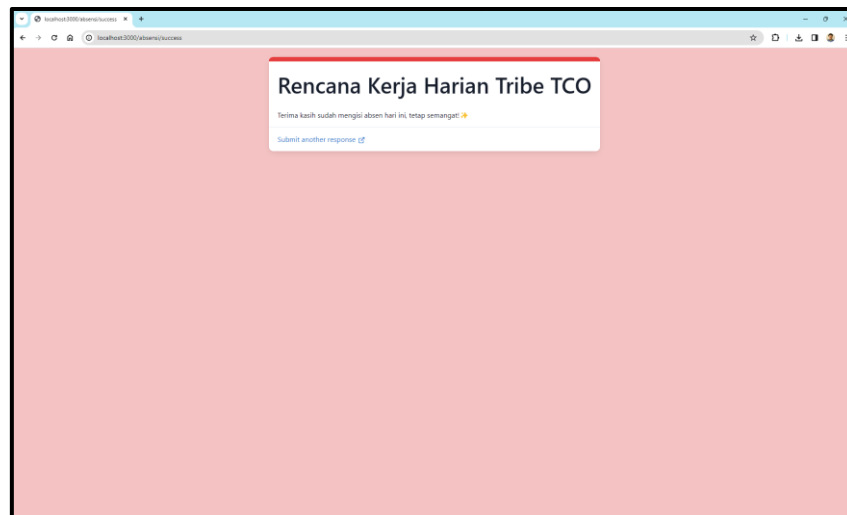
Status Pegawai *

Select Status

Gambar 4. 30. Halaman *form* absensi

Pada halaman sukses merupakan halaman yang menampilkan pesan bahwa absensi telah berhasil di rekam. Halaman ini muncul setelah pengguna selesai memasukan semua *form* yang ada pada halaman absen dan setelah pengguna menekan tombol *submit*.

Berikut pada gambar 4.31. adalah tampilan halaman sukses jika karyawan berhasil melakukan *submit* pada halaman absen.



Gambar 4. 31. Halaman sukses ketika berhasil absen

4.5.2. Halaman *Login*

Halaman *login* adalah halaman yang muncul ketika pengguna mencoba untuk masuk sebagai *admin*. Berikut pada gambar 4.32. merupakan isi *code* halaman *login*.

```

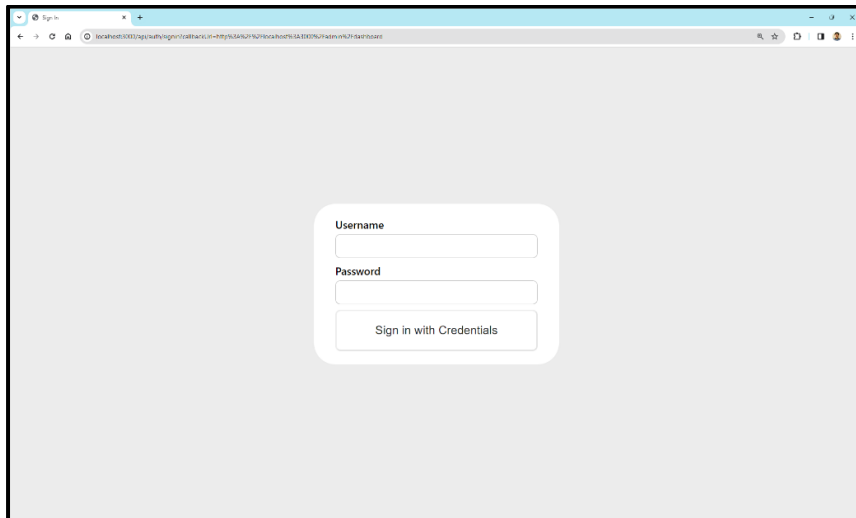
1  const authOptions: NextAuthOptions = {
2    session: {
3      strategy: "jwt",
4    },
5    secret: "admin123",
6    providers: [
7      CredentialsProvider({
8        type: "credentials",
9        name: "credentials",
10       credentials: {
11         username: { label: "Username", type: "text" },
12         password: { label: "Password", type: "password" },
13       },
14     }),
15   ],
16   async authorize(credentials, req) {
17     const { username, password } = credentials as {
18       username: string;
19       password: string;
20     };
21     const user = await db.user.findFirst({
22       where: {
23         username,
24         password,
25       },
26     });
27     if (!user) {
28       return null;
29     }
30     if (user.username === "admin" && user.password === "admin123") {
31       return user;
32     }
33     return null;
34   },
35   callbacks: {
36     async jwt({ token, account, profile, user }) {
37       if (!account || account.provider !== "credentials") {
38         return token;
39       }
40       token.id = user.id;
41       token.name = user.name;
42       token.email = user.email;
43       token.role = user.role;
44       return token;
45     },
46   },
47 };

```

Gambar 4. 32. Isi *code* untuk halaman *login*

Di halaman ini, pengguna diminta untuk memasukkan *username* dan *password*. Jika pengguna berhasil melakukan *login*, mereka akan dialihkan ke halaman selanjutnya, yaitu halaman *dashboard*.

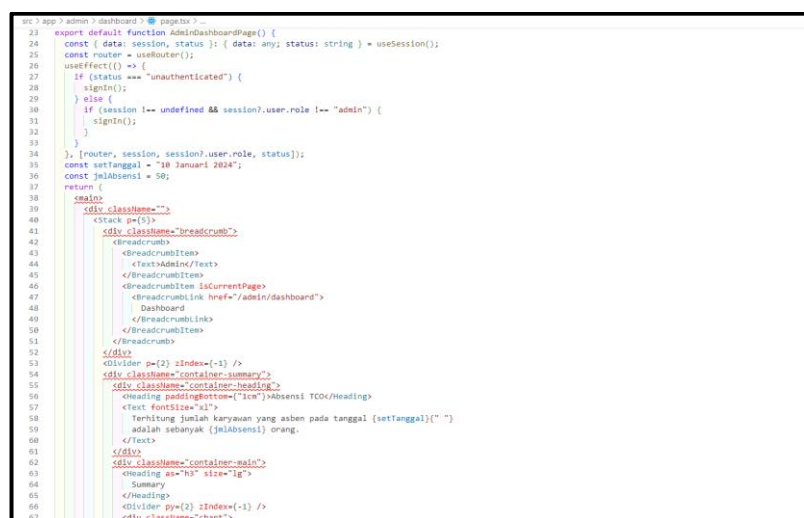
Berikut pada gambar 4.33. adalah tampilan dari halaman *login*.



Gambar 4. 33. Halaman *login*

4.5.3. Halaman *Dashboard*

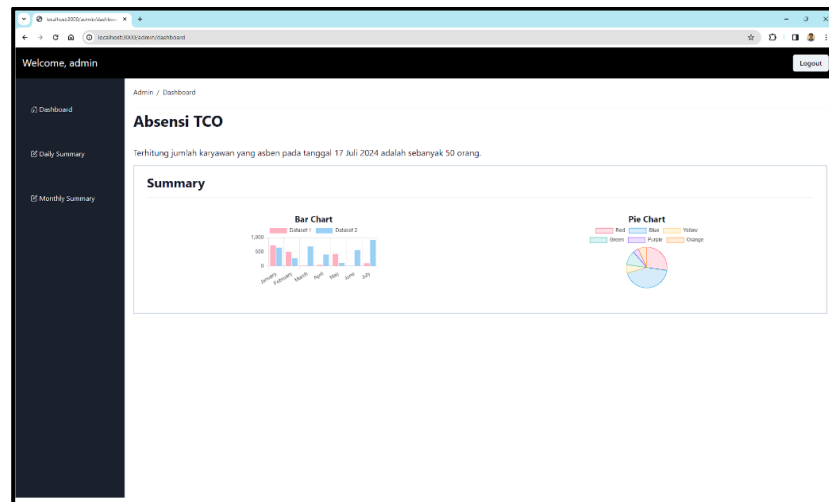
Halaman *Dashboard* merupakan halaman yang ditampilkan pertama kali jika pengguna berhasil *login* sebagai *admin*. Berikut ini pada gambar 4.34. merupakan isi code halaman dashboard.



Gambar 4. 34. Isi code halaman *dashboard*

Pada halaman *dashboard* terdapat *navbar* yang berada paling atas tampilan *website* dan *sidebar* pada bagian kiri tampilan *website* serta bagian utama yang menampilkan judul “Absensi TCO”.

Berikut pada gambar 4.35. merupakan tampilan dari halaman *dashboard*.



Gambar 4. 35. Tampilan halaman *dashboard*

Pada bagian kiri *navbar* menampilkan pesan “Welcome, admin” dan pada bagian kanan *navbar* terdapat sebuah tombol *logout* untuk *admin* melakukan *logout*.

Pada *sidebar* terdapat tiga navigasi, yaitu diantaranya *dashboard* untuk mengarah ke halaman *dashboard*, *daily summary* untuk mengarah ke halaman *daily summary*, dan tombol *monthly summary* untuk mengarah ke halaman *monthly summary*.

Pada bagian utama menampilkan sebuah *breadcrumb*. *Breadcrumb* adalah elemen *web* yang memiliki fungsi sebagai navigasi halaman di dalam *web*. Selain *breadcrumb*, terdapat judul halaman yang bertuliskan "Absensi TCO", yang diikuti oleh keterangan mengenai jumlah karyawan yang telah mengisi *form* absensi pada tanggal sekian. Di samping itu, terdapat juga sebuah *summary* yang menampilkan dua diagram data, yaitu diagram batang dan diagram *pie*.

Diagram batang menunjukkan total jumlah karyawan yang telah mengisi *form* absensi berdasarkan hari atau tanggal. Sementara itu, diagram *pie*

menampilkan jumlah karyawan yang mengisi *form* absensi, yang diurutkan berdasarkan *squad*-nya.

4.5.4. Halaman *Daily Summary*

Halaman *diary summary* merupakan halaman bagian dari *dashboard*. Pada halaman *daily summary* ini terdapat sebuah *navbar* dan *sidebar* yang sama dengan yang ada pada halaman *dashboard*.

Berikut pada gambar 4.36. merupakan isi code halaman dan fitur *daily summary*.

```

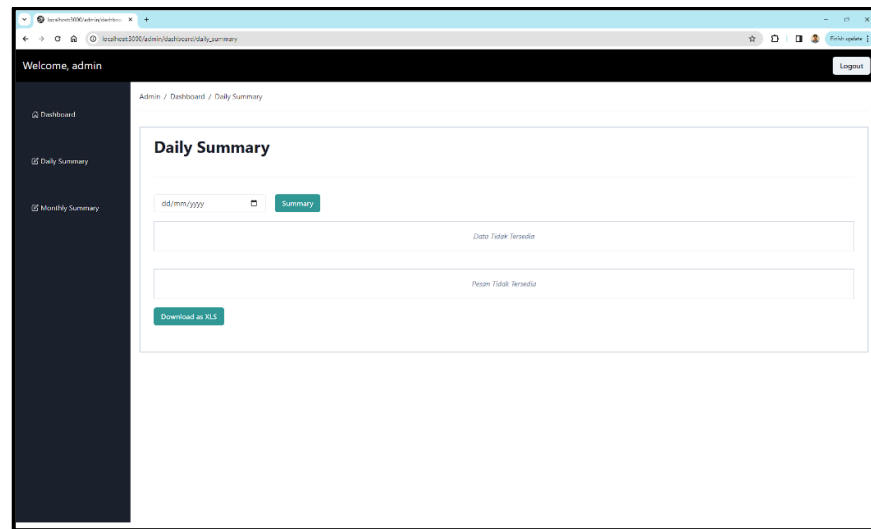
20 export default function AdminDashboardPage() {
21   const { data: session, status } = { data: any; status: string } = useSession();
22   const router = useRouter();
23   console.log(session);
24   console.log(status);
25   useEffect(() => {
26     if (status === "unauthenticated") {
27       router.push("/login");
28       signIn();
29     } else {
30       if (session !== undefined && session?.user?.role !== "admin") {
31         signIn();
32       }
33     }
34   }, [router, session, session?.user?.role, status]);
35   return (
36     <main>
37       <Stack pt={5} spacing={5}>
38         <Breadcrumbs>
39           <BreadcrumbsItem>
40             <Text>Admin</Text>
41           </BreadcrumbsItem>
42           <BreadcrumbsItem>
43             <BreadcrumbsLink href="/admin/dashboard">Dashboard</BreadcrumbsLink>
44           </BreadcrumbsItem>
45           <BreadcrumbsItem isActivePage>
46             <BreadcrumbsLink href="/admin/dashboard/daily_summary">
47               Daily Summary
48             </BreadcrumbsLink>
49           </BreadcrumbsItem>
50         </Breadcrumbs>
51         <div id={1} />
52         <div className="container-summary">
53           <div className="container-heading">
54             <heading paddingBottom="10px">Daily Summary</heading>
55           </div>
56           </div>
57           <div className="container-main">
58             <div className="chart">
59               <div className="button-sum">
60                 <input
61                   placeholder="Select Date and Time"
62                   size="md"
63                   type="date"
64                   className="date-input"
65                 />
66                 <button colorScheme="teal" variant="solid">
67                   Summary
68                 </button>
69               </div>
70               <div className="sum-field-container">
71                 <div className="sum-field">
72                   <Text
73                     alignItems="center">

```

Gambar 4. 36. Isi *code* halaman dan fitur *daily summary*

Fitur dari halaman *daily summary* berfungsi merekap data absensi karyawan yang di filter berdasarkan tanggal atau harian. Fitur *daily summary* menghasilkan sebuah hasil rekap absensi berupa sebuah tabel yang nantinya digunakan pihak manajemen karyawan untuk memudahkan rekapitulasi data absensi.

Berikut pada gambar 4.37. merupakan tampilan dari halaman *daily summary*.



Gambar 4. 37. Tampilan halaman daily summary

Pada halaman utama dari *Daily Summary* ini menampilkan *breadcrumb* yang sama dengan halaman *dashboard*. Selain itu, pada halaman utama juga terdapat fitur *Daily Summary* itu sendiri. Fitur tersebut diawali dengan judul halaman "*Daily Summary*", kemudian terdapat sebuah *input form* tanggal dan tombol "*summary*". Input ini berfungsi sebagai filter tanggal yang akan direkap, dan tombol berfungsi menampilkan hasil rekap di bagian bawah. Saat *form* tanggal sudah terisi dan tombol *summary* ditekan, akan menampilkan absensi karyawan pada tanggal tersebut dalam bentuk tabel. Selain menampilkan absensi karyawan dalam bentuk tabel, juga akan menampilkan sebuah *template message* yang nantinya dilaporkan oleh pihak manajemen bertujuan sebagai laporan untuk pihak manajemen. Semua *output* yang sudah dijelaskan dapat diunduh dalam format *file .xls*.

4.5.5. Halaman *Monthly Summary*

Halaman *monthly summary* merupakan halaman bagian dari *dashboard*. Pada halaman *monthly summary* ini terdapat sebuah *navbar* dan *sidebar* yang sama dengan yang ada pada halaman *dashboard*.

Berikut pada gambar 4.38. merupakan isi *code* halaman dan fitur *monthly summary*.

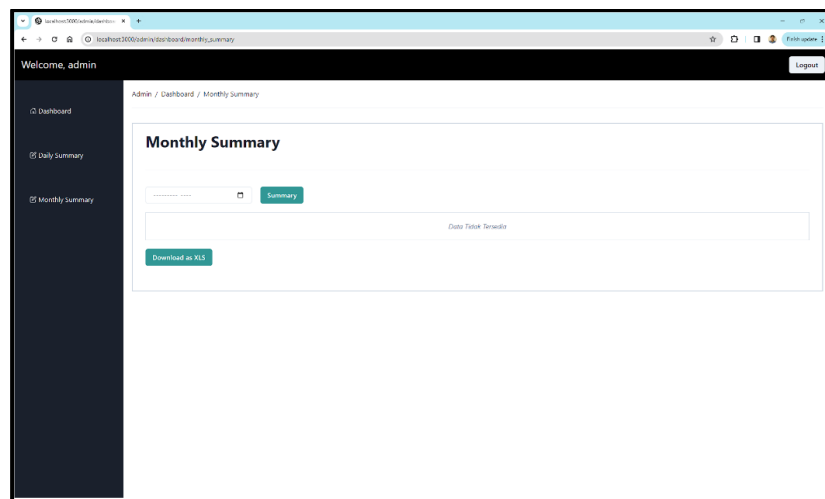
```

20 export default function AdminDashboardPage() {
21   const { data: session, status } = { data: any; status: string } = useSession();
22   const router = useRouter();
23   useEffect(() => {
24     if (status === "unauthenticated") {
25       signIn();
26     } else {
27       if (session !== undefined && session?.user.role !== "admin") {
28         signIn();
29       }
30     }
31   }, [router, session, session?.user.role, status]);
32
33   return (
34     <main>
35       <Stack px(5) spacing(1)>
36         <Breadcrumb>
37           <BreadcrumbItem>
38             <TextAdmin/Text>
39           </BreadcrumbItem>
40           <BreadcrumbItem>
41             <BreadcrumbLink href="/admin/dashboard">Dashboard</BreadcrumbLink>
42           </BreadcrumbItem>
43           <BreadcrumbItem isCurrentPage>
44             <BreadcrumbLink href="/admin/dashboard/daily_summary">
45               Monthly Summary
46             </BreadcrumbLink>
47           </BreadcrumbItem>
48         </Breadcrumb>
49       </div>
50       <div className="container-summary">
51         <div className="container-heading">
52           <Heading paddingBottom="10px">Monthly Summary/Headings</Heading>
53           <div py={1}>
54             <div>
55               <div className="container-main">
56                 <div className="chart">
57                   <div className="button-sum">
58                     <Input
59                       placeholder="Select Date and Time"
60                       size="md"
61                       type="month"
62                       className="date-input"
63                     />
64                     <button colorScheme="teal" variant="solid">
65                       Summary
66                     </button>
67                   </div>
68                   <div className="sum-field-container">
69                     <div className="sum-field">
70                       <Text
71                         align="center"
72                         display="flex">

```

Gambar 4. 38. Isi *code* dari halaman dan fitur *monthly summary*

Berikut pada gambar 4.39. merupakan tampilan dari halaman *monthly summary*.



Gambar 4. 39. tampilan halaman *monthly summary*

Pada halaman utama dari *Monthly Summary* ini menampilkan *breadcrumb* yang sama dengan halaman *dashboard*. Selain itu, pada halaman utama juga terdapat fitur *Monthly Summary* itu sendiri. Fitur tersebut diawali dengan judul halaman "*Monthly Summary*", kemudian terdapat sebuah *input form* bulan dan tombol "*summary*". Input ini berfungsi sebagai filter tanggal yang akan direkap,

dan tombol berfungsi menampilkan hasil rekap di bagian bawah. Saat *form* sudah terisi dan tombol *summary* ditekan, akan menampilkan absensi karyawan pada bulan tersebut dalam bentuk tabel. Semua output yang sudah dijelaskan dapat diunduh dalam format file .xls.

4.7. Pengujian *Website Absensi Karyawan*

Pengujian sistem dilakukan untuk memastikan sistem berjalan dengan semestinya. Pengujian yang dilakukan merupakan uji coba kualitas sistem dengan menggunakan metode *blackbox testing*. *Black box testing* merupakan salah satu metode pengujian sistem yang berfokus kepada sisi fungsionalitas khususnya pada input aplikasi apakah sesuai dengan apa yang diharapkan atau tidak. Tahapan pengujian merupakan salah satu tahap yang harus ada pada siklus pengembangan sistem sebelum *website* di publikasi. Pada laporan ini menggunakan metode pengujian *black box* karena ingin mengetahui fungsionalitas sistem ini.

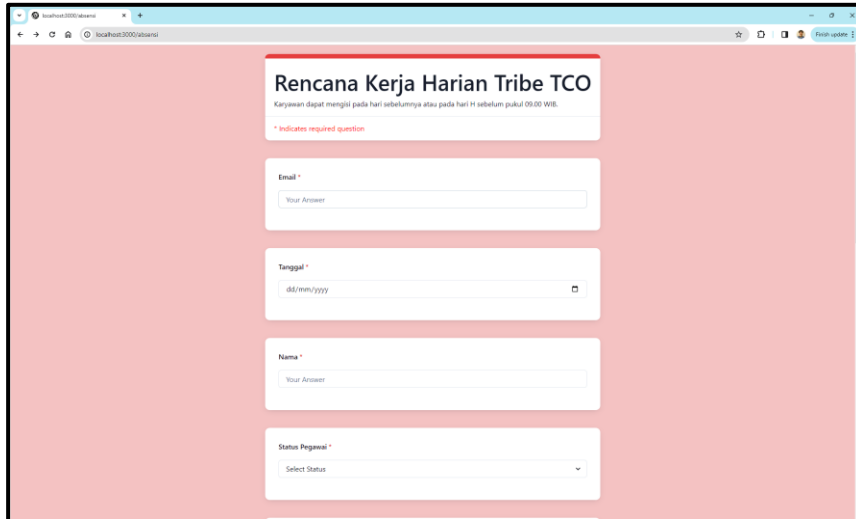
Hasil pengujian sistem ini dideskripsikan dengan menganalisis *output* yang dihasilkan oleh sistem berdasarkan *input* yang diberikan pengguna oleh sistem ini dari setiap menu dan fitur yang terdapat dalam sistem absensi karyawan ini. Berikut menu-menu atau fitur-fitur yang diujikan dalam tahap pengujian.

1. Pengujian sistem absensi pada *website* akan ditunjukkan pada tabel 4.5.

Tabel 4. 5. Pengujian *black box* pada halaman absensi

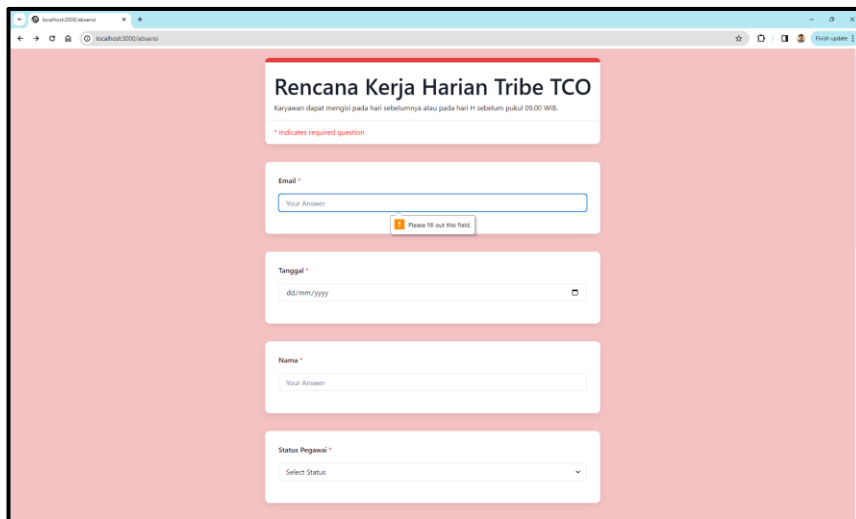
Skenario dan Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
Mengakses dan melihat halaman <i>web</i> dengan mengetikan <i>URL website absensi</i>	Dapat diakses dan tampil semua komponen seperti <i>header</i> , <i>navbar</i> , <i>content</i> , dan <i>footer</i>	Sistem dapat diakses dan menampilkan semua komponen seperti <i>header</i> , <i>navbar</i> , <i>content</i> , dan <i>footer</i>	Berhasil	Merujuk pada gambar 4.40.

Skenario dan Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
<i>Form absensi tidak diisi</i>	Pengguna tetap pada halaman absensi dan muncul <i>alert</i> untuk mengisi <i>form</i> yang masih kosong.	Muncul <i>alert</i> ketika pengguna belum melengkapi semua <i>form</i> setelah pengguna menekan tombol <i>submit</i>	Berhasil	Merujuk pada gambar 4.41.
<i>Form input email pada halaman absensi tidak sesuai</i>	Pengguna tetap pada halaman absensi dan muncul <i>alert</i> untuk mengecek ulang <i>form email</i> yang tidak sesuai	Muncul <i>alert</i> ketika semua <i>form</i> dilengkapi tetapi <i>form email</i> tidak mengandung tanda “@” pada <i>inputnya</i>	Berhasil	Merujuk pada gambar 4.42.
Mengisi semua <i>form</i> dan melakukan <i>submit</i> , dan tercatat pada <i>database</i>	Pengguna berhasil submit lalu dialihkan ke halaman sukses dan pada <i>database</i> tercatat semua <i>form</i> yang diisi	Pengguna dapat melakukan <i>submit</i> lalu dialihkan ke halaman sukses dan <i>submit</i> berhasil terekam di <i>database</i>	Berhasil	Merujuk pada gambar 4.43. dan gambar 4.44.



The screenshot shows a web browser window with the URL 'localhost:3000/absensi'. The page has a light red background. At the top, there is a white box with the title 'Rencana Kerja Harian Tribe TCO' and a subtitle 'Karyawan dapat mengisi pada hari sebelumnya atau pada hari H sebelum pukul 09.00 WIB.' Below this, there is a red asterisk icon and the text '* Indicates required question'. The form consists of four white input boxes: 'Email *' with a placeholder 'Your Answer', 'Tanggal *' with a placeholder 'dd/mm/yyyy' and a calendar icon, 'Nama *' with a placeholder 'Your Answer', and 'Status Pegawai *' with a dropdown menu showing 'Select Status'.

Gambar 4. 40. Pengujian halaman absensi 1



The screenshot shows the same web browser window as before, but now the 'Email *' field has a validation error. The placeholder text 'Your Answer' is still present, but there is a red border around the field and a small orange icon with the text 'Please fill out this field.' below it. The other fields ('Tanggal *', 'Nama *', and 'Status Pegawai *') remain unchanged.

Gambar 4. 41. Pengujian halaman absensi 2

Rencana Kerja Harian Tribe TCO
Karyawan dapat mengisi pada hari sebelumnya atau pada hari H sebelum pukul 09:00 WIB.

* Indicates required question

Email *

faizanyuliansyah.com

Please include an '@' in the email address. 'faizanyuliansyah.com' is missing an '@'.

Tanggal *

11/02/2024

Nama *

Faizan

Status Pegawai *

Digital Talent / Pro Hire

Gambar 4. 42. Pengujian halaman absensi 3

Rencana Kerja Harian Tribe TCO

Terdima kasih sudah mengisi absen hari ini, tetap semangat! 🌟

[Submit another response](#)

Gambar 4. 43. Pengujian halaman absensi 4

Id	createdAt	email	date	name	employeeStatus	squad	condition	conditionDesc	workFrom
1	2024-02-14T08:44:49.740Z	faizanyuliansyah@gmail.com	2024-02-14T08:42:12.863Z	Faizan Dwi Nuliansyah	Karyawan Tetap	Project & Platform	Sakit	sakit	work from Office

Gambar 4. 44. Pengujian halaman absensi 5

2. Pengujian sistem *login* pada *website* akan ditunjukkan pada tabel 4.6.

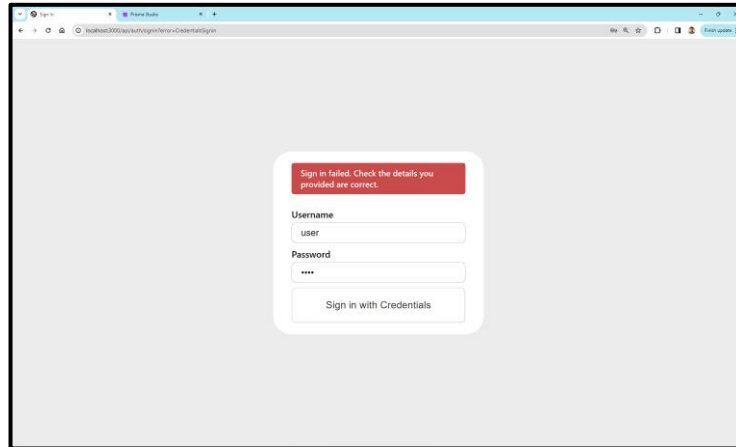
Tabel 4. 6. Pengujian *black box* pada *login*

Skenario dan Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
Mengakses dan melihat halaman <i>web</i> dengan mengetikan <i>URL website</i> absensi	Dapat diakses dan tampil semua komponen seperti <i>header, navbar, content, dan footer</i>	Sistem dapat diakses dan menampilkan semua komponen seperti <i>header, navbar, content, dan footer</i>	Berhasil	Merujuk pada gambar 4.45.
<i>Input username</i> dan <i>password</i> tidak diisi	Pengguna tetap pada halaman login dan muncul <i>alert</i> untuk mengisi <i>form</i> kosong	Pengguna tetap pada halaman login dan muncul <i>alert</i> untuk mengisi <i>form</i> kosong	Berhasil	Merujuk pada gambar 4.46.
<i>Input username</i> dan <i>password</i> diisi dengan data yang salah atau tidak sesuai	Pengguna tetap pada halaman login dan muncul <i>alert</i> bahwa <i>username</i> dan <i>password</i> yang <i>diinputkan</i> salah.	Pengguna tetap pada halaman login dan muncul <i>alert</i> bahwa <i>username</i> dan <i>password</i> yang <i>diinputkan</i> salah	Berhasil	Merujuk pada gambar 4.47.

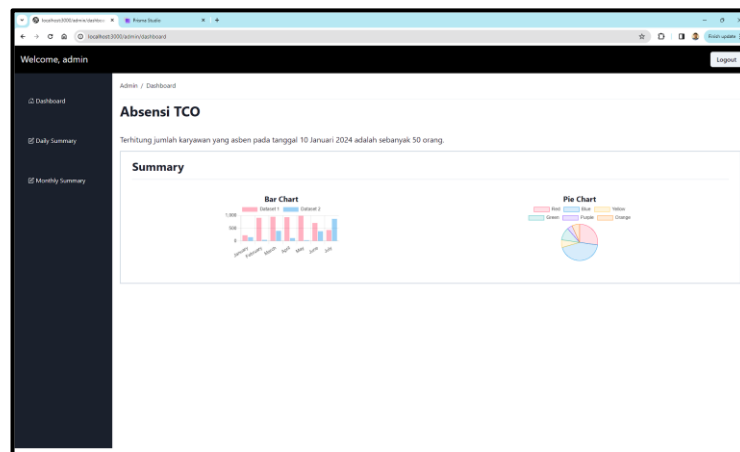
Skenario dan <i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
Input <i>username</i> dan <i>password</i> diisi dengan data yang benar	Pengguna dialihkan ke halaman <i>dashboard</i>	Pengguna dialihkan ke halaman <i>dashboard</i>	Berhasil	Merujuk pada gambar 4.48.

Gambar 4. 45. Pengujian halaman *login* 1

Gambar 4. 46. Pengujian halaman *login* 2



Gambar 4. 47. Pengujian halaman *login* 3



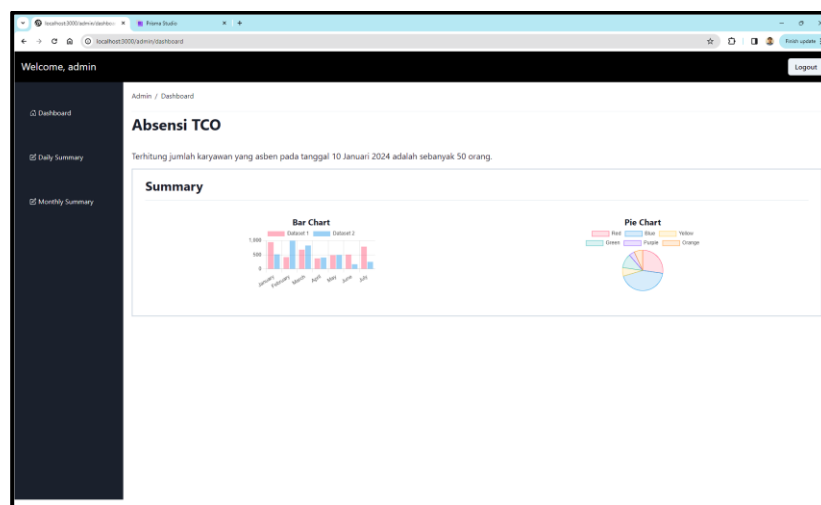
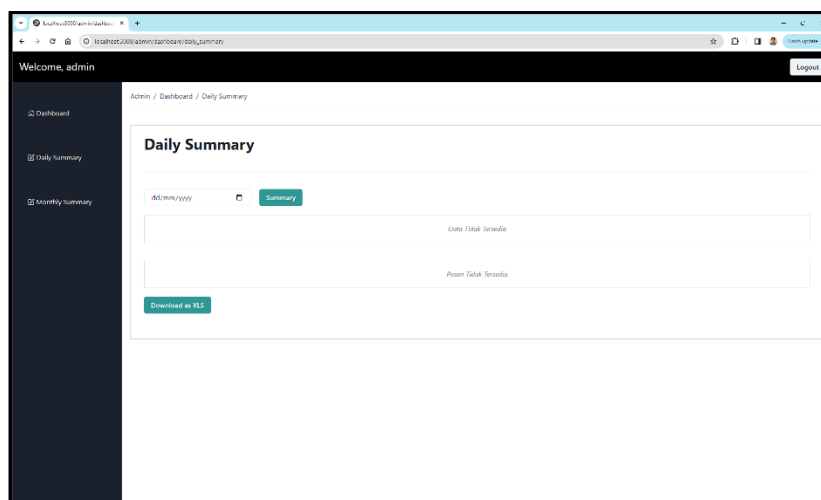
Gambar 4. 48. Pengujian halaman *login* 4

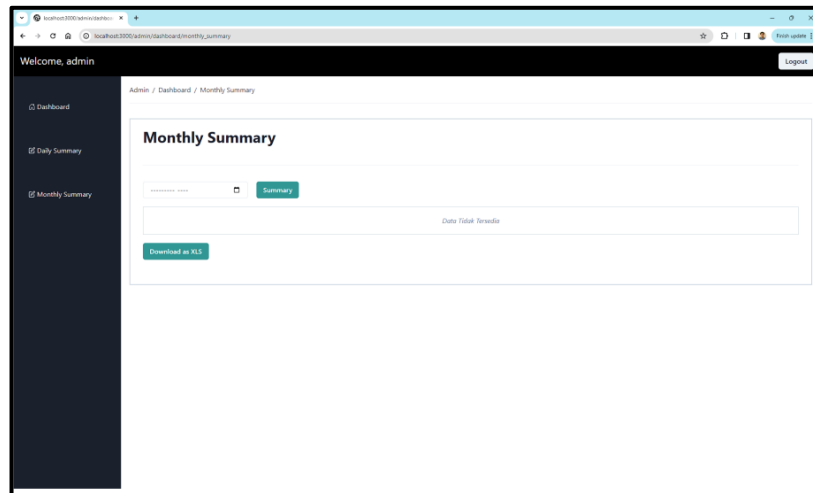
3. Pengujian sistem pada halaman *dashboard* akan ditunjukkan pada tabel 4.7.

Tabel 4. 7. Pengujian *black box* sistem pada halaman *dashboard*

Skenario dan Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
Mengakses halaman <i>dashboard</i> dalam keadaan belum <i>login</i>	Pengguna dialihkan ke halaman <i>login</i>	Pengguna dialihkan ke halaman <i>login</i>	Berhasil	
Mengakses dan melihat halaman <i>web</i> dengan mengetikkan <i>URL</i> halaman <i>dashboard</i>	Dapat diakses dan tampil semua komponen seperti <i>header</i> , <i>navbar</i> , <i>content</i> , dan <i>footer</i>	Sistem dapat diakses dan menampilkan semua komponen seperti <i>header</i> , <i>navbar</i> , <i>content</i> , dan <i>footer</i>	Berhasil	Merujuk pada gambar 4.49.
Menekan tombol <i>logout</i> pada <i>navbar</i> untuk mengeluarkan akun <i>admin</i>	Pengguna dialihkan ke halaman <i>login</i> dan akun <i>admin</i> berhasil dikeluarkan	Pengguna dialihkan ke halaman <i>login</i> dan akun <i>admin</i> berhasil dikeluarkan	Berhasil	
Menekan <i>link daily summary</i> pada <i>sidebar</i>	Pengguna dialihkan ke halaman <i>daily summary</i>	Pengguna dialihkan ke halaman <i>daily summary</i>	Berhasil	Merujuk pada gambar 4.50.

Skenario dan Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
Menekan <i>link monthly summary</i> pada <i>sidebar</i>	Pengguna dialihkan ke halaman <i>monthly summary</i>	Pengguna dialihkan ke halaman <i>monthly summary</i>	Berhasil	Merujuk pada gambar 4.51.

Gambar 4. 49. Pengujian halaman *dashboard 1*Gambar 4. 50. Pengujian halaman *dashboard 2*

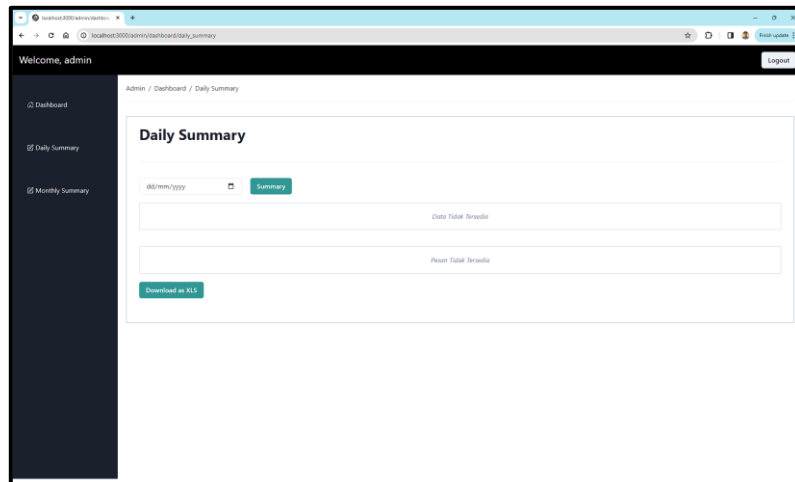
Gambar 4. 51. Pengujian halaman *dashboard* 3

4. Pengujian fitur *daily summary* pada *website* akan ditunjukkan pada tabel 4.8.

Tabel 4. 8. Pengujian *black box* pada fitur *daily summary*

Skenario dan Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
Mengakses halaman <i>daily summary</i> dalam keadaan belum <i>login</i>	Pengguna dialihkan ke halaman <i>login</i>	Pengguna dialihkan ke halaman <i>login</i>	Berhasil	
Mengakses dan melihat halaman <i>web</i> dengan mengetikkan <i>URL</i> halaman	Dapat diakses dan tampil semua komponen seperti <i>header</i> , <i>navbar</i> , <i>content</i> , dan <i>footer</i>	Sistem dapat diakses dan menampilkan semua komponen seperti <i>header</i> , <i>navbar</i> , <i>content</i> ,	Berhasil	Merujuk pada gambar 4.52.

Skenario dan Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
<i>daily summary</i>		dan <i>footer</i>		
Mengisi <i>form</i> tanggal dan menekan tombol <i>summary</i>	Pengguna dapat mengisi tanggal pada <i>form</i> dan menekan tombol <i>summary</i>	Pengguna dapat mengisi tanggal pada <i>form</i> dan menekan tombol <i>summary</i>	Berhasil	
Menampilkan hasil <i>summary</i> berupa data tabel dan <i>template</i> pesan	Sistem dapat menampilkan hasil <i>summary</i> berupa data tabel dan data <i>template</i> pesan	Sistem dapat menampilkan hasil <i>summary</i> berupa data tabel dan data <i>template</i> pesan	Berhasil	
Menekan tombol <i>Download as XLS</i> dan dapat mengunduh <i>file .xls</i>	Sistem dapat mengeksekusi perintah pada tombol <i>Download as XLS</i> dan pengguna dapat mengunduh <i>file .xls</i>	Sistem dapat mengeksekusi perintah pada tombol <i>Download as XLS</i> dan pengguna dapat mengunduh <i>file .xls</i>	Berhasil	



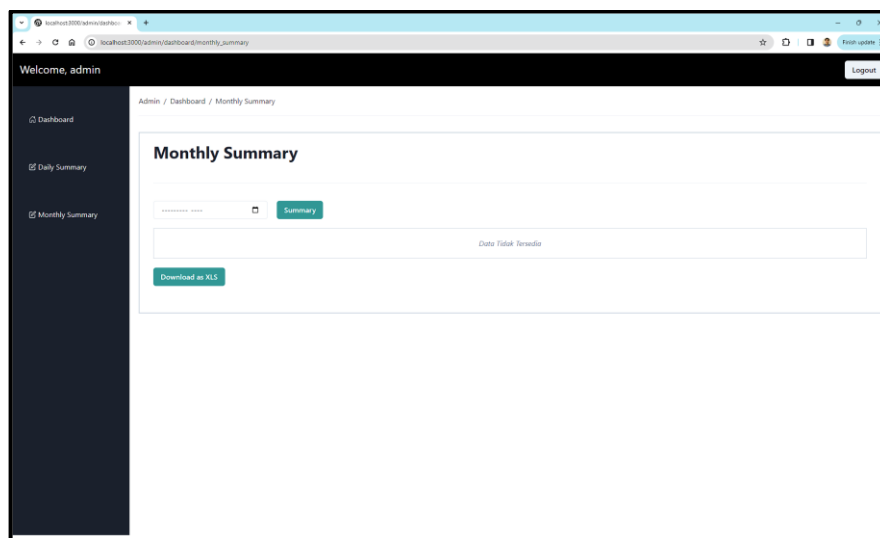
Gambar 4. 52. Pengujian halaman *daily summary* 1

5. Pengujian *fitur monthly summary* pada *website* ditunjukkan pada tabel 4.8.

Tabel 4. 9. Pengujian *black box* pada *monthly summary*

Skenario dan Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
Mengakses halaman <i>monthly summary</i> dalam keadaan belum <i>login</i>	Pengguna dialihkan ke halaman <i>login</i>	Pengguna dialihkan ke halaman <i>login</i>	Berhasil	
Mengakses dan melihat halaman <i>web</i> dengan mengetikkan <i>URL</i> halaman <i>monthly summary</i>	Dapat diakses dan tampil semua komponen seperti <i>header</i> , <i>navbar</i> , <i>content</i> , dan <i>footer</i>	Sistem dapat diakses dan menampilkan semua komponen seperti <i>header</i> , <i>navbar</i> , <i>content</i> , dan <i>footer</i>	Berhasil	Merujuk pada gambar 4.53.
Mengisi <i>form</i> tanggal dan menekan tombol <i>summary</i>	Pengguna dapat mengisi tanggal pada <i>form</i> dan menekan tombol <i>summary</i>	Pengguna dapat mengisi tanggal pada <i>form</i> dan menekan tombol <i>summary</i>	Berhasil	
Menampilkan hasil <i>summary</i> berupa data tabel	Sistem dapat menampilkan hasil <i>summary</i> berupa data tabel	Sistem dapat menampilkan hasil <i>summary</i> berupa data tabel	Berhasil	

Skenario dan <i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan	Keterangan
Menekan tombol <i>Download as XLS</i> dan dapat mengunduh <i>file .xls</i>	Sistem dapat mengeksekusi perintah pada tombol <i>Download as XLS</i> dan pengguna dapat mengunduh <i>file .xls</i>	Sistem dapat mengeksekusi perintah pada tombol <i>Download as XLS</i> dan pengguna dapat mengunduh <i>file .xls</i>	Berhasil	

Gambar 4. 53. Pengujian halaman *monthly summary* 1

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan uraian dari bab-bab sebelumnya yang membahas mengenai *website* absensi ini, serta setelah penulis menguji sistem tersebut dapat disimpulkan bahwa:

1. Sistem absensi berbasis *website* menggunakan *framework javascript*, *next.js* juga menggunakan *prisma ORM* dan *PostgreSQL* sebagai *database* telah berhasil dibuat dan berjalan dengan baik dan stabil.
2. Sistem absensi berbasis *website* yang terintegrasi dengan *database* sehingga dapat dikelola dimana saja dan kapan saja, selain itu juga dapat mempercepat proses rekap data absensi.
3. Dengan adanya sistem absensi karyawan ini sudah dapat membantu pihak manajemen untuk merekap absensi karyawan dengan cara yang lebih efisien dan telah berhasil mengurangi adanya kesalahan dalam rekapitulasi data.

5.2. Saran

Pengimplementasian sistem absensi berbasis *website* ini masih dapat dikembangkan lagi oleh pihak manapun yang ingin melakukan pengembangan lebih lanjut mengenai sistem ini, sehingga kedepannya proyek ini menjadi lebih baik karena adanya tambahan beberapa fitur dengan mempertimbangkan beberapa hal yang disarankan untuk diaplikasikan, diantaranya yaitu:

1. Akun *admin* pada sistem absensi berbasis *website* ini dapat diintegrasikan pada *database* untuk nantinya di kembangkan data akun *admin* yang lebih dinamis, yaitu *admin* bisa merubah *username* ataupun *password* kedepannya jika diperlukan perubahan.
2. Selain diintegrasikan pada *database* akun *admin* pada sistem absensi berbasis *website* ini dapat diintegrasikan pada *gmail*, ataupun nomor

telepon (*whatsapp*), dan lain-lain untuk kebutuhan, seperti *2-Step Verification* pada proses *login* ataupun pada proses manajemen akun *admin*.

3. Rekapitulasi data absensi pada sistem absensi berbasis *website* ini dapat dikembangkan dengan menambah sebuah diagram data dengan tujuan agar pihak manajemen atau *admin* dapat dengan mudah membaca hasil rekapitulasi.

DAFTAR PUSTAKA

- Adani, M. R. (2020, Agustus 7). *Pengenalan Apa Itu Framework dan Jenisnya untuk Web Development*. Retrieved from Sekawan Media: <https://www.sekawanmedia.co.id/blog/pengertian-framework/>
- Adani, M. R. (2021, June 22). *Data Flow Diagram (DFD): Pengertian, Jenis, Fungsi & Contoh*. Retrieved from Artikel, Programming: <https://www.sekawanmedia.co.id/blog/dfd-adalah/>
- Adi Nugroho, Y. R. (2009). PERANCANGAN APLIKASI RENCANA ANGGARAN BIAYA (RAB) (STUDI KASUS PADA DINAS PEKERJAAN UMUM KOTA SALATIGA). *JURNAL INFORMATIKA*, 10(1), 10-18.
- Ariffud. (2023, May 24). *Use Case Diagram: Pengertian, Contoh, Simbol, Cara Membuatnya*. Retrieved from Niagahoster Blog: <https://www.niagahoster.co.id/blog/use-case-diagram-adalah/>
- Bratha, W. G. (2021). LITERATURE REVIEW KOMPONENSISTEM INFORMASI MANAJEMEN: SOFTWARE, DATABASE DAN BRAINWARE. *Jurnal Ekonomi Manajemen Sistem Informasi*, 3(3), 344-360.
- Cegielski. (2015). *Introduction to Information System*.
- Dianty, M. A. (2016). *Flowmap*. Retrieved from Sistem Informasi Manajemen: <https://sisteminformasimanajemen15089.wordpress.com/2016/12/14/flowmap/>
- Famy, S. (2022). *RANCANG BANGUN SISTEM INFORMASI MANAJEMEN BERBASIS WEB MENGGUNAKAN NEXTJS DI CV. SANJAYA ABADI BARU*.
- Febio, R. S. (2011). MEMBANGUN APLIKASI E-LIBRARY MENGGUNAKAN HTML, PHP SCRIPT, DAN MYSQL DATABASE. *Jurnal PROCESSOR*, 6(2), 38-54.

- Fernando, D. (2018). Visualisasi Data Menggunakan Google Data Studio. *Seminar Nasional Rekayasa Teknologi Informasi* (pp. 71-77). Serang: Fakultas Teknologi Informasi, Universitas Serang Raya.
- git-scm.com. (n.d.). *Tentang Version Control*. Retrieved from GIT SCM: <https://git-scm.com/book/id/v2/Memulai-Tentang-Version-Control>
- HP. (2018). https://repository.bsi.ac.id/index.php/unduh/item/2473/File_13-Bab-II-Landasan-Teori.pdf, 6-22. Retrieved from https://repository.bsi.ac.id/index.php/unduh/item/2473/File_13-Bab-II-Landasan-Teori.pdf
- Humisar Hasugian, A. N. (2012). RANCANG BANGUN SISTEM INFORMASI INDUSTRI KREATIF BIDANG PENYEWAAN SARANAOLAHRAGA. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2012* (pp. 606-612). Semarang: Fakultas Teknologi Informasi.
- Jagoan Hosting, T. (2023, Maret). *Apa itu API? Jenis, Fungsi & Contohnya*. Retrieved from Jagoan Hosting: <https://www.jagoanhosting.com/blog/apa-itu-web-api/>
- Kresna Prasmadewa, R. T. (2016). Perancangan Aplikasi Usaha Kecil Mikro dan Menengah Berbasis Mobile Android (Studi Kasus: Sentra UMKM Tingkir Lor-Salatiga). *Jurnal Teknik Informatika dan Sistem Informasi*, 2(3), 351-358.
- Leitch. (2011). *Sistem Informasi Akuntansi*. Ghalia Indonesia.
- Musyaffa et al, M. I. (2024, January 1). RANCANG BANGUN PURECOMPUTE PLATFORM E-COMMERCE UNTUK BELANJA LAPTOP BERBASIS WEBSITE. *Jurnal Sistem Informasi dan Teknologi Komputasi*, 1(1), 21-29.
- Nugroho, D. S. (2023). Inovasi Pembaruan Desain Website Penyederhana Link Unesa Menggunakan Typescript Dan Node.js. *Jurnal Ilmu Teknik*, 1(2), 35-40.
- O'brien. (2011). *Management Information System* (Vol. 10). New York: McGraw-Hill/Irwin.

- Pahlevi, O. (2018). SISTEM INFORMASI INVENTORIBARANGMENGGUNAKAN METODE OBJECT ORIENTEDDIPT. LIVAZA TEKNOLOGI INDONESIA JAKARTA. *Jurnal PROSISKO*, 5(1), 27-35.
- Revoupedia. (2023). *NPM (Node Package Manager)*. Retrieved from Revoupedia: <https://revou.co/kosakata/npm>
- Revoupedia. (n.d.). *Apa itu Library dalam Pemrograman*. Retrieved from Revoupedia: <https://revou.co/kosakata/library>
- Sano, A. V. (n.d.). *Beberapa Definisi Tentang Data, Informasi, dan Sistem Informasi Menurut Beberapa Ahli*. Retrieved from Computer Science: <https://binus.ac.id/malang/2020/12/beberapa-definisi-tentang-data-informasi-dan-sistem-informasi-menurut-beberapa-ahli/>
- Santoso, R. (2014). *Tuntunan Praktis Membangun Sistem Informasi Akutansi Dengan Visual Basic dan Microsoft SQL Server*. CV. Andi Offset.
- Sauerwein, J. C. (1964-1984). *Standard Reference Data Publications*. Gaithersburg: NBS Specials Publication 708.
- Setiawan, I. (2022). *Komparasi KinerjaIntegrated Development Environment (IDE) Dalam Mengeksekusi Perintah Python*. *Jurnal Sains Teknologi Dan Sistem Informasi*, 2(1), 52-59.
- Setiawan, R. (2021, August 25). *Memahami ERD, Model Data, dan Komponennya*. Retrieved from Dicoding: <https://www.dicoding.com/blog/memahami-erd/>
- sis.binus.ac.id. (n.d.). *Apa itu Git & Github*. Retrieved from BINUS School of Information Systems: <https://sis.binus.ac.id/2023/06/27/apa-itu-git-github/>
- Sosanto et al, D. M. (2021). MEMBANGUN SISTEM INFORMASI PRESENSI PENGUNJUNG PERPUSTAKAAN UNIVERSITAS MAHAKARYA ASIA DENGAN MEMANFAATKAN QR CODE MENGGUNAKAN CODEIGNITER 3. *Jurnal Teknik Informatika Mahakarya*, 4(2), 14-23.

Sunyoto, A. (2010). AJAX (Asynchronous JavaScript and XML). *Jurnal DASI*, 11(3), 1-11.

Waryanto. (2021). *Pengertian Website, Fungsi, dan Jenis-Jenisnya*. Retrieved from Peta Network: <https://www.peta-network.com/pengertian-website/>

Yudianto, J. N. (2003). *Jaringan Komputer Dan Pengertiannya*.