

**PENGEMBANGAN *USER INTERFACE* APLIKASI *S2P-MOBILE*
HSE BERBASIS IONIC FRAMEWORK PADA FITUR *LOGIN*,
PANIC BUTTON, DAN NOTIFIKASI**



Disusun Oleh:

N a m a : Ikhwan Alfi Mushaffa Asadilah

NIM : 21523031

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN *USER INTERFACE* APLIKASI *S2P-MOBILE* HSE
BERBASIS IONIC FRAMEWORK PADA FITUR *LOGIN*,
PANIC BUTTON, DAN NOTIFIKASI**

TUGAS AKHIR JALUR MAGANG



Nama : Ikhwan Alfi Mushaffa Asadilah
NIM : 21523031

Yogyakarta, 1 November 2025

Pembimbing,


(Dr. Ahmad Luthfi, S.Kom., I.Ti.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN *USER INTERFACE* APLIKASI *S2P-MOBILE HSE*
BERBASIS *IONIC FRAMEWORK* PADA FITUR *LOGIN*,
PANIC BUTTON, DAN NOTIFIKASI**

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta,

Tim Penguji

Ketua Penguji

Dr. Ahmad Luthfi, S.Kom., M.Kom.



Anggota 1

Sheila Nurul Huda, S.Kom., M.Cs.



Anggota 2

Taufiq Hidayat, S.T., M.C.S.



Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)



HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Ikhwan Alfi Mushaffa Asadilah

NIM : 21523031

Tugas akhir dengan judul:

**PENGEMBANGAN *USER INTERFACE* APLIKASI S2P-MOBILE HSE
BERBASIS IONIC FRAMEWORK PADA FITUR *LOGIN*, *PANIC BUTTON*,
DAN NOTIFIKASI**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 25 Desember 2025



(Ikhwan Alfi Mushaffa Asadilah)

HALAMAN PERSEMBAHAN

Segala puji bagi Allah Swt., Tuhan Yang Maha Pengasih lagi Maha Penyayang atas berbagai karunia dan rahmat-Nya. Terima kasih dipanjatkan atas ridho kedua orang tua dan dukungan adik-adik tercinta, karena ridho Allah tidak terlepas dari ridho mereka. Yang tak kalah penting, terima kasih kepada dosen pembimbing dan pihak PT Nocola IoT Solution yang telah memberikan bimbingan dan kesempatan berharga selama pelaksanaan magang hingga terselesaikannya laporan tugas akhir ini.

HALAMAN MOTO

“Allah memang tidak menjanjikan kehidupanmu akan selalu mudah tetapi dua kali Allah berjanji bahwa: fa inna ma’al-usri yusra, inna ma’al-usri yusra”

(QS. Al-Insyirah 94:5-6)

"Semua jatuh bangunmu hal yang biasa, angan dan pertanyaan waktu yang menjawabnya, berikan tenggat waktu bersedih lah secukupnya, rayakan perasaanmu sebagai manusia."

(Baskara Putra-Hindia)

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah Subhanahu Wa Ta'ala atas segala rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan laporan tugas akhir dengan judul “Pengembangan *User Interface* Aplikasi S2P-Mobile HSE Berbasis Ionic Framework pada Fitur *Login*, *Panic Button*, dan Notifikasi”. Laporan tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Penyusunan laporan ini tidak terlepas dari bantuan, dukungan, serta doa dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan kerendahan hati, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah Swt. yang telah memberikan kesehatan, kekuatan, dan kelancaran untuk menyelesaikan tugas akhir ini mulai dari pelaksanaan magang sampai dengan selesainya penulisan laporan akhir ini.
2. Kedua orang tua penulis, Bapak Arief Maryono Khaerudin dan Ibu Titik Prihati, yang senantiasa memberikan dukungan baik itu doa, moral, maupun materi yang tidak ternilai agar penulis dapat melakukan perkuliahan dengan lancar sampai akhir.
3. Saudari kandung penulis, Muhammad Samsul Lutfi, dan Putri Zulfa Alfianan Azzahro yang telah memberikan dukungan kepada penulis agar dapat menjalankan perkuliahan dengan baik.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Jurusan Informatika Universitas Islam Indonesia.
5. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
6. Ibu Erika Ramadhani, S.T., M.Eng., selaku Dosen Pembimbing Akademik yang telah memberikan bimbingan, arahan, dan motivasi selama masa perkuliahan hingga penyusunan tugas akhir.
7. Bapak Dr. Ahmad Luthfi, S.Kom., M.Kom., selaku Dosen Pembimbing Tugas Akhir yang dengan sabar telah membimbing, memberikan masukan, serta membagikan ilmu yang sangat berharga kepada penulis selama proses penyusunan laporan ini.
8. Kepada seluruh petinggi dan karyawan PT Nocola Iot Solution yang telah memberikan kesempatan dan ilmu baru selama melakukan kegiatan magang.

9. Cinta Annastasya Sari, yang senantiasa memberikan doa, dukungan, dan semangat kepada penulis selama proses penyusunan tugas akhir ini.
10. Rekan-rekan seperjuangan, khususnya Cygnus, Dhika, dan Dava, fatih, yang selalu memberikan semangat, inspirasi, serta kerja sama yang baik selama proses penyusunan tugas akhir ini.
11. Teman-teman “Wong Angel”, yang telah memberikan kebersamaan, dukungan, serta semangat selama proses perkuliahan hingga penyusunan tugas akhir ini.

Akhir kata, penulis berharap semoga Allah Subhanahu Wa Ta’ala membalas segala kebaikan semua pihak yang telah membantu dengan pahala dan keberkahan yang berlipat ganda. Semoga laporan ini dapat memberikan manfaat serta menjadi referensi yang berguna bagi pengembangan ilmu pengetahuan dan teknologi di masa mendatang.

Yogyakarta, 25 Desember 2025



(Ikhwan Alfi Mushaffa Asadilah)

SARI

Penelitian ini membahas implementasi sisi *frontend* aplikasi S2P-Mobile, yang mencakup pengelolaan antarmuka pengguna, logika interaksi, integrasi API, serta pemanfaatan fitur perangkat menggunakan Ionic Framework. Pada aplikasi S2P-Mobile, sebuah aplikasi yang digunakan untuk mendukung digitalisasi sistem *Health, Safety, and Environment* (HSE) di PT Sumber Segara Primadaya (S2P). Pengembangan dilakukan sebagai bagian dari kegiatan magang di PT Nocola IoT Solution, dengan fokus pada tiga fitur utama, yaitu halaman *login* dengan CAPTCHA, fitur *Panic Button* untuk pelaporan darurat, serta sistem notifikasi *real-time*. Seluruh proses pengembangan menggunakan Ionic Framework berbasis Angular dan Capacitor untuk mendukung kemampuan lintas platform pada perangkat Android dan iOS. Metode pengembangan yang digunakan adalah *Agile Development* yang diimplementasikan dalam tiga *sprint* utama. Setiap *sprint* mencakup perencanaan, implementasi antarmuka, integrasi logika *frontend*, dan pengujian fungsional. Desain UI/UX diperoleh dari tim desain melalui Figma dan diadaptasi secara modular agar mudah diuji serta dapat berfungsi stabil di berbagai ukuran layar. Pengujian fungsional dilakukan menggunakan simulator iOS melalui Xcode dan perangkat fisik iPhone XR untuk memastikan konsistensi tampilan, akurasi interaksi antarmuka, serta respons aplikasi terhadap fitur perangkat seperti GPS dan penyimpanan lokal. Hasil pengujian menunjukkan bahwa seluruh fitur berjalan sesuai desain dan spesifikasi. Halaman *login* dengan CAPTCHA berhasil memvalidasi input pengguna dan menampilkan elemen antarmuka secara responsif. Fitur *Panic Button* mampu menampilkan form secara stabil serta memperoleh koordinat lokasi perangkat dengan akurat melalui integrasi Capacitor Geolocation. Sistem notifikasi *real-time* dapat menampilkan daftar pesan, memperbarui status notifikasi, dan menyimpan data secara lokal sehingga tetap dapat diakses meski tanpa koneksi internet. Penelitian ini menyimpulkan bahwa penggunaan Ionic Framework dan Angular terbukti efektif dalam pengembangan aplikasi lintas platform yang membutuhkan integrasi fitur perangkat keras. Selain itu, penerapan metode Agile membantu proses pengembangan menjadi lebih terstruktur dan adaptif. Kegiatan magang ini juga memberikan manfaat akademik dan profesional bagi penulis, terutama dalam memahami penerapan teori perkuliahan dalam lingkungan industri serta meningkatkan keterampilan teknis, komunikasi, dan pemecahan masalah dalam pengembangan aplikasi *mobile* modern.

Kata kunci: Ionic Framework, Angular, *User Interface*, *Panic Button*, CAPTCHA, Notifikasi *Real-Time*, *Agile Development*, HSE, S2P-Mobile.

GLOSARIUM

Ionic Framework	Kerangka kerja sumber terbuka untuk membangun aplikasi mobile lintas platform (<i>cross-platform</i>) menggunakan HTML, CSS, dan JavaScript. Ionic memungkinkan pengembang membuat aplikasi Android dan iOS dari satu basis kode dengan antarmuka yang konsisten.
Angular	Framework berbasis TypeScript yang digunakan untuk mengatur logika, struktur, dan interaksi antarmuka pengguna (UI) pada aplikasi S2P-Mobile. <i>Angular</i> mendukung konsep data binding dan component-based architecture.
Capacitor	Library yang terintegrasi dengan Ionic Framework untuk menghubungkan aplikasi berbasis web dengan fitur perangkat asli (native) seperti GPS, kamera, NFC, dan penyimpanan lokal.
CAPTCHA	Mekanisme keamanan pada halaman <i>login</i> untuk memastikan bahwa pengguna yang mengakses aplikasi adalah manusia, bukan bot atau program otomatis.
HSE	Pendekatan manajemen yang berfokus pada perlindungan kesehatan, keselamatan, dan lingkungan di tempat kerja. Dalam konteks penelitian ini, HSE menjadi dasar pengembangan aplikasi S2P-Mobile.
<i>Panic Button</i>	Fitur pelaporan insiden darurat pada aplikasi S2P-Mobile yang memungkinkan pengguna mengirim laporan dengan cepat dan akurat, dilengkapi dengan data lokasi otomatis melalui GPS.
UI/UX	UI adalah tampilan visual yang digunakan pengguna untuk berinteraksi dengan aplikasi, sedangkan UX mencakup pengalaman dan kenyamanan pengguna selama menggunakan aplikasi.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Ruang Lingkup Magang	3
1.3 Tujuan.....	4
1.4 Manfaat.....	4
1.5 Sistematika Penulisan.....	4
BAB II DASAR TEORI	6
2.1 <i>Health Safety and Environment (HSE)</i>	6
2.2 <i>Antarmuka Pengguna (User Interface) dan Pengalaman Pengguna (User Experience)</i>	7
2.3 <i>Agile Development</i>	8
2.4 Pengembangan Frontend Aplikasi Mobile	9
2.5 <i>Ionic Framework</i>	10
2.6 Fitur Login dengan CAPTCHA	11
2.7 Fitur Panic Button	12
2.8 Fitur Notifikasi	13
2.9 Tinjauan Pustaka	14
BAB III PELAKSANAAN MAGANG.....	23
3.1 Gambaran Umum Sistem S2P-Mobile	23
3.1.1 Ruang Lingkup Tugas Akhir dalam Sistem.....	24
3.2 Proses Bisnis Aplikasi S2P-Mobile.....	26

3.2.1	Proses Otentikasi Pengguna	26
3.2.2	Proses Pelaporan Panic Button	27
3.2.3	Proses Distribusi Notifikasi <i>Real-Time</i>	29
3.2.4	Ringkasan Interaksi Antar Komponen	30
3.3	Manajemen Proyek	31
3.3.1	Perkenalan Lingkungan Perusahaan (<i>welcoming</i>)	32
3.3.2	Pengarahan dari Supervisor dan <i>Instalasi</i> Ionic Framework	35
3.3.3	Training Ionic Framework	36
3.3.4	Pengembangan Aplikasi S2P-Mobile	41
3.4	Hasil	52
3.4.1	Halaman <i>Login</i>	53
3.4.2	Halaman <i>Panic Button</i>	55
3.4.3	Halaman Notifikasi	57
3.5	Pembahasan	59
3.5.1	Keterkaitan Hasil Pengujian dengan Tujuan Magang	60
3.5.2	Temuan dan Tantangan Selama Magang	61
3.5.3	Kesimpulan Pembahasan	61
BAB IV REFLEKSI PELAKSANAAN MAGANG		62
4.1	Refleksi Pelaksanaan Magang	62
4.2	Pembelajaran Magang	63
4.2.1	Manfaat Magang	64
4.2.2	Kendala, Hambatan, dan Tantangan	64
BAB V KESIMPULAN DAN SARAN		66
5.1	Kesimpulan	66
5.2	Saran	67
DAFTAR PUSTAKA		68
LAMPIRAN		71

DAFTAR TABEL

Tabel 2. 1 Tinjauan Pustaka.....	17
Tabel 3. 1 Aktivitas Magang.....	31
Tabel 3. 2 Daftar Penugasan Pengembangan Aplikasi S2P-Mobile.....	44
Tabel 3. 3 Instrumen Pengujian Fungsional	49
Tabel 3. 4 Hasil Pengujian Fungsional Aplikasi S2P-Mobile	50

DAFTAR GAMBAR

Gambar 2. 1 <i>Agile Development</i>	8
Gambar 3. 1 Foto Profil PT Nocola IoT Solution.....	33
Gambar 3. 2 Struktur Organisasi PT Nocola IoT Solution.....	34
Gambar 3. 3 Antarmuka Aplikasi Pokedex (<i>UI</i>).	37
Gambar 3. 4 Kode Pengambilan Data dari API.....	38
Gambar 3. 5 Hasil Tampilan Daftar Pokémon di Aplikasi.....	39
Gambar 3. 6 Kode Antarmuka Daftar Produk.	40
Gambar 3. 7 Kode Pengambilan Data Produk dari API	40
Gambar 3. 8 Tampilan Akhir Aplikasi Katalog Produk.	41
Gambar 3. 9 Diagram Siklus Agile Development pada Aplikasi S2P-Mobile.....	43
Gambar 3. 10 Desain Figma Halaman <i>Login</i> dengan CAPTCHA	45
Gambar 3. 11 Desain Figma Halaman <i>Panic Button</i>	46
Gambar 3. 12 Desain Figma Halaman Notifikasi <i>Real-Time</i>	48
Gambar 3. 13 Kode Program Halaman Login	54
Gambar 3. 14 Potongan Kode Fungsi <code>generateCaptcha()</code> dan <code>verifyCaptcha()</code>	54
Gambar 3. 15 Tampilan Akhir Halaman Login dengan CAPTCHA.....	55
Gambar 3. 16 Potongan Kode Tampilan Formulir Pelaporan Darurat pada Fitur <i>Panic Button</i> ..	56
Gambar 3. 17 Potongan Kode Fungsi <code> kirimLaporan()</code>	56
Gambar 3. 18 Tampilan Akhir Halaman <i>Panic Button</i>	57
Gambar 3. 19 Potongan Kode Tampilan Daftar Notifikasi.	58
Gambar 3. 20 Potongan Kode Fungsi <code>fetchNotifications()</code>	58
Gambar 3. 21 Potongan Kode Fungsi <code>markAsRead()</code>	58
Gambar 3. 22 Tampilan Akhir Halaman Notifikasi <i>Real-Time</i>	59

BAB I PENDAHULUAN

1.1 Latar belakang

Health, Safety, and Environment (HSE) merupakan elemen penting dalam menjaga keselamatan tenaga kerja sekaligus memastikan keberlanjutan operasi industri. Prinsip ini menjadi sangat krusial bagi sektor berisiko tinggi seperti Pembangkit Listrik Tenaga Uap (PLTU), di mana proses kerja melibatkan sistem bertekanan dan bersuhu tinggi seperti *boiler*, *economizer*, hingga instalasi listrik bertegangan besar. Jika tidak dikelola dengan baik, kondisi tersebut dapat menimbulkan kecelakaan serius yang berdampak pada keselamatan pekerja dan kestabilan produksi. Menurut (Amir et al., 2025), penerapan sistem manajemen HSE yang terencana dan berkelanjutan berperan besar dalam mengurangi angka kecelakaan kerja sekaligus meningkatkan efisiensi serta produktivitas industri.

Di sisi lain, sebagian perusahaan masih mengandalkan sistem HSE konvensional yang bergantung pada pencatatan manual dan komunikasi lisan. Pendekatan tersebut sering kali menyebabkan miskomunikasi, keterlambatan, serta ketidaktepatan dalam pelaporan karena informasi dari lapangan tidak tersampaikan secara *real-time* kepada tim pengawas. (Mouza Salsabila et al., 2023) menjelaskan bahwa pelaporan manual kerap menjadi hambatan dalam proses evakuasi maupun penanganan insiden karena data yang diterima tidak lengkap dan cenderung terlambat. Permasalahan tersebut menunjukkan perlunya penerapan sistem digital yang mampu memfasilitasi pelaporan dan pemantauan keselamatan kerja secara cepat, akurat, dan terintegrasi.

Kemajuan teknologi informasi kemudian membuka peluang baru bagi digitalisasi sistem HSE. Aplikasi berbasis *mobile* dinilai sebagai solusi paling efektif karena memiliki keunggulan dalam hal aksesibilitas, fleksibilitas, serta kemampuan menampilkan data lapangan secara langsung. (Arno et al., 2014) membuktikan bahwa penggunaan aplikasi *checklist safety* berbasis Android dengan metode *Job Safety Analysis* (JSA) mampu mempercepat proses audit sekaligus meningkatkan ketepatan hasil pemeriksaan. Hal ini diperkuat oleh temuan (Purnawati et al., 2023) yang merancang aplikasi *K3 Agent* untuk membantu petugas HSE mendokumentasikan laporan dan pedoman keselamatan secara digital, sehingga mampu mengurangi ketergantungan pada sistem manual. Lebih lanjut, (Zulkarnain et al., 2025) menegaskan bahwa penerapan notifikasi digital dan integrasi *Internet of Things* (IoT) dalam sistem pelaporan keselamatan dapat meningkatkan kecepatan respon serta memperkuat efektivitas komunikasi antar pegawai di lapangan.

Sebagai langkah strategis menuju transformasi digital, PT Sumber Segara Primadaya (S2P) mengembangkan aplikasi *S2P-Mobile* sebagai platform untuk mendukung pelaksanaan sistem HSE perusahaan. Aplikasi ini difokuskan untuk menunjang aktivitas keselamatan seperti pelaporan insiden, inspeksi lapangan, *checklist* harian, hingga distribusi informasi HSE secara digital. Dalam penelitian ini, pengembangan diarahkan pada tiga fitur utama antarmuka pengguna (*user interface*), yang memiliki peran penting bagi sistem HSE S2P:

1. *Login* dengan CAPTCHA, dikembangkan sebagai lapisan keamanan tambahan untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses aplikasi. Fitur ini penting bagi PT Sumber Segara Primadaya karena aplikasi memuat data sensitif terkait laporan insiden dan aktivitas keselamatan. CAPTCHA mencegah akses ilegal, serangan brute force, dan potensi penyalahgunaan akun, sehingga menjaga integritas dan kerahasiaan data HSE perusahaan.
2. *Panic Button*, dirancang sebagai formulir pelaporan darurat yang terstruktur dan mudah digunakan dalam kondisi kritis. Di area PLTU yang memiliki risiko kerja tinggi, kecepatan pelaporan sangat menentukan respons awal terhadap insiden. Dengan integrasi data kejadian dan lokasi otomatis (GPS), fitur ini mempermudah pekerja untuk mengirim laporan yang lengkap dalam hitungan detik. Bagi PT Sumber Segara Primadaya, fitur ini membantu mempercepat respons tim HSE, meminimalkan potensi kecelakaan lanjutan, serta meningkatkan efektivitas penanganan insiden di lapangan.
3. Notifikasi *Real-Time*, berfungsi sebagai sarana komunikasi keselamatan yang konsisten dan cepat antara manajemen HSE dan pekerja. Informasi keselamatan seperti peringatan bahaya, jadwal inspeksi, atau pengumuman penting harus tersampaikan secara serentak dan tepat waktu kepada seluruh unit kerja. Fitur ini membantu PT Sumber Segara Primadaya meningkatkan *awareness* pekerja terhadap potensi risiko, menjaga alur komunikasi darurat, serta memastikan informasi keselamatan tetap dapat diakses meskipun perangkat dalam kondisi *offline*.

Dalam proses pengembangannya, penelitian ini melibatkan peran sebagai *frontend* developer yang bertanggung jawab merancang dan mengimplementasikan antarmuka untuk ketiga fitur utama aplikasi *S2P-Mobile* menggunakan Ionic Framework. Framework ini dipilih karena kemampuannya dalam mengembangkan aplikasi lintas platform menggunakan satu basis kode, sehingga proses pengembangan menjadi lebih efisien dari segi waktu, biaya, dan pemeliharaan (Akbar et al., 2024; Bidah & Noor Islami, 2024). Pengembangan dilakukan dengan menggunakan metode *Agile Development*, yang berfokus pada iterasi cepat, fleksibilitas tinggi, serta kolaborasi

tim yang intensif agar hasil pengembangan dapat secara dinamis menyesuaikan kebutuhan pengguna. Pendekatan ini memastikan bahwa antarmuka aplikasi S2P-*Mobile* tetap adaptif, mudah digunakan, serta relevan dengan kondisi operasional di lapangan.

Selain aspek teknis, penelitian ini juga menaruh perhatian besar pada peningkatan *User Experience* (UX). Menurut (Pratama & Wahyuni, 2024) serta (Iqbal Sain et al., 2025), keberhasilan sebuah aplikasi sangat dipengaruhi oleh kualitas UX yang mencakup kemudahan navigasi, konsistensi desain, dan kejelasan informasi yang disampaikan kepada pengguna. Oleh karena itu, penelitian ini tidak hanya memberikan kontribusi akademis dalam memperluas kajian tentang pengembangan antarmuka aplikasi berbasis *mobile* untuk sistem *Health, Safety, and Environment* (HSE), tetapi juga memberikan manfaat praktis dalam mendukung digitalisasi keselamatan kerja di lingkungan industri berisiko tinggi.

1.2 Ruang Lingkup Magang

Kegiatan ini difokuskan pada pengembangan antarmuka pengguna (*User Interface*) aplikasi S2P-*Mobile* yang digunakan di lingkungan kerja PT Sumber Segara Primadaya (PLTU S2P). Lingkup penelitian dibatasi pada tiga fitur utama, yaitu:

1. Fitur *Login* dengan CAPTCHA pengembangan antarmuka yang menyediakan lapisan keamanan tambahan untuk mencegah akses yang tidak sah. Penelitian ini hanya membahas aspek tampilan dan interaksi pengguna, tanpa menyentuh sistem *backend* atau server autentikasi.
2. Fitur Panic Button implementasi antarmuka berupa formulir pelaporan darurat yang memungkinkan pengguna melaporkan insiden dengan cepat dan terstruktur. Fokus penelitian terbatas pada desain form input, validasi data, serta integrasi sederhana dengan perangkat mobile GPS.
3. Fitur Notifikasi pengembangan antarmuka untuk penyimpanan informasi keselamatan kerja secara *real-time*. Lingkungan penelitian mencakup desain tampilan notifikasi, pengelolaan status baca, dan penyimpanan lokal agar notifikasi tetap dapat diakses secara *offline*.

Penelitian ini tidak mencakup pengembangan sistem *backend*, server notifikasi, maupun integrasi penuh dengan sistem HSE perusahaan. Peran pengembang *frontend mobile* dibatasi pada aspek antarmuka, pengalaman pengguna (*user experience*), serta responsivitas aplikasi dengan memanfaatkan Ionic Framework dan metode *Agile Development*.

1.3 Tujuan

Penelitian ini dimaksudkan untuk mengembangkan antarmuka pengguna pada aplikasi S2P-Mobile yang dibangun menggunakan Ionic Framework dan pendekatan *Agile Development*, guna mendukung upaya digitalisasi *Health, Safety, and Environment* (HSE) di lingkungan PT Sumber Segara Primadaya.

1. Mendesain antarmuka fitur Login yang dilengkapi CAPTCHA sebagai tambahan lapisan keamanan, sehingga hanya pengguna yang sah saja yang dapat mengakses aplikasi.
2. Mengembangkan antarmuka fitur Panic Button berbentuk formulir pelaporan darurat yang cepat, terstruktur, serta mudah dipakai oleh pekerja di lapangan.
3. Membangun antarmuka fitur Notifikasi *Real-Time* yang memudahkan penyebaran informasi HSE secara efektif, dengan kemampuan pemantauan status baca dan akses yang tetap memungkinkan dalam mode *offline*.
4. Menerapkan proses pengembangan antarmuka pengguna berbasis Ionic Framework dengan pendekatan *Agile Development* sebagai bentuk implementasi langsung metode dan teknologi pengembangan aplikasi *mobile* di lingkungan industri.

1.4 Manfaat

Manfaat yang diantisipasi dari pengembangan aplikasi S2P-mobile ini meliputi:

1. Memperkuat keamanan akses ke aplikasi melalui fitur *Login* dengan CAPTCHA, sehingga hanya pengguna yang sah yang dapat memasuki sistem.
2. Memfasilitasi pekerja untuk melaporkan insiden darurat dengan cepat dan terstruktur berkat kehadiran fitur *Panic Button*.
3. Meningkatkan keefektifan penyebaran informasi HSE melalui fitur Notifikasi *Real-Time* yang tetap dapat diakses walaupun dalam situasi *offline*.
4. Memberikan pengalaman praktis bagi peneliti dalam menerapkan antarmuka pengguna (UI) berbasis Ionic Framework dengan metode *Agile Development* di lingkungan industri.

1.5 Sistematika Penulisan

Laporan tugas akhir ini disusun secara sistematis agar memudahkan pembaca dalam memahami proses pengembangan antarmuka aplikasi *mobile* berbasis *Health, Safety & Environment* (HSE) menggunakan Ionic Framework. Adapun sistematika penulisan laporan ini adalah sebagai berikut:

- a. BAB I Pendahuluan, Bab ini membahas pendahuluan, latar belakang, ruang lingkup magang, tujuan, manfaat, serta sistematika penulisan dari laporan tugas akhir. Bagian ini memberikan gambaran umum mengenai alasan pemilihan topik, konteks proyek pengembangan, dan kontribusi peneliti dalam kegiatan magang.
- b. BAB II Dasar Teori, Bab ini membahas landasan teori dan tinjauan pustaka yang berkaitan dengan pengembangan aplikasi S2P-Mobile, termasuk teori tentang Keselamatan dan Kesehatan Kerja (K3), konsep *Health, Safety, & Environment (HSE)*, *User Interface (UI)* dan *User Experience (UX)*, *Agile Development*, serta *Ionic Framework* dan *Capacitor Preferences*. Bab ini juga menyajikan penelitian terdahulu sebagai acuan pengembangan proyek.
- c. BAB III Pelaksanaan Magang, Bab ini menjelaskan secara rinci proses pelaksanaan magang di PT Nocola IoT Solution, yang mencakup manajemen proyek, tahapan pengembangan aplikasi, implementasi fitur utama seperti *Login* dengan *CAPTCHA*, *Panic Button*, dan *Notifikasi Real-Time*, serta hasil dan pembahasan pengujian aplikasi.
- d. BAB IV Refleksi Pelaksanaan Magang, Bab ini berisi refleksi dan pembelajaran yang diperoleh selama menjalani kegiatan magang, termasuk perbandingan antara teori yang telah dipelajari di perkuliahan dan praktik yang diterapkan di industri. Bab ini juga menguraikan manfaat, kendala, serta tantangan yang dihadapi selama proses magang.
- e. BAB V Penutup, Bab ini memuat kesimpulan dari hasil pelaksanaan magang serta saran yang dapat digunakan sebagai bahan pertimbangan untuk pengembangan proyek atau penelitian selanjutnya.

BAB II DASAR TEORI

2.1 Health Safety and Environment (HSE)

Health, Safety, and Environment (HSE) merupakan aspek penting dalam dunia industri yang berfungsi untuk menjamin keselamatan tenaga kerja, menjaga kelestarian lingkungan, serta memastikan keberlangsungan operasional perusahaan. HSE tidak hanya menjadi bentuk kepatuhan terhadap regulasi pemerintah, tetapi juga bagian dari strategi manajemen risiko yang berperan dalam menjaga efisiensi dan produktivitas kerja. (Amir et al., 2025) menegaskan bahwa penerapan sistem manajemen HSE yang terencana dan dilakukan secara berkesinambungan dapat menurunkan angka kecelakaan kerja, meningkatkan kesadaran pekerja terhadap keselamatan, serta berkontribusi terhadap peningkatan kinerja industri secara menyeluruh.

Meski demikian, pelaksanaan HSE di berbagai perusahaan masih menghadapi kendala, terutama pada sistem pelaporan yang belum sepenuhnya terintegrasi secara digital. Penggunaan metode manual seperti pencatatan kertas dan laporan lisan sering menimbulkan keterlambatan informasi, kesalahan input, bahkan kehilangan data penting. Berdasarkan penelitian (Askar et al., 2021), sistem pelaporan keselamatan kerja yang masih berbasis manual di fasilitas kesehatan menyebabkan proses tindak lanjut menjadi lebih lambat karena tidak adanya umpan balik langsung antara pelapor dan pihak pengawas. Hal tersebut menunjukkan pentingnya penerapan sistem digital agar komunikasi dan pelaporan dapat berjalan lebih cepat, tepat, dan efisien.

Perkembangan teknologi informasi membawa peluang besar bagi penguatan implementasi HSE melalui digitalisasi proses kerja. Aplikasi *mobile* menjadi salah satu solusi yang efektif karena memungkinkan petugas untuk melakukan pelaporan langsung di lokasi kerja secara *real-time*, sehingga data keselamatan dapat dikelola dengan lebih cepat dan akurat. (Purnawati et al., 2023) mengembangkan aplikasi “K3 Agent” berbasis Android yang membantu petugas HSE dalam mendokumentasikan laporan, memantau aktivitas, serta menyebarkan informasi keselamatan kerja secara lebih efisien. Temuan ini sejalan dengan hasil penelitian (Arno et al., 2014), yang menunjukkan bahwa penggunaan aplikasi Checklist Safety berbasis *Job Safety Analysis* (JSA) mampu mempercepat proses audit dan meningkatkan ketepatan hasil pemeriksaan dibandingkan sistem pelaporan manual.

Dengan demikian, digitalisasi sistem HSE dapat menjadi langkah strategis untuk membangun budaya kerja yang lebih adaptif dan berorientasi pada keselamatan. Pemanfaatan teknologi *mobile* tidak hanya mempercepat penyampaian informasi dan meningkatkan akurasi

data, tetapi juga memperkuat koordinasi antara pekerja dan pengawas. Transformasi digital dalam sistem HSE diharapkan dapat menciptakan lingkungan kerja yang aman, efisien, dan selaras dengan prinsip industri 4.0.

2.2 Antarmuka Pengguna (*User Interface*) dan Pengalaman Pengguna (*User Experience*)

Antarmuka Pengguna (UI) berfungsi sebagai penghubung utama antara pengguna dan sistem, sedangkan Pengalaman Pengguna (UX) mencakup keseluruhan persepsi, kemudahan, serta tingkat kenyamanan yang dirasakan selama berinteraksi dengan aplikasi. Desain UI/UX yang ideal tidak hanya berfokus pada tampilan visual, tetapi juga menjadi dasar dalam implementasi antarmuka frontend yang efisien dan konsisten. Menurut (Pratama & Wahyuni, 2024), penerapan prinsip UI/UX yang tepat membantu pengguna memahami fungsi-fungsi aplikasi dengan cepat tanpa memerlukan proses adaptasi yang panjang, karena antarmuka yang intuitif dan responsif mampu menciptakan kesan positif terhadap sistem yang digunakan.

Dalam konteks aplikasi yang mendukung sistem Kesehatan, Keselamatan, dan Lingkungan (HSE), desain UI/UX memiliki peran penting dalam memastikan pengguna dapat mengoperasikan fitur secara efisien, terutama pada situasi yang membutuhkan respons cepat seperti keadaan darurat. Fitur-fitur seperti *Login*, *Panic Button*, dan Notifikasi *Real-Time* harus dirancang dengan memperhatikan aspek aksesibilitas, keterbacaan ikon, dan kejelasan tata letak agar mudah digunakan di lapangan. (Iqbal Sain et al., 2025) menekankan bahwa desain yang berpusat pada pengguna (*User-Centered Design*) dapat meningkatkan efektivitas interaksi sekaligus meminimalkan risiko kesalahan saat mengoperasikan aplikasi, khususnya dalam konteks pekerjaan dengan tingkat risiko tinggi.

Lebih lanjut, (Alamsyah et al., 2022) mengemukakan bahwa keberhasilan pengalaman pengguna tidak hanya ditentukan oleh keindahan visual, tetapi juga oleh konsistensi desain antar elemen serta kesesuaian fungsi aplikasi dengan kebutuhan pengguna. Implementasi desain UI/UX yang baik mampu mendorong keterlibatan pengguna, menumbuhkan rasa percaya terhadap sistem, dan menciptakan pengalaman penggunaan yang nyaman serta efisien. Prinsip UI/UX digunakan sebagai acuan dalam implementasi antarmuka frontend pada tiga fitur utama aplikasi S2P-Mobile, yaitu *Login* dengan CAPTCHA, *Panic Button*, dan Notifikasi *Real-Time*. Tujuannya adalah untuk menghasilkan antarmuka yang tidak hanya menarik secara visual, tetapi juga mudah digunakan, efisien, dan sesuai dengan kebutuhan operasional di lingkungan industri berisiko tinggi.

2.3 Agile Development



Gambar 2. 1 *Agile Development*
Sumber: (D'Ambra (n.d.))

Metode *Agile Development* merupakan pendekatan modern dalam pengembangan perangkat lunak yang menitikberatkan pada fleksibilitas, komunikasi tim yang efektif, serta perbaikan yang dilakukan secara berulang (*iterative improvement*). Proses ini dibagi menjadi beberapa siklus pendek atau *sprint*, di mana setiap siklus menghasilkan bagian sistem yang dapat digunakan dan dievaluasi. Menurut (Akwilla & Jollyta, 2024), kerangka kerja *Scrum* dalam *Agile* mendorong kolaborasi lintas fungsi dan keterlibatan aktif seluruh anggota tim, sehingga setiap perubahan kebutuhan pengguna dapat direspons secara cepat. Model ini juga membantu pengembang menyesuaikan arah proyek berdasarkan umpan balik pengguna secara langsung, menjadikan proses pengembangan lebih adaptif dan terukur.

Dalam konteks pengembangan aplikasi mobile, *Agile* memberikan keuntungan signifikan karena memungkinkan pengembang untuk melakukan evaluasi dan penyempurnaan desain antarmuka secara berkelanjutan. (Suhari et al., 2022) menegaskan bahwa pendekatan *Agile* menciptakan alur kerja yang dinamis dan terorganisir, terutama pada proyek lintas platform seperti *Ionic Framework*. Dengan prinsip *continuous improvement* dan *user collaboration*, pengembang dapat melakukan revisi secara cepat sesuai kebutuhan pengguna. Setiap *sprint* umumnya diakhiri dengan sesi evaluasi berupa *review* dan *retrospective* untuk menilai kemajuan proyek, mengidentifikasi kendala teknis, serta menyusun langkah-langkah perbaikan untuk iterasi berikutnya.

Dalam penelitian ini, metode *Agile* digunakan dalam pengembangan tiga fitur utama aplikasi S2P-Mobile, yaitu *Login* dengan CAPTCHA, *Panic Button*, dan *Notifikasi Real-Time*. Melalui pembagian pekerjaan berdasarkan *sprint*, peneliti sebagai pengembang frontend mobile dapat melakukan iterasi desain dan pengujian fungsional secara rutin. Pendekatan ini memungkinkan

proses pengembangan berjalan lebih efisien, terstruktur, dan kolaboratif, sekaligus memastikan bahwa setiap komponen aplikasi sesuai dengan kebutuhan pengguna di lapangan. Penerapan metode Agile juga memperkuat integrasi antara desain antarmuka dan sistem HSE, sehingga hasil pengembangan tidak hanya fungsional, tetapi juga responsif terhadap perubahan dan tantangan di lingkungan industri.

2.4 Pengembangan Frontend Aplikasi Mobile

Pengembangan frontend aplikasi mobile merupakan proses implementasi lapisan antarmuka sistem yang berfungsi sebagai penghubung langsung antara pengguna dan sistem backend. Pada lapisan ini, frontend bertanggung jawab dalam menampilkan informasi, menerima input pengguna, mengelola alur interaksi, serta memastikan setiap fungsi sistem dapat diakses dan dijalankan sesuai dengan kebutuhan operasional. Berbeda dengan perancangan *User Interface* (UI) dan *User Experience* (UX) yang berfokus pada aspek visual dan pengalaman pengguna, pengembangan frontend menitikberatkan pada aspek teknis implementasi antarmuka, validasi input, pengelolaan state aplikasi, serta integrasi dengan layanan backend melalui *Application Programming Interface* (API).

Dalam aplikasi S2P-Mobile, pengembangan frontend berperan penting dalam mendukung digitalisasi sistem *Health, Safety, and Environment* (HSE). Seluruh interaksi pengguna dengan sistem, mulai dari proses autentikasi, pelaporan keadaan darurat, hingga penerimaan informasi keselamatan, diimplementasikan pada sisi frontend. Oleh karena itu, frontend tidak hanya berfungsi sebagai tampilan visual, tetapi juga sebagai lapisan kontrol yang memastikan alur proses bisnis sistem berjalan secara benar dan terstruktur sebelum data diteruskan ke backend.

Implementasi frontend pada S2P-Mobile dilakukan menggunakan Ionic Framework berbasis Angular. Ionic menyediakan komponen antarmuka siap pakai yang mendukung pengembangan lintas platform, sedangkan Angular berperan dalam pengelolaan logika aplikasi, pengikatan data (data binding), pengaturan navigasi, serta pengelolaan alur proses antarkomponen. Melalui pendekatan ini, aplikasi S2P-Mobile dapat berjalan secara konsisten pada perangkat Android dan iOS dengan satu basis kode, sehingga proses pengembangan dan pemeliharaan aplikasi menjadi lebih efisien.

Pada fitur autentikasi pengguna, frontend bertugas menangani validasi awal input pengguna, termasuk pemeriksaan kelengkapan data dan verifikasi CAPTCHA sebelum permintaan *login* dikirimkan ke *backend*. Proses ini memastikan bahwa hanya data yang telah divalidasi di sisi klien yang diteruskan ke sistem, sehingga membantu meningkatkan keamanan dan mengurangi beban

pemrosesan pada server. Setelah autentikasi berhasil, frontend mengelola penyimpanan data sesi pengguna pada perangkat menggunakan mekanisme penyimpanan lokal.

Pada fitur Panic Button, frontend berperan dalam menampilkan formulir pelaporan darurat, memvalidasi data yang dimasukkan pengguna, serta mengakses fitur perangkat seperti lokasi (geolocation) melalui integrasi dengan Capacitor. Data laporan yang telah tervalidasi kemudian dikirimkan ke backend untuk diproses lebih lanjut. Dengan demikian, frontend berfungsi sebagai pengendali utama alur pelaporan darurat yang cepat dan terstruktur di lingkungan kerja.

Sementara itu, pada fitur Notifikasi *Real-Time*, frontend bertanggung jawab dalam menerima dan menyajikan informasi notifikasi kepada pengguna, mengelola status pesan (dibaca atau belum dibaca), serta menyimpan data notifikasi pada penyimpanan lokal agar tetap dapat diakses ketika jaringan tidak tersedia. Implementasi ini mendukung konsep offline-first yang penting bagi aplikasi yang digunakan di area operasional dengan keterbatasan konektivitas jaringan.

Berdasarkan implementasi tersebut, pengembangan frontend pada aplikasi S2P-Mobile tidak hanya berfungsi sebagai lapisan presentasi, tetapi juga sebagai bagian integral dari sistem yang menghubungkan pengguna dengan proses bisnis HSE secara digital. Dengan pengelolaan logika interaksi, validasi data, dan integrasi layanan backend yang tepat, frontend menjadi komponen penting dalam memastikan sistem S2P-Mobile dapat digunakan secara efektif, aman, dan andal di lingkungan industri.

2.5 Ionic Framework

Ionic Framework merupakan salah satu kerangka kerja sumber terbuka yang dirancang untuk membangun aplikasi mobile lintas platform menggunakan teknologi web seperti HTML, CSS, dan JavaScript. Framework ini memungkinkan pengembang membuat satu basis kode (*single codebase*) yang dapat dijalankan di berbagai sistem operasi, termasuk Android dan iOS. Integrasi Ionic dengan *Angular* dan *Capacitor* memberikan kemudahan dalam mengakses fitur-fitur perangkat asli seperti kamera, GPS, serta sistem notifikasi. (Akbar et al., 2024) menyebutkan bahwa Ionic menawarkan efisiensi tinggi dalam proses pengembangan karena dapat menghasilkan aplikasi *hybrid* dengan performa mendekati aplikasi *native*, namun dengan kebutuhan waktu dan biaya yang lebih rendah.

Selain efisiensi, *Ionic* juga menyediakan berbagai komponen antarmuka siap pakai (pre-built UI components) yang mengikuti standar desain dari material *design* maupun iOS *Human Interface Guidelines*. Hal ini memudahkan pengembang dalam menciptakan antarmuka yang seragam di

berbagai perangkat tanpa perlu melakukan penyesuaian besar terhadap kode. (Ariesta et al., 2024) menjelaskan bahwa arsitektur *webview* pada Ionic memungkinkan integrasi sistem secara fleksibel serta mempermudah proses pemeliharaan aplikasi, terutama dalam pengembangan berskala besar yang menuntut ketahanan dan konsistensi tampilan.

Selanjutnya, (Bidah & Noor Islami, 2024) menegaskan bahwa Ionic merupakan solusi yang ideal bagi organisasi yang memerlukan keseimbangan antara performa aplikasi dan kompatibilitas lintas platform. Framework ini memungkinkan pengembang untuk mempercepat siklus iterasi serta menerapkan fitur-fitur antarmuka yang kompleks tanpa mengorbankan kestabilan aplikasi.

Dalam konteks penelitian ini, Ionic Framework dipilih karena kemampuannya yang unggul dalam mendukung pengembangan antarmuka pada tiga fitur utama aplikasi S2P-Mobile, yaitu *Login* dengan CAPTCHA, *Panic Button*, dan *Notifikasi Real-Time*. Dengan pendekatan *cross-platform*, Ionic memungkinkan proses pengembangan berlangsung lebih cepat, adaptif, dan efisien, sekaligus menjaga konsistensi tampilan dan fungsionalitas aplikasi di berbagai perangkat mobile. Keunggulan ini menjadikan Ionic sebagai teknologi yang tepat dalam upaya mendukung digitalisasi sistem *Health, Safety, and Environment* (HSE) di PT Sumber Segara Primadaya (S2P), terutama untuk memastikan akses informasi yang cepat, stabil, dan mudah digunakan di lingkungan kerja lapangan.

2.6 Fitur Login dengan CAPTCHA

CAPTCHA (*Completely Automated Public Turing Test to Tell Computers and Humans Apart*) merupakan sistem autentikasi yang dirancang untuk membedakan interaksi pengguna manusia dari program otomatis (bot). Teknologi ini banyak digunakan pada halaman *login* untuk melindungi sistem dari serangan *brute force*, pembuatan akun palsu, serta aktivitas otomatis lainnya yang dapat membahayakan keamanan data. (Andarini et al., 2023) menjelaskan bahwa penerapan CAPTCHA berbasis teks dengan tampilan sederhana dan kontras warna yang jelas mampu meningkatkan keberhasilan pengguna manusia dalam proses verifikasi tanpa menurunkan tingkat keamanan aplikasi. Pendekatan tersebut menekankan pentingnya keseimbangan antara keamanan dan kemudahan penggunaan.

Namun demikian, pengembangan CAPTCHA masih menghadapi tantangan dalam menjaga kenyamanan pengguna sekaligus mempertahankan ketahanan sistem terhadap serangan otomatis. (Arifin & Yunus, 2025) menegaskan bahwa desain CAPTCHA yang terlalu rumit justru dapat menurunkan pengalaman pengguna, terutama pada aplikasi *mobile* yang menuntut efisiensi waktu

dan kemudahan navigasi. Oleh karena itu, rancangan CAPTCHA perlu mempertimbangkan aspek fungsional dan pengalaman pengguna agar tetap efektif dan mudah digunakan.

Selain itu, (Adesina et al., 2023) mengembangkan sistem CAPTCHA berbasis kombinasi teks dan gambar yang mengharuskan interaksi langsung antara pengguna dengan elemen visual tertentu. Metode ini memperkuat proses autentikasi karena pengguna harus melakukan tindakan spesifik untuk membuktikan identitasnya sebagai manusia. Sejalan dengan itu, (Dinh & Hoang, 2023) memperkenalkan CAPTCHA adaptif yang memanfaatkan pola perilaku pengguna sebagai bagian dari proses verifikasi, sehingga sistem menjadi lebih interaktif dan ramah bagi pengguna perangkat *mobile* tanpa mengurangi tingkat keamanannya.

Berdasarkan prinsip-prinsip tersebut, fitur Login dengan CAPTCHA pada aplikasi S2P-Mobile dikembangkan sebagai lapisan keamanan tambahan untuk mencegah akses tidak sah. Desain antarmuka dibuat ringkas dan mudah dipahami agar pekerja dapat melakukan proses *login* dengan cepat, bahkan dalam kondisi kerja dengan tekanan waktu tinggi. Melalui integrasi antara Ionic Framework dan *Angular*, sistem CAPTCHA dikembangkan menggunakan fungsi `generateCaptcha()` untuk menghasilkan kode acak serta `verifyCaptcha()` untuk memvalidasi masukan pengguna secara langsung di sisi *frontend*. Pendekatan ini memastikan proses autentikasi berlangsung aman, efisien, dan tetap mempertahankan kenyamanan pengguna sesuai dengan prinsip UI/UX modern.

2.7 Fitur Panic Button

Fitur Panic Button pada aplikasi S2P-Mobile berfungsi sebagai sarana pelaporan insiden darurat yang dirancang agar proses penyampaian informasi dapat dilakukan dengan cepat, akurat, dan terorganisir. Melalui antarmuka sederhana berupa tombol dan formulir digital, fitur ini memungkinkan pekerja untuk segera melaporkan kejadian di lapangan. (Sultan, 2021) menekankan bahwa sistem pelaporan berbasis digital yang efektif harus memberikan akses cepat serta alur pelaporan yang jelas, sehingga informasi terkait keadaan darurat dapat diterima dengan segera oleh pihak yang berwenang tanpa kendala teknis.

Selain menonjolkan aspek kecepatan, sistem pelaporan juga dituntut untuk memastikan keakuratan dan kelengkapan data agar tindak lanjut dapat dilakukan secara tepat. (Ali et al., 2024) menjelaskan bahwa penggunaan aplikasi *mobile* dengan formulir pelaporan terstruktur membantu meminimalkan kesalahan pencatatan sekaligus mempercepat proses penanganan insiden. Transformasi dari sistem manual ke digital ini terbukti meningkatkan efisiensi kerja serta mempermudah proses dokumentasi keselamatan di lapangan.

Penelitian (Dewantara et al., 2022) memperkuat pandangan tersebut dengan menunjukkan bahwa digitalisasi pelaporan keselamatan memungkinkan data disampaikan secara *Real-Time* dan tersampaikan dengan baik, sehingga mempermudah proses audit maupun evaluasi keselamatan kerja. Selain itu, (Yogama et al., 2022) menegaskan bahwa antarmuka yang sederhana dan responsif menjadi faktor kunci dalam mendukung efektivitas sistem pelaporan berbasis aplikasi, terutama pada kondisi darurat yang menuntut kecepatan serta kemudahan penggunaan.

Dalam penelitian ini, fitur Panic Button dikembangkan dalam bentuk formulir pelaporan digital yang berisi elemen masukan seperti jenis kejadian, unit kerja, lokasi, dan keterangan tambahan. Antarmuka dirancang agar mudah dioperasikan dan tetap responsif di berbagai perangkat, sehingga laporan dapat dikirimkan dalam hitungan detik. Melalui pemanfaatan *Capacitor Geolocation* pada Ionic Framework, sistem secara otomatis merekam koordinat lokasi pengguna dan mengirimkannya bersama laporan ke server. Pendekatan ini menjadikan proses pelaporan lebih efisien, data yang diterima lebih lengkap, dan komunikasi antara pekerja serta tim pengawas menjadi lebih efektif. Dengan demikian, fitur Panic Button menjadi bagian penting dalam mendukung digitalisasi sistem pelaporan HSE di PT Sumber Segara Primadaya (S2P).

2.8 Fitur Notifikasi

Fitur Notifikasi *Real-Time* merupakan salah satu elemen utama dalam aplikasi S2P-Mobile yang berperan penting dalam penyampaian informasi secara cepat dan tepat kepada pengguna. Dalam konteks *Health, Safety, and Environment* (HSE), fitur ini berfungsi sebagai sarana komunikasi yang memungkinkan penyebaran pesan keselamatan, peringatan darurat, maupun informasi penting lainnya kepada pekerja di lapangan. Menurut (Susanto et al., 2024), penerapan sistem pelaporan digital seperti *Unsafe Action* dan *Unsafe Condition* terbukti mampu meningkatkan kecepatan tanggapan terhadap potensi bahaya sekaligus memperkuat budaya keselamatan di lingkungan kerja industri.

Agar notifikasi berfungsi optimal, perancangannya harus memperhatikan kejelasan isi pesan, waktu pengiriman, serta relevansi konteks dengan kebutuhan pengguna. (Mirza et al., 2024) menyatakan bahwa sistem notifikasi yang dirancang dengan baik mampu menyampaikan informasi penting secara cepat tanpa mengganggu aktivitas kerja pengguna, terutama pada situasi kritis yang menuntut tindakan segera. Hasil penelitian mereka menunjukkan bahwa penerapan *push notification* berbasis Android memungkinkan pesan keselamatan tetap diterima secara *Real-Time* meskipun aplikasi tidak sedang dijalankan di latar depan.

Dari sisi tampilan, efektivitas komunikasi juga sangat dipengaruhi oleh rancangan antarmuka pengguna. (Zulkarnain et al., 2025) menegaskan bahwa struktur pesan yang singkat, penggunaan warna yang kontras, serta tata letak yang sederhana dapat meningkatkan keterbacaan dan keterlibatan pengguna terhadap notifikasi yang diterima. Desain visual yang baik memastikan pesan keselamatan dapat segera dipahami tanpa menyebabkan kebingungan atau gangguan pada pekerjaan pengguna.

Dalam penelitian ini, fitur Notifikasi *Real-Time* pada aplikasi S2P-Mobile dikembangkan untuk mengirimkan informasi keselamatan dan peringatan penting secara langsung kepada pengguna. Fitur ini dilengkapi dengan indikator status baca (read/unread) serta dukungan akses *offline* melalui penyimpanan lokal menggunakan Capacitor Preferences. Dengan integrasi teknologi tersebut, sistem tetap mampu menampilkan dan menyimpan notifikasi secara lokal melalui Capacitor Preferences meskipun koneksi internet sedang tidak tersedia. Pendekatan tersebut diharapkan dapat meningkatkan efektivitas komunikasi HSE di lingkungan PT Sumber Segara Primadaya (S2P), sehingga penyampaian pesan keselamatan menjadi lebih cepat, efisien, dan berkelanjutan.

2.9 Tinjauan Pustaka

Beragam penelitian telah dilakukan dalam rangka mendukung transformasi digital pada sistem *Health, Safety, and Environment* (HSE) serta meningkatkan efektivitas penerapan keselamatan kerja melalui pemanfaatan teknologi berbasis mobile. Secara umum, tujuan dari berbagai penelitian tersebut adalah mempercepat proses pelaporan, meningkatkan ketepatan data, serta memperkuat komunikasi dua arah antara pekerja dan pihak manajemen. Berdasarkan hasil penelusuran literatur, penelitian-penelitian terdahulu dapat diklasifikasikan ke dalam beberapa fokus utama, yaitu pengembangan aplikasi HSE berbasis *mobile* dan web, penerapan teknologi keamanan digital berbasis CAPTCHA, pengembangan fitur Panic Button, implementasi sistem notifikasi keselamatan secara *real-time*, penerapan kerangka kerja lintas platform seperti Ionic, serta pengembangan antarmuka pengguna (UI/UX) dengan pendekatan *Agile Development*.

Fokus pertama membahas mengenai digitalisasi sistem HSE yang dikembangkan melalui platform *mobile*. Sejumlah penelitian terdahulu menunjukkan bahwa digitalisasi mampu meningkatkan kecepatan serta efektivitas pelaporan di lingkungan kerja. (Purnawati et al., 2023) mengembangkan aplikasi K3 Agent berbasis Android untuk mendukung proses audit dan dokumentasi keselamatan kerja secara digital, yang terbukti mampu mengefisienkan proses administrasi dan pelaporan HSE. Senada dengan itu, (Arno et al., 2014) menerapkan aplikasi

checklist safety berbasis *Job Safety Analysis* (JSA) untuk mempercepat proses inspeksi dan meningkatkan akurasi hasil pemeriksaan lapangan. Selain itu, (Amir et al., 2025) melalui implementasi sistem pelaporan Unsafe Action dan Unsafe Condition menunjukkan bahwa digitalisasi pelaporan mampu mempercepat penanganan potensi bahaya sekaligus memperkuat budaya keselamatan di lingkungan kerja. Hasil ini sejalan dengan temuan (Askar et al., 2021) yang mengembangkan sistem pelaporan berbasis web untuk meningkatkan efektivitas pengawasan serta mempercepat respon terhadap kejadian di tempat kerja. Berdasarkan keseluruhan temuan tersebut, dapat disimpulkan bahwa penerapan teknologi digital berperan penting dalam mewujudkan sistem pelaporan HSE yang cepat, akurat, dan terdokumentasi dengan baik.

Fokus kedua berhubungan dengan peningkatan keamanan sistem aplikasi melalui penerapan teknologi CAPTCHA. (Adesina et al., 2023) menunjukkan bahwa CAPTCHA berbasis teks tetap menjadi metode autentikasi yang efisien untuk aplikasi *mobile* karena mudah diimplementasikan dan tidak membebani sistem. (Andarini et al., 2023) merancang CAPTCHA gabungan antara teks dan citra guna memperkuat keamanan proses *login* tanpa mengurangi kenyamanan pengguna. (Arifin & Yunus, 2025) mengembangkan CAPTCHA adaptif yang dapat menyesuaikan tingkat kesulitan berdasarkan perilaku pengguna, sedangkan (Dinh & Hoang, 2023) memperkenalkan CAPTCHA berbasis perilaku (behavioral CAPTCHA) yang lebih interaktif dan ramah pengguna. Hasil-hasil penelitian tersebut menjadi dasar dalam penerapan fitur *Login* dengan CAPTCHA pada aplikasi S2P-Mobile, yang menggabungkan aspek keamanan dengan kemudahan penggunaan untuk mendukung efisiensi kerja di lingkungan industri.

Fokus berikutnya berkaitan dengan fitur pelaporan darurat digital atau Panic Button yang berfungsi untuk mempercepat proses pelaporan insiden secara terstruktur dan *real-time*. (Sultan, 2021) menegaskan pentingnya antarmuka yang sederhana dan mudah diakses agar pengguna dapat segera melakukan pelaporan dalam situasi darurat. (Ali et al., 2024) menyoroti bahwa formulir pelaporan berbasis digital yang dilengkapi dengan pengisian terstruktur dapat mempercepat tindak lanjut serta meningkatkan keakuratan data yang diterima oleh pihak pengawas. Di sisi lain, (Dewantara et al., 2022) menekankan bahwa sistem pelaporan berbasis aplikasi mampu mempercepat alur informasi dan memperkuat dokumentasi keselamatan kerja. Sementara itu, (Yogama et al., 2022) menambahkan bahwa antarmuka pengguna (UI) yang intuitif dapat membantu pekerja melakukan pelaporan tanpa hambatan teknis di lapangan. Berdasarkan hasil-hasil tersebut, fitur Panic Button pada aplikasi S2P-Mobile dirancang menggunakan formulir digital terintegrasi dengan *Capacitor* Geolocation untuk mendeteksi lokasi pengguna secara otomatis.

Selain pelaporan darurat, sistem notifikasi *Real-Time* juga merupakan elemen penting dalam digitalisasi HSE, karena berfungsi sebagai sarana komunikasi keselamatan yang efisien antara pekerja dan manajemen. (Susanto et al., 2024) membuktikan bahwa pelaporan digital Unsafe Action dan Unsafe Condition dapat mempercepat respon terhadap potensi bahaya dan meningkatkan budaya keselamatan di tempat kerja. (Zulkarnain et al., 2025) mengembangkan sistem *push notification* berbasis *mobile* untuk memastikan pesan keselamatan dapat diterima pengguna secara langsung bahkan saat koneksi internet terbatas. (Mirza et al., 2024) menekankan bahwa efektivitas notifikasi sangat dipengaruhi oleh desain pesan yang jelas, singkat, serta tampilan visual yang menarik. Ketiga penelitian tersebut menjadi dasar pengembangan fitur Notifikasi *Real-Time* pada aplikasi S2P-Mobile, yang memanfaatkan penyimpanan lokal berbasis Capacitor Preferences sehingga pesan tetap dapat diakses meskipun perangkat tidak terhubung ke jaringan.

Dari sisi teknologi pengembangan, (Akbar et al., 2024), (Bidah & Noor Islami, 2024), serta (Ariesta et al., 2024) menegaskan bahwa penggunaan Ionic Framework memberikan keunggulan dari segi efisiensi, kompatibilitas, dan waktu pengembangan. Ionic memungkinkan aplikasi lintas platform dijalankan hanya dengan satu basis kode, yang kemudian dapat dikompilasi untuk Android dan iOS. Integrasinya dengan *Angular* dan *Capacitor* juga mendukung penggunaan fitur perangkat keras seperti kamera, GPS, dan notifikasi. Hal ini menjadikan Ionic sebagai solusi ideal dalam pengembangan aplikasi industri seperti S2P-Mobile yang memerlukan kinerja tinggi dan pemeliharaan yang efisien.

Dalam konteks desain antarmuka, (Pratama & Wahyuni, 2024) menegaskan bahwa kualitas pengalaman pengguna (UX) menjadi kunci utama keberhasilan aplikasi *mobile*, yang mencakup kemudahan navigasi dan kejelasan tampilan. (Iqbal Sain et al., 2025) menyatakan bahwa desain antarmuka berbasis empati pengguna dapat meningkatkan kenyamanan interaksi serta mengurangi kesalahan penggunaan aplikasi. Selain itu, (Alamsyah et al., 2022) menambahkan bahwa konsistensi visual dan tata letak yang terstruktur mampu memperkuat kepercayaan pengguna terhadap sistem digital. Prinsip-prinsip ini menjadi landasan utama dalam implemementasi antarmuka aplikasi S2P-Mobile agar mudah digunakan, adaptif, serta sesuai dengan karakteristik pengguna industri.

Dari sisi metodologi, (Suhari et al., 2022) menyebutkan bahwa penerapan *Agile Development* dengan kerangka kerja Scrum dapat mempercepat proses pengembangan serta memudahkan tim untuk beradaptasi terhadap perubahan kebutuhan pengguna. Temuan ini diperkuat oleh (Akwilla & Jollyta, 2024) yang menekankan bahwa Agile mendorong iterasi

singkat dan kolaborasi intensif antaranggota tim, sehingga setiap fitur yang dikembangkan dapat diuji dan disesuaikan secara berkelanjutan.

Secara keseluruhan, kajian pustaka ini menunjukkan adanya pergeseran paradigma menuju digitalisasi sistem HSE melalui pengembangan aplikasi lintas platform yang responsif, aman, dan ramah pengguna. Namun, sebagian besar penelitian terdahulu masih terbatas pada fungsi pelaporan dan keamanan, belum secara mendalam mengeksplorasi desain antarmuka lintas platform yang berfokus pada pengalaman pengguna di sektor industri berisiko tinggi. Oleh karena itu, penelitian ini berkontribusi dengan mengembangkan antarmuka pengguna aplikasi S2P-Mobile berbasis Ionic Framework, yang terdiri dari tiga fitur utama Login dengan CAPTCHA, Panic Button, dan Notifikasi *Real-Time* guna mendukung transformasi digital sistem HSE di PT Sumber Segara Primadaya (S2P). Berikut ini merupakan ringkasan beberapa penelitian terdahulu yang menjadi acuan dalam pengembangan aplikasi S2P-Mobile yang disajikan pada Tabel 2. 1 Tinjauan Pustaka

Tabel 2. 1 Tinjauan Pustaka

No	Peneliti	Judul	Metode	Hasil Penelitian
1.	Purnawati, D., Yuliana, A., & Ramadhan, M. (2023)	Design of HSE Digital Application with Android-Based <i>User Interface</i>	Studi pengembangan aplikasi berbasis Android	Aplikasi digital HSE berbasis Android meningkatkan efisiensi komunikasi dan dokumentasi keselamatan kerja di lapangan.
2.	Arno, M. A. Y. S., Fadillah, M. N., & Setiadi, D. (2014)	Penerapan Aplikasi Checklist Safety Berbasis Job Safety Analysis (JSA)	Studi implementasi lapangan	Sistem digital berbasis checklist mempercepat proses audit K3 dan meningkatkan akurasi pemeriksaan dibandingkan sistem manual.
3.	Firman Amir, Ahmad A. Hanafiyah, Fariz R. Maulana, Nursaidah, & Abdullah M. Ahsan (2025)	Implementasi Program Unsafe Action dan Unsafe Condition pada Sistem Pelaporan HSE Digital	Studi deskriptif kuantitatif	Digitalisasi pelaporan HSE mempercepat respon terhadap potensi bahaya dan meningkatkan budaya keselamatan kerja.
4.	Askar, N. F., Herawati, D., Susilawati, & Pratomo, D. N. (2022)	Pengembangan Sistem Pelaporan Keselamatan Berbasis Web pada Fasilitas Kesehatan	Penelitian pengembangan sistem	Sistem berbasis web mempermudah proses pengawasan serta mempercepat respon terhadap insiden keselamatan kerja.
5.	Adesina, A. O., Ayobioloja, P. S.,	Comparative Study of Text	Analisis komparatif	CAPTCHA berbasis teks lebih efisien untuk perangkat mobile

	& Obagbuwa, I. C. (2023)	and Graphical CAPTCHA Approaches		dibandingkan CAPTCHA berbasis gambar yang lebih kompleks.
6.	Andarini, R. Y., Hendradi, P., & Nugroho, S. (2023)	Hybrid CAPTCHA Design for Securing User Login on Web and Mobile Systems	Studi eksperimental	Kombinasi CAPTCHA teks dan gambar mampu meningkatkan keamanan <i>login</i> tanpa menurunkan kenyamanan pengguna.
7.	Arifin, A., & Yunus, M. (2023)	Adaptive CAPTCHA System Based on User Behavior	Penelitian eksperimental	CAPTCHA adaptif menyesuaikan tingkat kesulitan berdasarkan perilaku pengguna, meningkatkan keamanan autentikasi.
8.	Nghia Trong Dinh & Vinh Truong Hoang (2023)	Enhanced Behavioral CAPTCHA for Mobile Applications	Penelitian eksperimental	CAPTCHA berbasis perilaku meningkatkan keamanan <i>login</i> tanpa mengurangi pengalaman pengguna (UX).
9.	Sultan, M., Putra, E. R., & Farjam, H. (2024)	Rancang Bangun Sistem Pelaporan Insiden Berbasis <i>Panic Button</i> Digital	Studi perancangan sistem	Fitur <i>Panic Button</i> digital memungkinkan pelaporan darurat lebih cepat dan terstruktur di lingkungan kerja berisiko tinggi.
10.	Ali, O. V., Siboro, I., & Ramdan, M. (2022)	Penerapan Formulir Pelaporan Digital untuk Sistem Keselamatan Kerja	Studi pengembangan	Formulir digital mempercepat pelaporan dan meningkatkan kelengkapan data pada sistem keselamatan kerja.
11.	Dewantara, R. A., Irwansyah, M. A., & Pratiwi, H. S. (2023)	Implementasi Sistem Pelaporan K3 Berbasis Mobile Application	Penelitian deskriptif	Aplikasi pelaporan K3 berbasis mobile meningkatkan kecepatan distribusi informasi dan efektivitas tindak lanjut.
12.	Yogama, C. D., Djunaidi, Z., & Rahmawati, F. F. (2023)	Implementasi Program Pelaporan Unsafe Action Menggunakan Aplikasi Mobile	Studi deskriptif	Desain UI yang intuitif membantu pengguna melakukan pelaporan dengan cepat tanpa hambatan teknis.
13.	Dwi Susanto, R. A. C. Wardana, D. A. Aditia, & W. Haryono (2023)	Penerapan Sistem Pelaporan Unsafe Action dan Unsafe Condition Berbasis Aplikasi Mobile	Studi implementatif	Aplikasi pelaporan mobile meningkatkan kecepatan respon terhadap potensi bahaya di area kerja industri.

14.	Zulkarnain, M. N., Dzakwan, R. H., & Arrahman, T. (2025)	Pengembangan Sistem Push Notification untuk Aplikasi Mobile Keselamatan Kerja	Studi eksperimental	Sistem notifikasi <i>Real-Time</i> meningkatkan efektivitas komunikasi keselamatan di lapangan.
15.	Mirza, M. N., Lourinx, E., Nugroho, R. A., & Syaifudin, A. (2024)	Design Optimization of <i>Real-Time</i> Notification Interface in Safety Applications	Studi UI/UX eksperimental	Desain pesan notifikasi yang jelas dan singkat meningkatkan keterlibatan serta pemahaman pengguna.
16.	Akbar, M. I., Fahlevi, M. R., & Muaafii, D. A. (2024)	Implementasi Ionic Framework pada Pengembangan Aplikasi Lintas Platform	Studi implementasi teknologi	Ionic Framework memungkinkan efisiensi pengembangan aplikasi lintas platform dengan satu basis kode.
17.	Ariesta, A., Wijaya, A. H., & Atmaja, M. A. (2023)	Optimization of Cross-Platform Apps Using Ionic and Capacitor Framework	Studi teknis	Integrasi Ionic dan Capacitor mempercepat proses pengembangan dan mempermudah akses fitur perangkat asli.
18.	Bidah & Noor Islami, R. (2024)	Efektivitas Framework Ionic dalam Pengembangan Aplikasi Mobile	Studi komparatif	Ionic terbukti efisien, ringan, dan kompatibel untuk pengembangan aplikasi lintas platform di industri.
19.	Pratama, R. E., & Wahyuni, E. G. (2024)	Evaluasi User Experience pada Aplikasi Mobile Berbasis Android	Penelitian UX kuantitatif	Kualitas UX berpengaruh signifikan terhadap kepuasan dan keterlibatan pengguna aplikasi mobile.
20.	Iqbal Sain, M., Rizkiawan, M. A., Rahmat, M. M. A., & Sidiq, M. (2025)	Penerapan Prinsip UI/UX Empatik dalam Pengembangan Aplikasi Mobile	Studi eksperimental	Desain antarmuka berorientasi empati pengguna meningkatkan efektivitas interaksi dan mengurangi kesalahan penggunaan.
21.	Rahmat Alamsyah, I. M. Nugroho, & S. Alam (2024)	Pengaruh Konsistensi Desain UI terhadap Kepercayaan Pengguna	Penelitian deskriptif kuantitatif	Konsistensi desain visual memperkuat kepercayaan pengguna terhadap aplikasi digital di sektor industri.
22.	Akwilla, H., & Jollyta, D. (2024)	Penerapan Agile Development pada Proyek Aplikasi Mobile	Studi implementatif	Pendekatan Agile meningkatkan fleksibilitas tim dan mempercepat siklus iterasi pengembangan.

23.	Suhari, A., Faqih, A., & Basysyar, F. M. (2023)	Penerapan Metode Scrum dalam Pengembangan Aplikasi Mobile Berbasis Agile	Studi studi kasus	Agile berbasis Scrum memfasilitasi kolaborasi antar tim dan menghasilkan fitur yang cepat diuji serta diperbarui.
-----	---	--	-------------------	---

2.10 Analisis Penelitian Terdahulu

Penelitian-penelitian yang dipaparkan pada Tabel 2. 1 Tinjauan Pustaka menunjukkan perkembangan signifikan dalam digitalisasi sistem *Health, Safety, and Environment (HSE)*, khususnya melalui penerapan teknologi *mobile* untuk meningkatkan efektivitas pelaporan dan komunikasi keselamatan kerja. Secara umum, seluruh penelitian tersebut dapat dikelompokkan ke dalam beberapa fokus utama yang relevan dengan pengembangan fitur pada aplikasi S2P-Mobile, yaitu digitalisasi HSE, penggunaan CAPTCHA sebagai metode autentikasi, pengembangan fitur Panic Button, penerapan notifikasi *Real-Time*, penggunaan Ionic Framework sebagai teknologi lintas platform, implementasi antarmuka UI/UX, serta metodologi *Agile Development* yang mendukung proses pengembangan aplikasi.

Fokus pertama berkaitan dengan digitalisasi sistem HSE. Sejumlah penelitian terdahulu seperti yang dilakukan oleh (Amir et al., 2025; Arno et al., 2014; Purnawati et al., 2023) menunjukkan bahwa penerapan aplikasi digital berbasis *mobile* dapat meningkatkan kecepatan pelaporan, memperbaiki akurasi data, serta memperkuat dokumentasi keselamatan kerja. Penelitian oleh (Askar et al., 2021) juga menegaskan bahwa sistem pelaporan berbasis web dapat mempercepat respon terhadap insiden dan mempermudah proses pemantauan keselamatan. Secara keseluruhan, kelompok penelitian ini memberikan landasan kuat bagi kebutuhan PT Sumber Segara Primadaya dalam melakukan transformasi digital melalui aplikasi S2P-Mobile.

Fokus kedua mengkaji teknologi CAPTCHA sebagai mekanisme keamanan untuk autentikasi pengguna. Penelitian (Adesina et al., 2023) menunjukkan bahwa CAPTCHA berbasis teks tetap menjadi pilihan efisien untuk perangkat *mobile*. (Andarini et al., 2023) memperkenalkan model hybrid CAPTCHA yang menggabungkan teks dan citra untuk meningkatkan keamanan. Penelitian lainnya oleh (Arifin & Yunus, 2025) dan (Dinh & Hoang, 2023) mengusulkan pendekatan CAPTCHA adaptif dan perilaku (*behavioral CAPTCHA*) sebagai metode autentikasi yang lebih aman namun tetap mempertahankan kenyamanan pengguna. Hasil penelitian ini menjadi dasar logis bagi pengembangan fitur Login dengan CAPTCHA pada aplikasi S2P-Mobile.

Fokus ketiga berkaitan dengan fitur Panic Button. Penelitian (Ali et al., 2024; Dewantara et al., 2022; Sultan, 2021) menunjukkan bahwa sistem pelaporan digital mampu mempercepat alur

informasi dalam situasi darurat dan meningkatkan akurasi data yang diterima. Sementara itu, (Yogama et al., 2022) menekankan pentingnya antarmuka pengguna yang intuitif untuk mempermudah pekerja dalam melakukan pelaporan di lapangan. Temuan-temuan ini mendukung perancangan fitur Panic Button pada aplikasi S2P-Mobile yang mengutamakan kecepatan dan kelengkapan laporan.

Fokus berikutnya mengangkat penggunaan notifikasi *Real-Time* sebagai media komunikasi keselamatan. (Susanto et al., 2024) menjelaskan bahwa digitalisasi notifikasi unsafe action dan unsafe condition mempercepat respon terhadap potensi bahaya. (Zulkarnain et al., 2025) menambahkan bahwa push notification berbasis *mobile* memastikan informasi keselamatan dapat diterima pekerja secara cepat meskipun dalam kondisi jaringan terbatas. Penelitian (Mirza et al., 2024) juga menyoroti bahwa desain pesan notifikasi yang ringkas dan jelas dapat meningkatkan pemahaman pengguna. Temuan ini menjadi landasan pengembangan fitur Notifikasi *Real-Time* pada aplikasi S2P-Mobile.

Dari sisi teknologi, penelitian oleh (Akbar et al., 2024; Ariesta et al., 2024; Bidah & Noor Islami, 2024) menunjukkan bahwa Ionic Framework merupakan solusi efektif untuk pengembangan aplikasi lintas platform karena efisiensinya, kompatibilitasnya, serta kemudahan integrasi dengan fitur perangkat *mobile* seperti GPS dan kamera. Relevansi penelitian ini mendukung penggunaan Ionic dalam pengembangan antarmuka S2P-Mobile.

Dalam konteks desain antarmuka dan pengalaman pengguna, penelitian (Alamsyah et al., 2022; Iqbal Sain et al., 2025; Pratama & Wahyuni, 2024) menegaskan bahwa kualitas UI/UX yang baik berpengaruh besar terhadap keterlibatan pengguna, kemudahan interaksi, serta tingkat kepercayaan pengguna terhadap aplikasi. Hasil penelitian ini merupakan pijakan penting dalam merancang UI/UX aplikasi S2P-Mobile yang intuitif dan mudah digunakan.

Terakhir, dari sisi metodologi pengembangan, penelitian oleh (Akwilla & Jollyta, 2024; Suhari et al., 2022) menunjukkan bahwa metode *Agile Development* memungkinkan proses pengembangan berlangsung lebih fleksibel, cepat, dan responsif terhadap perubahan kebutuhan pengguna. Dengan demikian, penerapan Agile dalam penelitian ini mendukung pengembangan fitur aplikasi S2P-Mobile secara bertahap dan terukur.

Secara keseluruhan, seluruh penelitian terdahulu memberikan dasar konseptual dan empiris terhadap pengembangan aplikasi S2P-Mobile, khususnya pada tiga fitur utama *Login* dengan CAPTCHA, *Panic Button*, dan Notifikasi *Real-Time*. Namun, belum ada penelitian sebelumnya yang mengintegrasikan ketiga fitur ini dalam satu aplikasi berbasis Ionic Framework untuk kebutuhan operasional industri berisiko tinggi seperti PLTU S2P. Oleh karena itu, penelitian ini

memberikan kontribusi baru melalui pengembangan antarmuka pengguna aplikasi keselamatan kerja yang adaptif, aman, dan efisien.

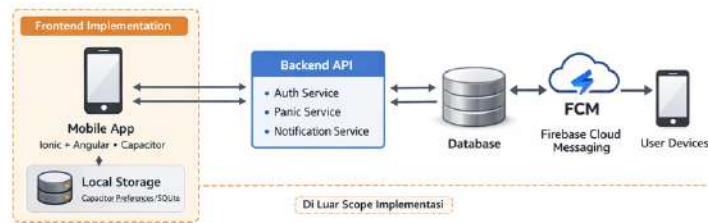
BAB III PELAKSANAAN MAGANG

3.1 Gambaran Umum Sistem S2P-Mobile

Aplikasi S2P-Mobile merupakan aplikasi *mobile* yang dikembangkan untuk mendukung digitalisasi sistem *Health, Safety, and Environment* (HSE) di lingkungan PT Sumber Segara Primadaya (S2P). Aplikasi ini memfasilitasi kebutuhan operasional HSE, terutama pada penyampaian informasi keselamatan, pelaporan kondisi/kejadian darurat, serta pemantauan informasi penting secara cepat melalui mekanisme notifikasi. Melalui pendekatan digital tersebut, alur komunikasi HSE diharapkan menjadi lebih efisien, terukur, dan responsif terhadap kondisi lapangan.

Secara umum, sistem S2P-Mobile terdiri dari tiga komponen utama. Pertama, aplikasi *mobile* S2P-Mobile yang berjalan pada perangkat Android dan iOS sebagai antarmuka pengguna (*frontend*). Komponen ini bertanggung jawab menampilkan halaman, menerima input pengguna, menampilkan umpan balik, serta mengelola sebagian data yang perlu tersedia pada perangkat, seperti status pesan “dibaca” dan riwayat notifikasi. Kedua, layanan *backend* (server dan API) yang menangani autentikasi pengguna, pemrosesan data laporan darurat, penyimpanan data sistem, serta penyediaan layanan API yang diakses oleh aplikasi *mobile*. Ketiga, layanan pengiriman notifikasi berbasis *Firebase Cloud Messaging* (FCM) yang berperan sebagai perantara distribusi notifikasi dari server ke perangkat pengguna secara *real-time*.

Pada sisi implementasi teknologi, aplikasi S2P-Mobile dibangun menggunakan Ionic Framework berbasis Angular sehingga mendukung pengembangan lintas platform dengan satu basis kode. Integrasi dengan fungsi perangkat, seperti lokasi (geolocation), penyimpanan lokal, dan notifikasi, dilakukan melalui Capacitor. Dalam operasionalnya, aplikasi berkomunikasi dengan backend melalui protokol HTTP menggunakan API. Dengan rancangan tersebut, sistem memastikan informasi HSE dapat dikirim dan diterima secara cepat, serta dapat tetap diakses ketika jaringan terbatas melalui penyimpanan lokal pada perangkat.



Gambar 3. 1 Arsitektur Sistem S2P-Mobile

Untuk memberikan gambaran yang lebih jelas mengenai hubungan antar komponen dalam sistem S2P-Mobile, diagram arsitektur sistem disajikan pada Gambar 3.1. Diagram ini menunjukkan keterkaitan antara aplikasi *mobile* sebagai *frontend*, layanan *backend* sebagai pengelola data dan logika sistem, serta layanan *Firestore Cloud Messaging* (FCM) sebagai mekanisme pengiriman notifikasi *real-time* ke perangkat pengguna.

3.1.1 Ruang Lingkup Tugas Akhir dalam Sistem

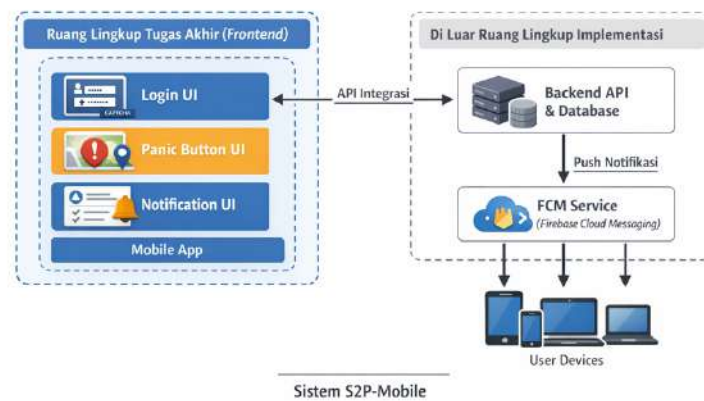
Pada Tugas Akhir jalur magang ini, ruang lingkup pekerjaan difokuskan pada pengembangan sisi *frontend* aplikasi S2P-Mobile sesuai peran sebagai *Frontend Mobile Developer* di PT Nocola IoT Solution. Fokus utama berada pada implementasi antarmuka pengguna (*user interface*) serta pengelolaan interaksi pengguna (*user experience*) pada sisi *frontend* aplikasi agar fitur yang dikembangkan dapat digunakan secara jelas, cepat, dan konsisten pada perangkat Android maupun iOS.

Bagian sistem yang dikembangkan pada Tugas Akhir ini mencakup tiga fitur inti, yaitu:

1. Autentikasi pengguna melalui halaman *login* yang dilengkapi CAPTCHA.
2. Fitur Panic Button sebagai formulir pelaporan darurat yang dapat mengikutsertakan informasi lokasi perangkat
3. Halaman notifikasi *real-time* yang mendukung status baca serta menyediakan akses informasi ketika jaringan terbatas melalui penyimpanan lokal.

Ketiga fitur tersebut dipilih karena berperan langsung pada kebutuhan operasional HSE, yaitu penguatan keamanan akses, percepatan pelaporan keadaan darurat, serta efektivitas penyebaran informasi keselamatan.

Adapun proses pada sisi *backend* seperti validasi kredensial pengguna, penerbitan token autentikasi, pemrosesan dan penyimpanan data laporan, manajemen basis data, serta pengiriman notifikasi melalui FCM, tidak termasuk dalam ruang lingkup implementasi pada Tugas Akhir ini. Meskipun demikian, alur interaksi antara aplikasi *mobile* dan komponen lainnya tetap dijelaskan untuk menunjukkan keterhubungan proses bisnis secara menyeluruh dan memastikan antarmuka yang dibangun dapat berintegrasi dengan layanan API yang tersedia.



Gambar 3. 2 Ruang Lingkup Tugas Akhir

Untuk memperjelas batasan pekerjaan pada Tugas Akhir jalur magang ini, ditampilkan diagram ruang lingkup sistem pada Gambar 3.2. Diagram tersebut menunjukkan bagian sistem S2P-Mobile yang menjadi fokus pengembangan penulis, yaitu sisi *frontend* aplikasi pada fitur Login dengan CAPTCHA, fitur Panic Button, dan Notifikasi *Real-Time*, serta keterkaitannya dengan layanan *backend* yang berada di luar ruang lingkup implementasi.

Tabel 3. 1 Ringkasan Desain Sistem & Ruang Lingkup TA

Komponen Sistem	Deskripsi Singkat	Ruang Lingkup TA?	Interaksi dengan Bagian Lain
<i>Frontend</i> (Ionic Framework +	- <i>Login</i> dengan CAPTCHA (validasi lokal)	Ya	- API call <i>Backend</i> (<i>login/post</i> laporan)

Angular + Capacitor)	- <i>Panic Button</i> (form + GPS auto) - Notifikasi (list, read status, offline storage)		- FCM receive + local storage
<i>Backend</i> (API Server + Database)	Auth server, simpan data HSE	Tidak	Terima API request dari <i>Frontend</i> Anda
FCM (Firebase Cloud Messaging)	Push <i>real-time</i> dari server	Tidak	<i>Frontend</i> Anda tampilkan setelah receive

3.2 Proses Bisnis Aplikasi S2P-Mobile

Proses bisnis aplikasi S2P-Mobile dirancang untuk mendukung alur kerja HSE secara digital mulai dari autentikasi pengguna hingga penyampaian informasi keselamatan melalui *notifikasi*. Alur utama sistem melibatkan interaksi antara pengguna, aplikasi *mobile*, layanan API pada *backend*, serta layanan notifikasi berbasis FCM. Dalam implementasinya, aplikasi memastikan proses penting seperti *login*, pelaporan darurat, dan *notifikasi* berjalan dengan terstruktur, memberikan umpan balik melalui mekanisme *frontend* seperti validasi form, indikator loading, dan pesan status. serta tetap mempertimbangkan kondisi penggunaan di lapangan yang dapat mengalami keterbatasan jaringan.

Secara garis besar, proses dimulai ketika pengguna melakukan autentikasi untuk memperoleh akses ke aplikasi. Setelah autentikasi berhasil, pengguna dapat menggunakan fitur *Panic Button* untuk mengirim laporan darurat beserta lokasi (jika diizinkan). Pada saat yang sama, sistem mendukung mekanisme notifikasi agar informasi keselamatan dan peringatan penting dapat diterima pengguna secara *real-time*. Subbab berikut menjelaskan proses bisnis per fitur secara lebih rinci.

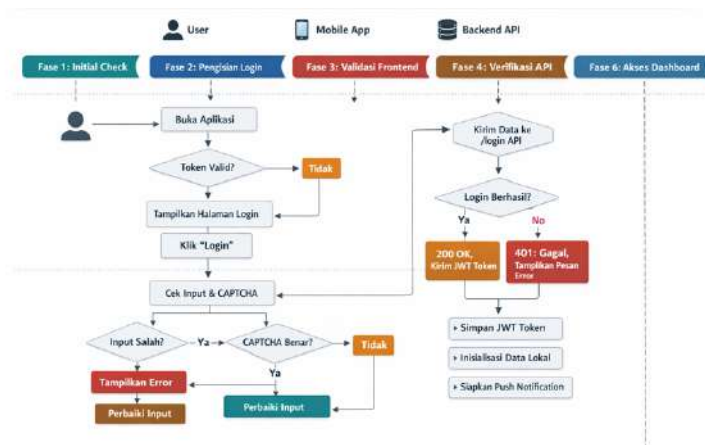
3.2.1 Proses Otentikasi Pengguna

Autentikasi pengguna merupakan gerbang utama akses ke sistem S2P-Mobile. Pengguna diwajibkan mengisi *username* dan *password* yang valid, serta melakukan verifikasi CAPTCHA sebagai lapisan tambahan untuk mengurangi risiko akses oleh sistem otomatis (bot). Pada saat halaman *login* ditampilkan, aplikasi menghasilkan CAPTCHA secara acak dan menampilkannya pada antarmuka. Pengguna kemudian memasukkan CAPTCHA sesuai tampilan sebelum proses *login* dikirimkan ke *backend*.

Setelah pengguna menekan tombol *login*, aplikasi melakukan validasi awal di sisi *frontend* untuk memastikan isian tidak kosong dan CAPTCHA sesuai. Apabila CAPTCHA tidak sesuai, aplikasi menampilkan pesan kesalahan, melakukan penyegaran CAPTCHA, dan meminta

pengguna mengulangi input. Jika validasi awal terpenuhi, aplikasi mengirim permintaan autentikasi ke API pada *backend*. Ketika autentikasi berhasil, aplikasi menerima token serta informasi pengguna yang diperlukan, kemudian menyimpannya pada penyimpanan lokal perangkat menggunakan Capacitor Preferences agar sesi pengguna dapat dipertahankan. Setelah itu, aplikasi melakukan inisialisasi kebutuhan pasca *login* (misalnya konfigurasi notifikasi pada perangkat sesuai kebijakan aplikasi) dan mengarahkan pengguna ke halaman utama.

Jika autentikasi gagal (misalnya kredensial tidak valid atau koneksi bermasalah), aplikasi menampilkan umpan balik yang sesuai dan mengembalikan pengguna ke kondisi form *login*. Mekanisme ini menjaga keamanan akses, sekaligus mempertahankan pengalaman pengguna melalui validasi bertahap dan umpan balik yang terkontrol.



Gambar 3. 3 Proses Autentikasi Pengguna

Untuk memperjelas alur proses autentikasi pengguna pada aplikasi S2P-Mobile, ditampilkan Diagram Proses Autentikasi Pengguna pada Gambar 3.3. Diagram tersebut menggambarkan tahapan *login* mulai dari input kredensial dan CAPTCHA oleh pengguna, proses validasi di sisi *frontend*, pengiriman data ke *backend* melalui API, hingga penyimpanan data autentikasi pada perangkat dan pengalihan pengguna ke halaman utama aplikasi.

3.2.2 Proses Pelaporan Panic Button

Fitur Panic Button digunakan untuk mendukung pelaporan keadaan darurat secara cepat dan terstruktur di area operasional. Ketika pengguna membuka halaman panic button, aplikasi menampilkan formulir pelaporan yang memuat informasi penting seperti jenis kejadian, unit, area/lokasi kerja, serta keterangan tambahan bila diperlukan. Untuk meningkatkan akurasi

pelaporan, aplikasi dapat memanfaatkan fitur lokasi perangkat (geolocation) sehingga koordinat dapat diperoleh secara otomatis sesuai izin yang diberikan pengguna.

Saat pengguna mengirim laporan, aplikasi melakukan validasi formulir di sisi *frontend* untuk memastikan *field* wajib telah terisi. Jika data belum lengkap, sistem menampilkan pesan kesalahan pada komponen yang relevan agar pengguna dapat segera memperbaiki input. Jika validasi terpenuhi, aplikasi membentuk payload laporan yang mencakup informasi kejadian dan koordinat lokasi (jika fitur lokasi diaktifkan), kemudian mengirimkannya ke API pada *backend*. Pada sisi pengguna, aplikasi menampilkan indikator proses (misalnya loading) sebagai umpan balik bahwa laporan sedang diproses.

Setelah *backend* mengembalikan respons berhasil, aplikasi menampilkan konfirmasi bahwa laporan darurat telah terkirim. Dalam konteks integrasi sistem, laporan panic button yang diterima *backend* dapat menjadi pemicu distribusi informasi kepada pihak terkait melalui mekanisme notifikasi. Dengan demikian, fitur Panic Button berperan sebagai pemicu utama alur respons darurat dalam sistem HSE berbasis digital.



Gambar 3. 4 Proses Pelaporan Panik Button

Untuk memperjelas alur proses pelaporan darurat melalui fitur Panic Button pada aplikasi S2P-Mobile, ditampilkan Diagram Proses Pelaporan Panic Button pada Gambar 3.4. Diagram tersebut menggambarkan tahapan pelaporan mulai dari pengisian formulir oleh pengguna, validasi data di sisi *frontend*, pengambilan lokasi perangkat, pengiriman data laporan ke *backend* melalui API, hingga penerimaan respons dan pemicu distribusi notifikasi kepada pihak terkait.

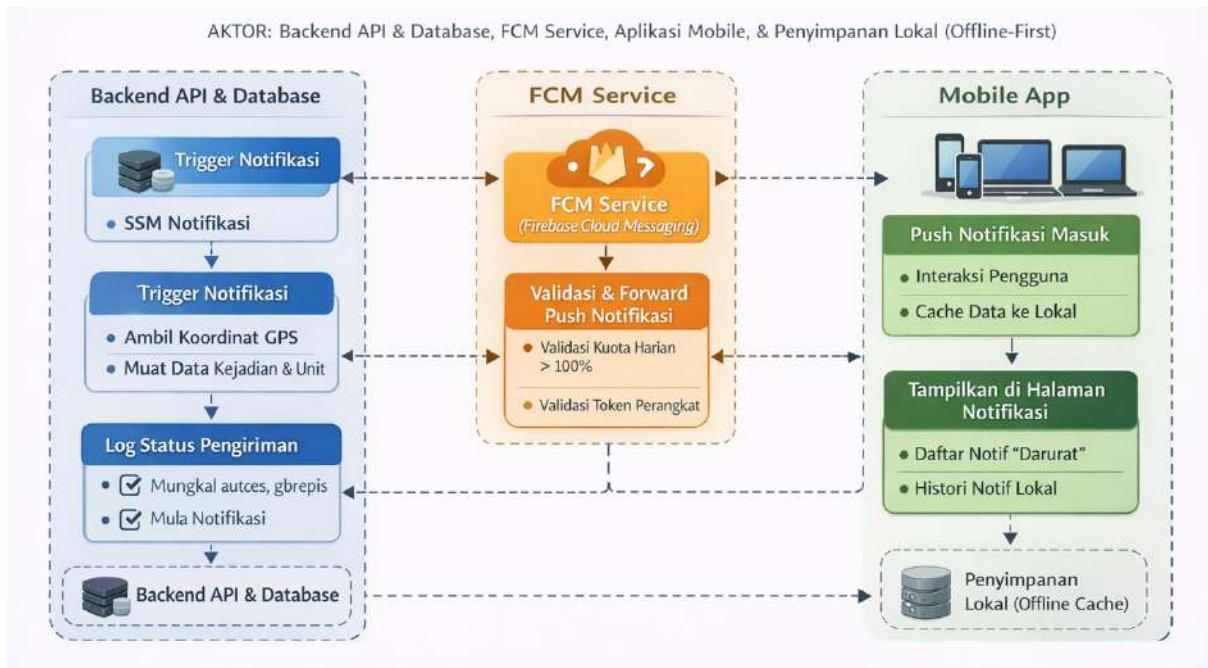
3.2.3 Proses Distribusi Notifikasi *Real-Time*

Fitur Notifikasi *Real-Time* berfungsi untuk mendistribusikan informasi keselamatan, peringatan kerja, dan pengumuman penting kepada pengguna aplikasi. Dalam alur sistem, notifikasi berasal dari *backend* dan dikirim ke perangkat pengguna melalui layanan FCM. Pada sisi aplikasi *mobile*, terdapat dua aspek utama, yaitu:

1. mekanisme penerimaan notifikasi pada perangkat.
2. penyajian serta pengelolaan daftar notifikasi pada halaman notifikasi di dalam aplikasi.

Pada tahap penerimaan notifikasi, perangkat pengguna perlu memberikan izin notifikasi sesuai kebijakan sistem operasi. Setelah izin diberikan, perangkat memperoleh identitas penerima berupa token perangkat (FCM token). Token ini digunakan agar *backend* dapat mengirim notifikasi ke perangkat yang tepat. Ketika notifikasi dikirim oleh *backend*, notifikasi dapat muncul sebagai notifikasi sistem (saat aplikasi berada di latar belakang) atau ditangani oleh aplikasi (saat aplikasi aktif). Informasi notifikasi kemudian dicatat agar dapat ditampilkan pada halaman notifikasi.

Pada halaman notifikasi, aplikasi menampilkan daftar pesan yang dikelola secara dinamis melalui *state frontend*, dilengkapi informasi waktu dan indikator status baca. Pengguna dapat membuka detail pesan serta menandai pesan sebagai telah dibaca. Untuk mendukung akses informasi ketika jaringan tidak stabil, aplikasi menyimpan data notifikasi pada penyimpanan lokal perangkat sehingga pesan yang telah diterima tetap dapat dilihat kembali saat perangkat berada dalam kondisi *offline*. Dengan demikian, fitur notifikasi tidak hanya berfungsi sebagai komunikasi cepat, tetapi juga menyediakan histori informasi keselamatan yang dapat diakses ulang sesuai kebutuhan pengguna di lapangan.



Gambar 3. 5 Proses Distribusi Notifikasi *Real-Time*

Untuk memperjelas alur distribusi notifikasi pada aplikasi S2P-Mobile, ditampilkan Diagram Proses Distribusi Notifikasi *Real-Time* pada Gambar 3.5. Diagram tersebut menggambarkan proses mulai dari pengiriman notifikasi oleh *backend* melalui layanan *Firebase Cloud Messaging* (FCM), penerimaan notifikasi pada perangkat pengguna, hingga penyimpanan dan penampilan pesan pada halaman notifikasi aplikasi, termasuk dukungan akses data secara *offline*.

3.2.4 Ringkasan Interaksi Antar Komponen

Untuk menegaskan keterkaitan antar komponen dalam sistem, interaksi utama S2P-Mobile dapat dirangkum sebagai berikut. Autentikasi pengguna terjadi melalui pertukaran data antara aplikasi *mobile* dan API pada *backend* untuk memverifikasi kredensial, kemudian data sesi disimpan secara lokal agar aplikasi dapat mempertahankan kondisi *login*. Pelaporan darurat melalui panic button dilakukan melalui pengiriman data formulir dan koordinat lokasi dari aplikasi ke *backend* melalui API. Notifikasi *Real-Time* didistribusikan dari *backend* ke perangkat melalui FCM, lalu disajikan dan dikelola pada halaman notifikasi dalam aplikasi, termasuk penyimpanan lokal untuk mendukung akses saat jaringan terbatas. Ringkasan ini menunjukkan bahwa fitur yang dikembangkan pada sisi antarmuka tetap terhubung dengan proses bisnis sistem secara menyeluruh melalui mekanisme integrasi API dan layanan notifikasi.

3.3 Manajemen Proyek

Program magang ini dilaksanakan di PT Nocola IoT Solution, sebuah perusahaan yang bergerak di bidang pengembangan perangkat lunak dan solusi *Internet of Things* (IoT). Kegiatan magang berlangsung selama enam bulan, mulai dari 15 September 2024 hingga 16 Maret 2025, di bawah bimbingan Bapak Hamzah Alvana selaku Supervisor Mobile Developer. Selama periode magang, peran *Mobile Developer* dijalankan dengan fokus pada pengembangan antarmuka pengguna (*User Interface*) aplikasi S2P-Mobile, yaitu aplikasi yang dirancang untuk mendukung digitalisasi sistem *Health, Safety, and Environment* (HSE) di PT Sumber Segara Primadaya (S2P).

Dalam implementasi proyek ini, pengembangan difokuskan pada antarmuka pengguna (*User Interface*) dan pengalaman pengguna (*User Experience*) untuk tiga fitur utama aplikasi, yaitu *Login* dengan CAPTCHA, *Panic Button*, dan *Notifikasi Real-Time*. Ketiga fitur ini dikembangkan menggunakan Ionic Framework yang didukung oleh *Angular* sebagai kerangka kerja utama, serta *Capacitor* untuk menghubungkan ke fungsi perangkat seperti GPS, penyimpanan lokal, dan *Notifikasi Real-Time*. Pengembangan dilakukan dengan mengutamakan prinsip desain adaptif, sehingga aplikasi menawarkan tampilan yang seragam, responsif, dan dapat diakses di berbagai perangkat Android dan iOS.

Serangkaian kegiatan magang dilaksanakan secara bertahap, dimulai dengan pengenalan lingkungan kerja dan mekanisme operasional perusahaan, dilanjutkan dengan instalasi kerangka kerja dan pelatihan, dan terakhir implementasi proyek serta pengujian fitur aplikasi. Melalui proses ini, diperoleh pengalaman praktis dalam menerapkan teknologi Ionic Framework dalam konteks industri, sekaligus memahami penerapan metode Agile Development dalam manajemen proyek di PT Nocola IoT Solution. Deskripsi lengkap mengenai kegiatan magang selama periode ini dapat dilihat pada Tabel 3.1 di bawah ini.

Tabel 3. 2 Aktivitas Magang

No	Deskripsi Kegiatan	Waktu Kegiatan	Durasi
1.	Perkenalan Lingkungan Perusahaan (<i>welcoming</i>)	17 September 2024	1 Hari
2.	Pengarahan dari Supervisor dan Instalasi Ionic Framework	18 September 2024	1 Hari
3.	<i>Training</i> Ionic Framework	19 September 2024 – 24 Oktober 2024	36 Hari
4.	Membuat Aplikasi Pokedex	28 September 2024 – 24 Oktober 2024	7 Hari
5.	Membuat Aplikasi CRUD	5 Oktober 2024 – 12 Oktober 2024	7 Hari

6.	Ikut serta dalam Proyek Pengembangan Aplikasi S2P-Mobile	25 Oktober 2024 – 4 Januari 2025	72 Hari
----	--	----------------------------------	---------

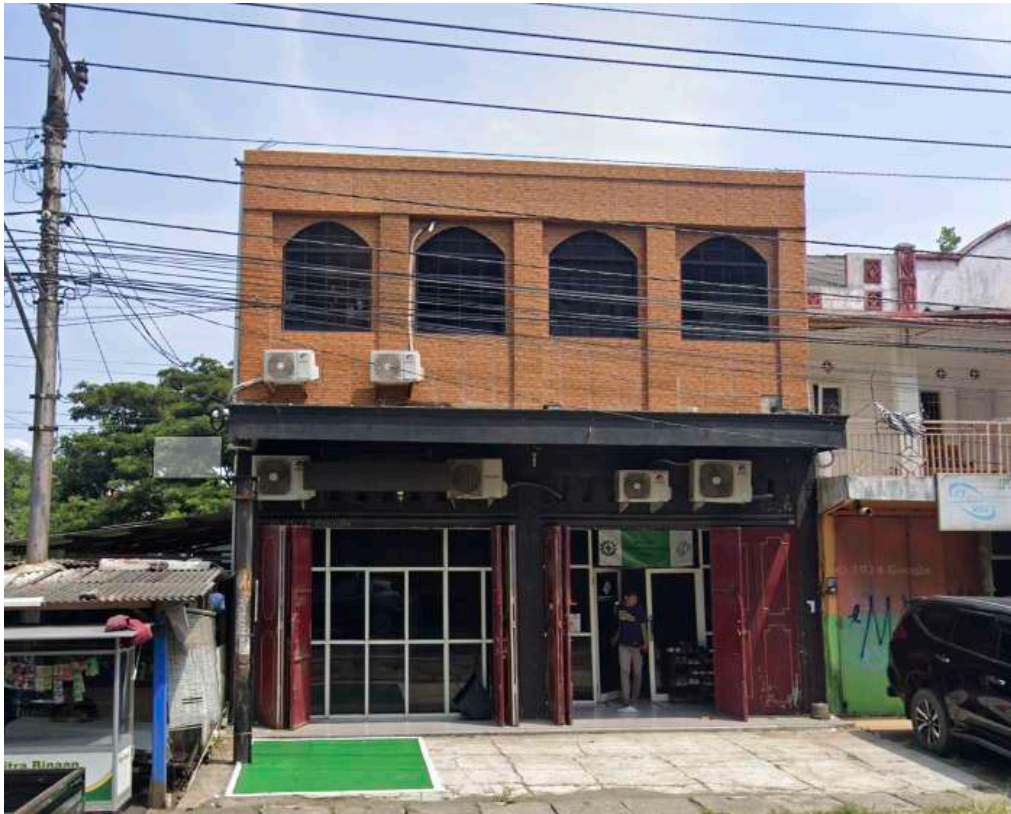
Setelah rangkaian kegiatan orientasi dan pelatihan awal selesai, aktivitas teknis dalam proyek S2P-Mobile tidak lagi disusun berdasarkan urutan harian, tetapi mengikuti pembagian *sprint project* sesuai metode Agile Development yang digunakan di PT Nocola IoT Solution. Setiap *sprint* difokuskan pada satu kelompok fitur utama, yaitu autentikasi pengguna melalui *Login* dengan CAPTCHA (*Sprint 1*), fitur pelaporan darurat *Panic Button* (*Sprint 2*), dan sistem Notifikasi *Real-Time* (*Sprint 3*). Rincian pelaksanaan masing-masing *sprint* dijelaskan lebih lanjut pada subbab 3.1.4 dan Tabel 3.3

3.3.1 Perkenalan Lingkungan Perusahaan (*welcoming*)

PT Nocola *IoT Solution* adalah perusahaan teknologi Indonesia yang berspesialisasi dalam pengembangan solusi *Internet of Things (IoT)*, perangkat lunak, dan sistem komunikasi terintegrasi. Didirikan pada tahun 2016 di Cilacap, Jawa Tengah, Nocola berkomitmen untuk menyediakan inovasi digital yang relevan dengan kebutuhan industri modern. Selama perkembangannya, perusahaan ini telah tumbuh menjadi salah satu pemain utama di sektor teknologi Indonesia, dengan peran penting dalam integrasi sistem, pengembangan aplikasi *mobile*, dan layanan *IoT* yang berfokus pada data dan otomatisasi.

Sebagai entitas yang mengutamakan penelitian dan inovasi teknologi, Nocola didukung oleh tim ahli dari berbagai bidang, termasuk *hardware, software, embedded sistem, dan multi-platform development*. Perusahaan ini juga menjalin kemitraan strategis dengan mitra teknologi domestik dan internasional untuk meningkatkan kualitas dan daya saing produknya. Layanan utama Nocola meliputi integrasi sistem, pengembangan IoT, solusi komunikasi jaringan, dan perangkat digital kustom untuk industri energi, manufaktur, dan pemerintahan.

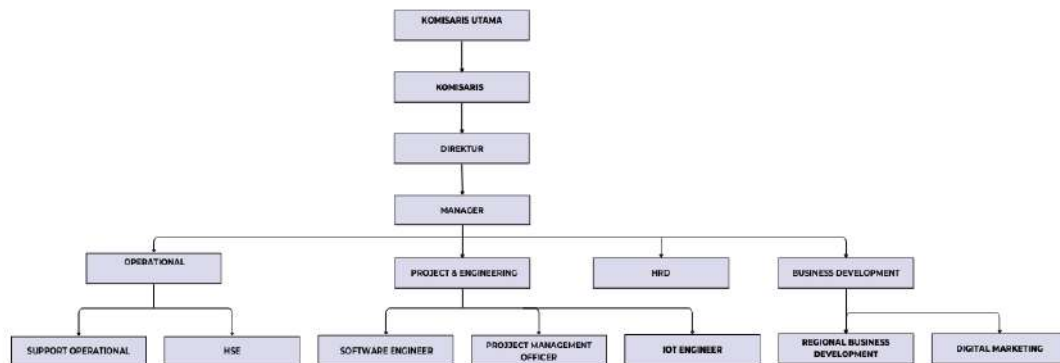
Dari segi geografis, kantor pusat Nocola berlokasi di Jl. Ir. Djuanda No. 117, Sidanegara, Cilacap, Jawa Tengah, sementara kantor workshop utama berlokasi di Jl. Lengkong No. 160, Mertasinga, Cilacap Utara, Cilacap, Jawa Tengah. Kantor pusat digunakan untuk manajemen, koordinasi proyek, dan administrasi perusahaan. Di sisi lain, sementara itu workshop digunakan sebagai pusat penelitian, pengujian perangkat, dan pengembangan teknologi IoT yang melibatkan kolaborasi antara tim teknis dari bidang perangkat lunak dan perangkat keras. Pada bagian ini ditampilkan foto profil perusahaan sebagai ilustrasi lingkungan kerja tempat pelaksanaan magang.



Gambar 3. 6 Foto Profil PT Nocola IoT Solution

Selain itu, untuk memberikan gambaran lebih jelas mengenai struktur organisasi perusahaan, ditampilkan pula bagan struktur organisasi PT Nocola IoT Solution, yang menunjukkan posisi divisi-divisi utama seperti HRD, *Engineering*, *Design*, serta *Mobile Developer* sebagai divisi tempat pelaksanaan magang dilakukan.

STRUKTUR ORGANISASI PT NOCOLA IOT SOLUTION



Gambar 3. 7 Struktur Organisasi PT Nocola IoT Solution

Selama fase awal magang, peserta magang diperkenalkan dengan lingkungan kerja, fasilitas kantor, dan staf di PT Nocola *IoT* Solution oleh Bapak Hamzah Alvana, yang menjabat sebagai Supervisor Pengembang Mobile dan mentor lapangan. Selain diperkenalkan kepada tim dan struktur organisasi, kegiatan orientasi juga mencakup pembimbingan oleh Bagian *Human Resource Development* (HRD) mengenai norma budaya kerja, peraturan internal, dan mekanisme administratif perusahaan. Untuk menjaga profesionalisme, semua magang diwajibkan menandatangani Perjanjian Kerahasiaan (NDA) sebagai bentuk kesepakatan untuk melindungi kerahasiaan data, proyek, dan semua informasi internal perusahaan.

Selama orientasi, gambaran umum mengenai aktivitas perusahaan diperoleh melalui pengarahan awal yang berhubungan dengan proyek pengembangan aplikasi *mobile* untuk berbagai klien. Salah satu proyek yang sedang dikembangkan saat itu adalah S2P-Mobile, yaitu aplikasi yang ditujukan untuk mendukung digitalisasi sistem *Health, Safety, and Environment* (HSE) di lingkungan PT Sumber Segara Primadaya (S2P) salah satu mitra kerja Nocola. Melalui pengarahan langsung dari supervisor, diperoleh pemahaman mengenai latar belakang, tujuan, serta ruang lingkup proyek. Berdasarkan pembagian tugas, pengembangan antarmuka (Frontend) difokuskan pada tiga fitur utama, yaitu Login dengan CAPTCHA, Panic Button, dan Notifikasi *Real-Time*. Ketiga fitur ini dipilih karena memiliki peran penting dalam mendukung keamanan, kecepatan pelaporan, serta penyebaran informasi keselamatan di lingkungan kerja industri energi.

Dengan adanya tahapan pengenalan ini, kegiatan magang memberikan pemahaman menyeluruh mengenai struktur organisasi, budaya kerja, dan arah pengembangan teknologi di Nocola. Melalui tahapan pengenalan ini, diperoleh pemahaman mengenai bagaimana sinergi antara perusahaan dan klien dibangun dalam bentuk kolaborasi proyek nyata. Orientasi awal ini menjadi pondasi penting untuk menjalankan tanggung jawab magang secara profesional sekaligus memastikan bahwa hasil kerja yang dihasilkan selaras dengan standar dan kebutuhan perusahaan.

3.3.2 Pengarahan dari Supervisor dan *Instalasi Ionic Framework*

Pada awal periode magang, Bapak Hamzah Alvana, yang bertindak sebagai *Supervisor* Pengembang Mobile dan mentor lapangan selama magang, memperkenalkan kerangka kerja dan teknologi inti yang digunakan oleh perusahaan. Teknologi utama yang dipelajari adalah Ionic Framework, yang menjadi landasan utama dalam pengembangan berbagai proyek aplikasi mobile di PT Nocola *IoT* Solution. Orientasi awal ini dirancang untuk mendukung penguasaan arsitektur operasional Ionic serta penyesuaian terhadap standar teknologi yang diterapkan oleh perusahaan. Sebagai sumber belajar, *supervisor* menyediakan modul internal dalam bentuk dokumen PDF, sambil mendorong penulis untuk mempelajari dokumentasi resmi Ionic yang dapat diakses melalui situs web <https://ionicframework.com/docs>.

Langkah awal yang dilakukan adalah menyiapkan lingkungan pengembangan agar sesuai dengan persyaratan proyek. Sebelum melakukan instalasi Ionic, dilakukan pemeriksaan terhadap ketersediaan elemen pendukung sistem, seperti Node.js, Node Package Manager (npm), dan Git. Node.js berfungsi sebagai lingkungan runtime untuk menjalankan kode JavaScript, sementara npm digunakan sebagai manajer paket untuk mengelola dependensi. Git, di sisi lain, digunakan untuk mengontrol versi kode sumber. Pemeriksaan versi dilakukan melalui terminal untuk memastikan bahwa semua dependensi terinstal dengan benar pada perangkat pengembangan yang digunakan selama pelaksanaan magang.

Setelah semua persyaratan sistem terpenuhi, proses instalasi Ionic CLI (*Command Line Interface*) melalui terminal menggunakan npm sebagai manajer paket. Ionic CLI berfungsi sebagai alat utama untuk membuat, mengonfigurasi, dan menjalankan proyek. Setelah proses instalasi selesai, dalam penelitian ini dilakukan pemeriksaan menggunakan perintah pemeriksaan versi untuk memastikan bahwa Ionic CLI telah diinstal dengan benar dan siap digunakan dalam pengembangan. Pengawas juga memastikan bahwa pengaturan versi ini sesuai dengan standar yang diterapkan oleh tim pengembangan *mobile* di perusahaan.

Langkah berikutnya melibatkan pembuatan proyek baru menggunakan perintah standar Ionic. Framework ini menyediakan beberapa templat awal, seperti *blank*, *tabs*, dan *sidemenu*, masing-masing dengan struktur halaman *default* yang berbeda. Templat *blank* dipilih karena memberikan fleksibilitas yang lebih besar dalam merancang antarmuka yang disesuaikan dengan kebutuhan proyek. Setelah proyek dibuat, direktori proyek diakses melalui terminal untuk menjalankan proses build awal dan memastikan sistem berjalan tanpa kesalahan.

Proses instalasi selesai dengan penambahan platform uji menggunakan *Capacitor*, sebuah perpustakaan yang memudahkan integrasi aplikasi dengan fungsi perangkat asli, seperti kamera, notifikasi, dan penyimpanan lokal. Dukungan untuk platform Android dan iOS ditambahkan melalui *Capacitor* agar aplikasi dapat diuji pada kedua sistem operasi tersebut. Sebelum melakukan pengujian pada perangkat fisik, penulis juga memastikan bahwa Android Studio dan Xcode telah terinstal dengan benar pada alat pengembangan. Dengan demikian, penyelesaian instalasi dan konfigurasi ini menandai kesiapan lingkungan kerja untuk fase berikutnya, yaitu pelatihan dan pengembangan aplikasi di bawah bimbingan supervisor proyek.

3.3.3 Training Ionic Framework

Setelah menyelesaikan tahap instalasi pengenalan teknologi, proses dilanjutkan ke fase pelatihan implementasi Ionic Framework. Fase ini bertujuan untuk memberikan pemahaman mendalam tentang konsep, struktur, dan implementasi Ionic Framework dalam proyek aplikasi *mobile* di PT Nocola IoT Solution. Selama pelatihan, bimbingan langsung oleh Bapak Hamzah Alvana sebagai Supervisor Pengembang Mobile. Pendekatan pelatihan ini menekankan pada pelatihan praktik langsung melalui penerapan teori ke dalam proyek latihan, yaitu Aplikasi Pokedex dan Aplikasi Katalog Produk. Kedua proyek ini berfungsi sebagai sarana untuk menguasai alur kerja Ionic, *Angular*, dan integrasi dengan API.

Pelatihan dimulai dengan pengenalan arsitektur proyek Ionic yang berbasis *Angular* Framework. Pemahaman terhadap cara kerja Ionic diperoleh melalui sistem routing, arsitektur berbasis komponen, dan dukungan *Capacitor* untuk mengakses fitur perangkat seperti kamera, GPS, dan penyimpanan lokal. Supervisor menjelaskan struktur folder proyek, yang meliputi direktori *src* untuk komponen dan halaman antarmuka, direktori *app* untuk pengelolaan rute, dan direktori *assets* untuk menyimpan gambar dan ikon. Wawasan ini menjadi dasar penting dalam pengembangan struktur proyek yang rapi, modular, dan efisien.

Selanjutnya, pelatihan dilanjutkan dengan pembuatan dua proyek latihan yang memiliki fungsi berbeda, yaitu Aplikasi Pokedex dan Aplikasi Katalog Produk. Aplikasi Pokedex digunakan

untuk memahami integrasi data eksternal melalui API publik, sementara Aplikasi Katalog Produk berfokus pada pengelolaan data dari API internal dan interaksi pengguna. Melalui dua latihan ini, berbagai komponen Ionic seperti `ion-card`, `ion-searchbar`, `ion-list`, `ion-refresher`, `ion-fab`, serta penerapan servis dalam *Angular*.

a. Aplikasi Pokedex

Aplikasi Pokedex dikembangkan sebagai latihan untuk menampilkan daftar Pokemon secara dinamis menggunakan API publik. Data diambil dari PokeAPI, kemudian ditampilkan dalam daftar menggunakan `ion-card` dan `ion-searchbar`. Setiap kartu menampilkan detail seperti nama, jenis, dan gambar Pokemon, sementara pengguna dapat mencari Pokemon tertentu menggunakan bidang pencarian. Potongan kode berikut menggambarkan struktur antarmuka halaman utama:

```
<ion-searchbar [(ngModel)]="searchTerm"
  (ionInput)="filterPokemon()"></ion-searchbar>

<ion-card *ngFor="let poke of filteredPokemons; let i = index"
  [color]="i % 3 === 0 ? 'warning' : i % 3 === 1 ? 'primary' :
  'success'"
  (click)="goToDetails(poke.id)">
  <ion-card-content>
    <ion-row>
      <ion-col size="8">
        <ion-card-title>{{ poke.name | titlecase }}</ion-card-title>
        <ion-card-subtitle>
          <ion-badge *ngFor="let t of poke.types" color="medium">{{ t |
titlecase }}</ion-badge>
        </ion-card-subtitle>
      </ion-col>
      <ion-col size="4">
        <ion-img [src]="poke.image"></ion-img>
      </ion-col>
    </ion-row>
  </ion-card-content>
</ion-card>
```

Gambar 3. 8 Antarmuka Aplikasi Pokedex (UI).

Snippet kode ini menampilkan daftar Pokemon dalam grid adaptif. Komponen `ion-searchbar` digunakan untuk menyaring data secara langsung berdasarkan nama Pokemon, sementara `ion-card` digunakan untuk menampilkan informasi yang diperoleh dari API. Warna pada setiap kartu diatur secara bergantian melalui kondisi logika (`i % 3`) untuk menghasilkan tampilan visual yang menarik dan dinamis.

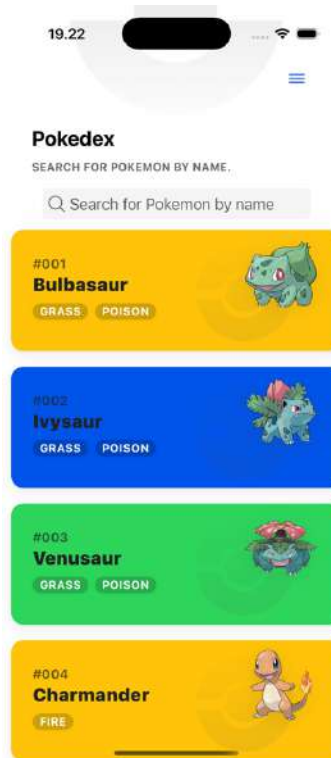
Proses pengambilan data dari API dilakukan dengan Observable dan Promise, sebagaimana ditunjukkan dalam cuplikan kode berikut:

```
this.pokemonService.getPokemons().subscribe((res) => {
  const requests = res.results.map((poke: any) =>
this.pokemonService.getPokemonDetail(poke.url));
  Promise.all(requests).then((details) => {
    this.pokemons = details.map((d) => ({
      id: d.id,
      name: d.name,
      types: d.types.map((t: any) => t.type.name),
      image: d.sprites.other['official-artwork'].front_default,
    }));
    this.filteredPokemons = this.pokemons;
  });
});
```

Gambar 3. 9 Kode Pengambilan Data dari API.

Kode ini menjelaskan proses permintaan asinkron untuk mengambil data Pokemon dari API. Fungsi `getPokemons()` menghasilkan daftar URL, yang kemudian diproses melalui `Promise.all()` sehingga data Pokemon yang detail dapat dimuat secara paralel dan lebih efisien. Setelah data diperoleh, hasilnya disimpan dalam variabel `pokemons` dan `filteredPokemons` untuk ditampilkan kepada pengguna.

Hasil akhir dari latihan aplikasi ini ditampilkan pada Gambar 3.10, di mana daftar Pokemon ditampilkan secara lengkap, termasuk nama, jenis, dan gambar karakter. Fitur pencarian berfungsi optimal untuk menampilkan hasil yang relevan berdasarkan kata kunci yang dimasukkan oleh pengguna.



Gambar 3. 10 Hasil Tampilan Daftar Pokémon di Aplikasi.

b. Aplikasi Katalog Produk

Aplikasi Katalog Produk dikembangkan untuk mengasah keterampilan penulis dalam menampilkan, mengelola, dan memperbarui data dari API internal perusahaan. Aplikasi ini menampilkan daftar produk melalui komponen ion-card, yang mencakup detail seperti nama, harga, dan deskripsi produk. Fitur tambahan seperti pull-to-refresh, pesan toast, dan tombol aksi mengambang juga diimplementasikan untuk memperkaya pengalaman interaksi pengguna.

Berikut adalah potongan kode untuk menampilkan daftar produk di halaman utama:

```
<ion-card *ngFor="let product of products"
  [routerLink]="['/product-detail', product._id]"
  routerDirection="forward">
  <img *ngIf="product.imagePath"
    [src]="'http://localhost:8080/uploads/' + product.imagePath"
    alt="{{ product.name }}" class="product-image" />
  <ion-card-header>
    <ion-card-title class="ion-text-capitalize">{{ product.name }}</ion-
card-title>
    <ion-card-subtitle>{{ product.price | currency }}</ion-card-subtitle>
  </ion-card-header>
  <ion-card-content>
    <p>{{ product.description }}</p>
  </ion-card-content>
</ion-card>
```

Gambar 3. 11 Kode Antarmuka Daftar Produk.

Setiap elemen ion-card menampilkan data produk secara dinamis dari server lokal. Atribut routerLink digunakan untuk mengarahkan pengguna ke halaman detail produk saat salah satu kartu dipilih. Selain itu, komponen ion-refresher digunakan untuk memperbarui data produk dengan menggeser layar ke bawah.

Data diambil dari API menggunakan fungsi fetchProducts(), seperti yang ditunjukkan di bawah ini:

```

async fetchProducts() {
  this.apiService.getProducts().subscribe(
    async (response: any) => {
      if (Array.isArray(response) && response.length > 0) {
        this.products = response.sort((a, b) =>
          new Date(b.createdAt).getTime() - new
Date(a.createdAt).getTime()
        );
      } else {
        this.products = [];
      }
    },
    async (error) => {
      await this.presentToast('Gagal memuat produk. Silakan coba lagi
nanti.', 'danger');
    }
  );
}

```

Gambar 3. 12 Kode Pengambilan Data Produk dari API

Kode ini mengimplementasikan mekanisme langganan untuk memantau respons dari server. Data produk diurutkan berdasarkan waktu pembuatan (createdAt), sehingga item terbaru muncul di bagian atas. Jika koneksi gagal, sistem akan menampilkan pesan toast sebagai umpan balik kepada pengguna. Melalui latihan ini, diperoleh pemahaman tentang konsep pengambilan data asinkron, pemrosesan respons API, dan implementasi notifikasi visual melalui toast.

Hasil tampilan latihan Aplikasi Katalog Produk ditampilkan pada Gambar 3.13, di mana daftar produk berhasil dimuat dengan tata letak kartu yang rapi, dan fitur penyegaran data berfungsi secara optimal.



Gambar 3. 13 Tampilan Akhir Aplikasi Katalog Produk.

3.3.4 Pengembangan Aplikasi S2P-Mobile

Sebelum perubahan, pengumpulan data dan pelaporan *Health, Safety, and Environment* (HSE) di PT Sumber Segara Primadaya (S2P) masih mengandalkan metode manual menggunakan formulir kertas dan catatan tulisan tangan. Pendekatan tradisional ini sering menimbulkan masalah, seperti keterlambatan dalam pengiriman laporan, risiko kesalahan saat mencatat data, dan kesulitan dalam pengolahan dan analisis data. Akibatnya, pengawasan kegiatan HSE di lapangan menjadi kurang efektif, yang pada gilirannya mempengaruhi kemampuan perusahaan untuk menangani insiden dengan cepat dan mengambil keputusan yang tepat oleh tim manajemen.

Sebagai tanggapan atas masalah-masalah tersebut, perusahaan memutuskan untuk melakukan transformasi digital pada sistem pelaporan HSE agar data dapat dikumpulkan dan dipantau secara langsung dan tepat waktu. Dalam upaya ini, PT Sumber Segara Primadaya berkolaborasi dengan PT Nocola IoT Solution untuk merancang dan mengembangkan aplikasi seluler bernama S2P-Mobile. Aplikasi ini bertujuan untuk menyederhanakan proses pelaporan, pemantauan, dan penanganan insiden di lapangan dengan pendekatan yang lebih akurat, cepat, dan terintegrasi langsung ke dalam sistem manajemen HSE (*Health, Safety, and Environment*).

Aplikasi S2P-Mobile dikembangkan menggunakan Ionic Framework yang dipadukan dengan *Angular* sebagai framework frontend. Pilihan teknologi ini didasarkan pada kemampuan Ionic

untuk mengembangkan aplikasi yang kompatibel di berbagai platform (*cross-platform*), seperti Android dan iOS, dari satu basis kode. Selain itu, penggunaan *Capacitor* sebagai runtime asli memungkinkan akses ke fitur perangkat seperti GPS, kamera, penyimpanan lokal, dan notifikasi, sehingga aplikasi dapat berjalan lebih responsif dan memberikan pengalaman pengguna yang lebih baik di berbagai jenis perangkat.

Dalam proyek pengembangan ini, peran Frontend Developer dilanjutkan dengan fokus pada pembuatan dan implementasi elemen Antarmuka Pengguna (UI) dan Pengalaman Pengguna (UX) untuk aplikasi S2P-Mobile. Fitur utama yang dikembangkan meliputi Otentikasi *Login* menggunakan CAPTCHA, Panic Button, dan Notifikasi *Real-Time*. Ketiga fitur ini diselesaikan melalui tiga *sprint* pengembangan yang terstruktur, di mana setiap *sprint* berfokus pada satu kelompok fitur utama. Pendekatan ini dipilih karena fitur-fitur tersebut memiliki peran penting dalam mendukung pelaporan keselamatan kerja digital, mempercepat komunikasi antar divisi, dan meningkatkan perlindungan data pengguna di lingkungan kerja PT Sumber Segara Primadaya.

a. Metode Pengembangan

Dalam proses pengembangan aplikasi S2P-Mobile di PT Nocola IoT Solution, metode Agile Development diterapkan karena dianggap paling sesuai untuk proyek yang menekankan kerja tim, fleksibilitas, dan kemampuan beradaptasi dengan kebutuhan pengguna yang terus berubah. Pendekatan ini memprioritaskan kolaborasi antar anggota tim, pengembangan bertahap, dan umpan balik berkelanjutan dari atasan dan pengguna akhir di setiap tahap. Strategi ini membuat proses pengembangan menjadi lebih efektif, dinamis, dan terfokus, dengan kemudahan menyesuaikan fitur sesuai dengan kondisi lapangan yang terus berkembang. Dalam konteks proyek S2P-Mobile, metode Agile direalisasikan melalui tiga *sprint* utama yang masing-masing berfokus pada:

1. Pengembangan halaman *Login* dengan CAPTCHA.
2. Penyusunan fitur Panic Button.
3. Implementasi sistem Notifikasi *Real-Time*.

Dalam proyek ini, metode *Agile* dibagi menjadi beberapa siklus kerja atau *sprint*, di mana setiap *sprint* berfokus pada satu fitur utama yang dikembangkan secara mandiri hingga mencapai tahap pengujian dan evaluasi. Setiap *sprint* dimulai dengan fase perencanaan, diikuti oleh pengembangan fitur, pengujian, dan tinjauan. Model pengembangan ini digambarkan secara visual pada Gambar 3.14 di bawah ini, yang menggambarkan siklus berulang dari setiap tahap dalam metode Agile Development.



Gambar 3. 14 Diagram Siklus Agile Development pada Aplikasi S2P-Mobile

Seperti yang ditunjukkan dalam diagram, proses Agile bersifat iteratif dan berkelanjutan, di mana hasil dari satu *sprint* digunakan sebagai acuan untuk perbaikan pada *sprint* berikutnya. Dalam proyek ini, pengembangan dilakukan melalui tiga *sprint* utama, yaitu pembuatan fitur Login dengan CAPTCHA pada *sprint* pertama, fitur Panic Button pada *sprint* kedua, dan fitur Notifikasi *Real-Time* pada *sprint* ketiga. Pembagian berdasarkan fitur ini membantu membuat proses pengembangan lebih terukur, lebih mudah dievaluasi, dan lebih lancar dalam koordinasi dengan supervisor.

Dalam praktiknya, peran pengembangan *frontend* difokuskan pada perancangan dan implementasi Antarmuka Pengguna atau *User Interface (UI)* dan Pengalaman Pengguna atau *User Experience (UX)* menggunakan Ionic Framework berbasis Angular. Setiap halaman aplikasi dibuat secara modular untuk memudahkan pengelolaan, pengujian, dan integrasi dengan elemen lain. Desain antarmuka mengikuti desain dari tim desain di Figma, dan diadaptasi ke Material Design untuk Android dan Human Interface Guidelines untuk iOS, menghasilkan tampilan dan nuansa yang konsisten, responsif, dan dapat diakses.

Selain itu, implementasi *Capacitor* dalam Ionic sangat membantu karena memungkinkan integrasi dengan fitur perangkat asli seperti GPS untuk melacak lokasi pada fitur *Panic Button*, Pemberitahuan Push untuk fitur Notifikasi *Real-Time*, dan serta penyimpanan lokal berbasis *Capacitor Preferences* sehingga aplikasi dapat terus beroperasi tanpa koneksi internet. Berkat mekanisme pengikatan data *Angular*, pembaruan data di server dapat disinkronkan secara instan ke antarmuka pengguna, memastikan kinerja yang stabil dan efisien.

Melalui penggunaan metode Agile Development, proses pengembangan aplikasi S2P-Mobile menjadi lebih terorganisir, adaptif, dan berfokus pada hasil. Setiap *sprint* menghasilkan produk yang siap diuji dan dioptimalkan, memungkinkan pengembang untuk melakukan perbaikan cepat sebelum beralih ke tahap berikutnya. Dengan strategi ini, pengembang telah memberikan

kontribusi penting dalam pengembangan tiga fitur kunci *Login* dengan CAPTCHA, *Panic Button*, dan Notifikasi *Real-Time* yang mendukung transformasi digital sistem *Health, Safety, and Environment* (HSE) di PT Sumber Segara Primadaya (S2P).

b. Tahapan Pengembangan

Proses pengembangan aplikasi S2P-Mobile dilakukan melalui serangkaian tahap berdasarkan metode Agile Development, di mana setiap tahap direalisasikan sebagai siklus kerja atau *sprint*. Setiap *sprint* memiliki tujuan spesifik untuk menyelesaikan satu fitur utama, mulai dari tahap desain, implementasi, pengujian, hingga evaluasi. Pendekatan ini memungkinkan pengembang untuk fokus pada penyelesaian fitur-fitur spesifik dalam jangka waktu yang terukur, sambil menjaga kualitas dan stabilitas antarmuka pengguna.

Dalam proyek ini, pengembangan aplikasi dilakukan melalui tiga siklus *sprint* utama, masing-masing berfokus pada pengembangan fitur Login dengan CAPTCHA, Panic Button, dan Notifikasi *Real-Time*. Semua aktivitas pengembangan mengikuti desain UI/UX yang dibuat oleh tim desain melalui Figma dan mematuhi instruksi teknis dari supervisor proyek selama periode magang. Rincian tahap kerja dapat dilihat pada Tabel 3.2 di bawah ini:

Tabel 3. 3 Daftar Penugasan Pengembangan Aplikasi S2P-Mobile

No	<i>Sprint</i>	Deskripsi Kegiatan	Waktu Kegiatan
1.	<i>Sprint 1</i>	Implementasi desain antarmuka halaman <i>Login</i> dengan CAPTCHA	25 Oktober 2024 – 15 November 2024
2.	<i>Sprint 1</i>	Pengujian fungsionalitas halaman <i>Login</i> pada simulator dan real-device	16 November – 18 November 2024
3.	<i>Sprint 2</i>	Implementasi desain antarmuka halaman <i>Panic Button</i>	19 November – 10 Desember 2024
4.	<i>Sprint 2</i>	Pengujian fungsionalitas halaman <i>Panic Button</i> pada simulator dan real-device	11 Desember – 13 Desember 2024
5.	<i>Sprint 3</i>	Implementasi desain antarmuka halaman Notifikasi <i>Real-Time</i>	14 Desember – 28 Desember 2024
6.	<i>Sprint 3</i>	Pengujian fungsionalitas halaman Notifikasi <i>Real-Time</i> pada simulator dan real-device	29 Desember 2024 – 5 Januari 2025

1. Implementasi Fitur

Proses implementasi dilakukan secara bertahap sesuai dengan pembagian *sprint*. Setiap fitur dikembangkan menggunakan pendekatan modular dengan Ionic Framework dan Angular, yang memungkinkan pengelolaan kode *frontend* secara terstruktur, integrasi komponen, serta pengelolaan state aplikasi secara konsisten. Selain itu, implementasi *frontend* memperhatikan

responsivitas tampilan agar aplikasi dapat berjalan optimal pada berbagai ukuran layar perangkat Android dan iOS.

a. *Sprint 1* Fitur Login dengan CAPTCHA

Pada *sprint* pertama, upaya pengembangan difokuskan pada pembuatan halaman *Login* dengan CAPTCHA, yang merupakan elemen kunci dalam proses autentikasi pengguna. Fitur ini dirancang untuk menjaga keamanan sistem dengan memverifikasi bahwa pengguna yang mengakses sistem adalah manusia, bukan program otomatis.

Implementasi halaman *login* dengan CAPTCHA dilakukan berdasarkan rancangan antarmuka yang telah disediakan, kemudian diterjemahkan ke dalam komponen *frontend* menggunakan ionic dan Angular, seperti yang ditunjukkan pada Gambar 3.15, yang mencakup tiga elemen utama, yaitu bidang untuk nama pengguna, bidang untuk kata sandi, dan bagian CAPTCHA yang menampilkan kombinasi acak huruf dan angka. CAPTCHA dibuat melalui fungsi `generateCaptcha()` di sisi Frontend, sementara fungsi `verifyCaptcha()` bertugas memvalidasi masukan pengguna sebelum memberikan akses ke halaman utama.



Gambar 3. 15 Acuan Desain Antarmuka Halaman *Login*

Selama implementasi, beberapa penyesuaian implementasi desain dilakukan berdasarkan saran dari supervisor di divisi pengembangan aplikasi mobile. Penyesuaian ini menekankan

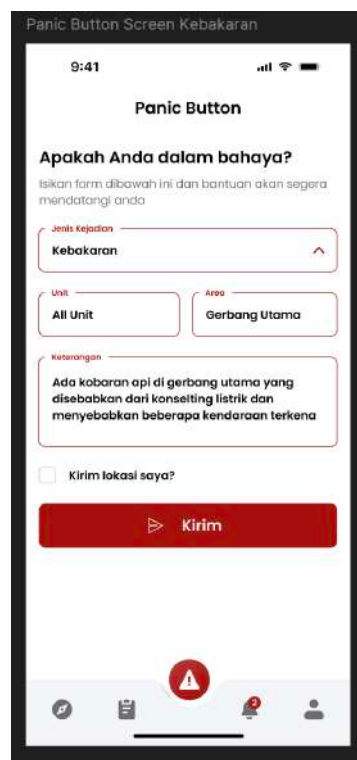
penyesuaian implementasi frontend dilakukan untuk meningkatkan kejelasan komponen input dan efektivitas validasi CAPTCHA agar lebih mudah digunakan pada berbagai ukuran layar. Hasil penyesuaian ini menghasilkan tampilan akhir seperti yang terlihat pada Gambar 3.25.

Berdasarkan hasil pengujian, halaman *login* berfungsi dengan baik di perangkat Android dan iOS, di mana fungsi validasi CAPTCHA bekerja secara langsung dan menolak masukan yang tidak valid. Fitur ini terbukti efektif dalam meningkatkan keamanan proses *login* pengguna.

b. *Sprint 2* Fitur Panic Button

Pada *sprint* kedua, upaya pengembangan beralih ke pembuatan fitur Panic Button, yang bertujuan untuk memudahkan pelaporan darurat secara cepat dan akurat. Fitur ini merupakan bagian penting dalam mendukung sistem *Health, Safety, and Environment* (HSE) di tempat kerja.

Rancangan antarmuka halaman *Panic Button* ditampilkan pada Gambar 3.16, dengan elemen utama berupa formulir pelaporan yang mencakup kolom untuk jenis insiden, unit kerja, lokasi, dan informasi pendukung. Untuk meningkatkan akurasi data, fitur ini terhubung dengan plugin *Capacitor Geolocation*, yang secara otomatis mengambil dan mengirim koordinat lokasi pengguna saat tombol “Kirim Pemberitahuan” ditekan.



Gambar 3. 16 Desain Figma Halaman *Panic Button*

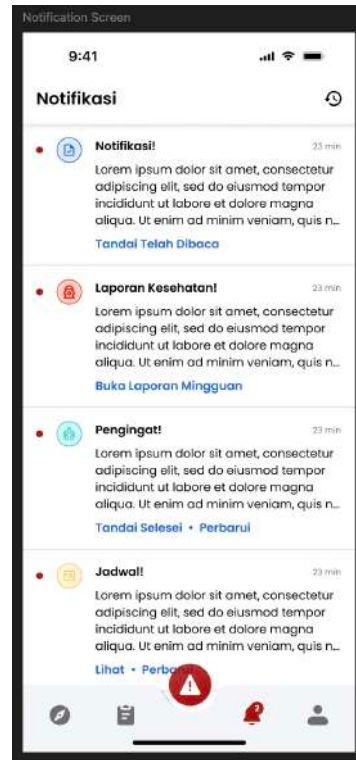
Selama tahap pengembangan, supervisor memberikan masukan untuk menyederhanakan tata letak agar pengguna dapat mengirim laporan dengan lebih cepat. Selain itu, penyesuaian implementasi *frontend* dilakukan pada komponen tombol dan alur interaksi pengguna agar proses pengiriman laporan darurat dapat dilakukan secara cepat dan jelas. Hasil penyesuaian ini dapat dilihat pada tampilan akhir pada Gambar 3.2.

Hasil pengujian menunjukkan bahwa fitur ini beroperasi dengan baik dan mampu mengirimkan laporan ke server dalam waktu kurang dari 3 detik. Data lokasi terdeteksi dengan akurasi sekitar ± 10 meter, dan notifikasi konfirmasi pengiriman muncul dengan benar di layar pengguna.

c. *Sprint 3* Fitur Notifikasi *Real-Time*

Pada tahap akhir pengembangan, upaya difokuskan pada implementasi fitur Notifikasi *Real-Time*, yang bertugas mengirimkan pesan keselamatan, peringatan kerja, dan informasi penting lainnya kepada pengguna aplikasi. Tujuan fitur ini adalah memastikan bahwa setiap pembaruan dari HSE dapat diterima dengan cepat dan tetap tersedia bahkan saat perangkat dalam keadaan *offline*.

Rancangan antarmuka halaman notifikasi ditampilkan pada Gambar 3.17, di mana tampilan disusun dalam tampilan daftar yang mencakup elemen pesan, waktu, dan indikator status dibaca. Pengembangan fitur ini menggunakan mekanisme notifikasi pada Ionic serta penyimpanan data lokal berbasis Capacitor Preferences, menerapkan pendekatan *offline-first* sehingga notifikasi tetap dapat diakses tanpa koneksi internet.



Gambar 3. 17 Desain Figma Halaman Notifikasi *Real-Time*

Selama implementasi, supervisor memberikan masukan untuk menambahkan mekanisme penanda status baca pada data notifikasi yang dikelola di sisi *frontend* dan memperbaiki jarak antar elemen untuk tampilan yang lebih seimbang. Setelah penyesuaian, tampilan akhir fitur ini dapat dilihat pada Gambar 3.29.

Berdasarkan hasil pengujian menggunakan simulator iOS serta perangkat fisik iPhone XR dengan sistem operasi iOS 18.7.2, fitur Notifikasi *Real-Time* menampilkan pesan dengan akurat dan sinkronisasi data notifikasi antara *backend* dan penyimpanan lokal *frontend* menggunakan Capacitor Preferences berjalan dengan baik. Pemberitahuan dapat diterima langsung melalui pemberitahuan push tanpa penundaan dan masih dapat dibuka saat aplikasi dalam mode *offline*.

2. Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan bahwa setiap komponen antarmuka pengguna dan interaksi pengguna pada aplikasi S2P-Mobile berjalan sesuai dengan spesifikasi desain dan kebutuhan fungsional yang telah ditentukan. Pengujian ini hanya berfokus pada aspek *frontend*, karena logika *backend*, validasi server, dan pemrosesan data ditangani oleh tim *backend* PT Nocola IoT Solution. Pengujian dilakukan pada dua lingkungan utama, yaitu simulator iOS melalui Xcode dan perangkat fisik iPhone XR, sebagai representasi penggunaan aplikasi pada kondisi nyata di lapangan.

a. Instrumen Pengujian

Instrumen pengujian yang digunakan dirangkum dalam Tabel 3.3 berikut.

Tabel 3. 4 Instrumen Pengujian Fungsional

No	Instrumen Pengujian	Detail Pengujian	Tujuan
1.	Xcode iOS Simulator	iPhone 15 Pro, iPhone SE	Menguji UI pada berbagai ukuran layar dan perangkat virtual
2.	Perangkat Fisik iPhone XR	iOS 18.7.3	Menguji interaksi nyata: GPS, modal, notifikasi lokal, input form, performa rendering
3.	Ionic CLI (<i>ionic serve / capacitor run</i>)	Build dan deploy aplikasi	Menjalankan aplikasi pada simulator dan perangkat fisik
4.	Safari Web Inspector	Debugging Frontend	Melihat error, log, dan kondisi runtime aplikasi
5.	Xcode Build Tools	Menjalankan aplikasi via USB	Deploy aplikasi ke perangkat fisik untuk pengujian langsung

Instrumen ini memastikan pengujian mencakup dua hal:

1. Validasi visual dan perilaku UI melalui simulator.
2. Validasi interaksi nyata melalui real device.

b. Tujuan Pengujian

Pengujian dilakukan untuk memastikan bahwa:

1. Halaman *Login* dengan CAPTCHA dapat menampilkan kode CAPTCHA secara acak, memvalidasi input pengguna, dan mengarahkan pengguna ke halaman berikutnya apabila validasi berhasil.
2. Fitur Panic Button mampu menampilkan form dengan benar, menampilkan pilihan data pada *dropdown*, dan mengambil lokasi perangkat dengan akurat ketika laporan dikirim.
3. Halaman Notifikasi dapat menampilkan daftar notifikasi, membuka detail pesan, menandai pesan sebagai dibaca, dan menyimpan status tersebut secara lokal.
4. Seluruh tampilan UI dapat menyesuaikan diri dengan berbagai ukuran layar tanpa terjadi pergeseran komponen atau kerusakan layout.

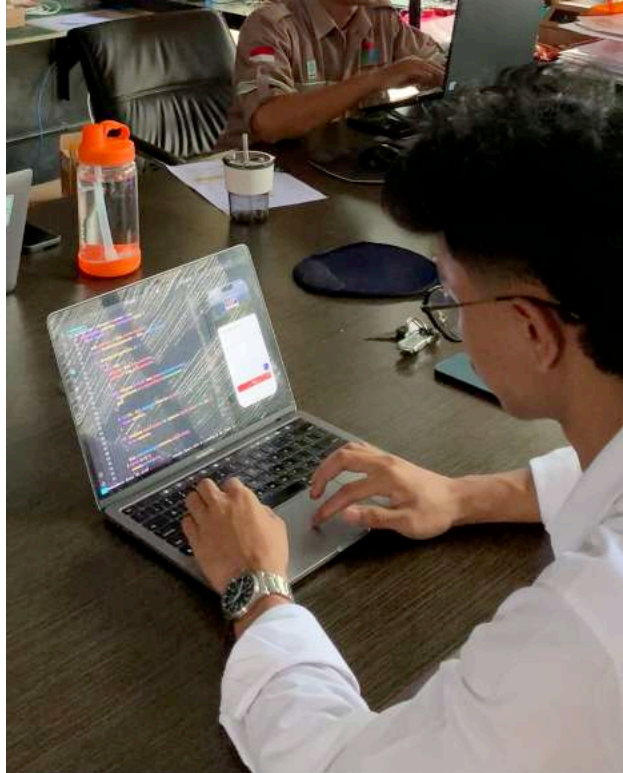
c. Skenario Hasil Pengujian

Pengujian dilakukan berdasarkan skenario fungsional untuk memastikan UI berjalan sesuai dengan rancangan.

Tabel 3. 5 Hasil Pengujian Fungsional Aplikasi S2P-Mobile

No	Fitur	Skenario Pengujian	Hasil Diharapkan	Hasil Pengujian	Status
1.	<i>Login + CAPTCHA</i>	CAPTCHA muncul secara acak	Kode terdiri dari 5 karakter acak	Tampil dengan benar	Lulus
2.	<i>Login + CAPTCHA</i>	Refresh CAPTCHA	Kode berubah saat tombol ditekan	Berhasil berubah	Lulus
3.	<i>Login + CAPTCHA</i>	Input Salah	Muncul pesan kesalahan	Pesan muncul dan CAPTCHA direset	Lulus
4.	<i>Login + CAPTCHA</i>	Input Benar	Pengguna diarahkan ke halaman utama	Aplikasi melanjutkan proses <i>login</i>	Lulus
5.	<i>Panic Button</i>	Form tampil lengkap	Semua field dapat dipilih	Berfungsi sesuai harapan	Lulus
6.	<i>Panic Button</i>	Dropdown dipilih	Data tampil dan dapat dipilih	Berfungsi sesuai harapan	Lulus
7.	<i>Panic Button</i>	Pengambilan lokasi	GPS memberikan koordinat	Lokasi terbaca kurang lebih 10 meter	Lulus
8.	<i>Panic Button</i>	Pengiriman laporan	Pesan konfirmasi muncul	Alert tampil dan form kosong kembali	Lulus
9.	Notifikasi	Daftar pesan tampil	Seluruh notifikasi terlihat	Tampilan normal	Lulus
10.	Notifikasi	Detail pesan	Modal terbuka	Modal menampilkan isi pesan	Lulus
11.	Notifikasi	Tandai membaca	Status berubah dan tersimpan lokal	Berhasil	Lulus
12.	UI Responsif	Uji berbagai ukuran layar	Layout tidak rusak	Tampilan konsisten	Lulus

d. Dokumentasi Pengujian



Gambar 3. 18 Pengujian Fungsional pada Simulator Menggunakan Xcode



Gambar 3. 19 Pengujian Fungsional pada Perangkat Fisik Menggunakan Xcode

e. Kesimpulan Pengujian

Berdasarkan hasil pengujian, seluruh fitur *frontend* yang dikembangkan telah berjalan dengan baik sesuai spesifikasi. Validasi CAPTCHA bekerja dengan benar, fitur Panic Button dapat memproses input dan membaca lokasi perangkat, serta halaman Notifikasi mampu menyimpan data secara lokal dan menampilkan daftar pesan secara konsisten. Tidak ditemukan kendala yang mengganggu proses penggunaan fitur selama pengujian pada simulator maupun perangkat fisik.

3.4 Hasil

Aplikasi S2P-Mobile dikembangkan khusus untuk PT Sumber Segara Primadaya (S2P) dengan tujuan mengoptimalkan sistem *Health, Safety, and Environment* (HSE) melalui pendekatan digital. Penggunaan kerangka kerja Ionic memudahkan pengembangan aplikasi lintas platform menggunakan satu set kode sumber, yang dapat beroperasi di perangkat Android dan iOS. Kerangka kerja ini dilengkapi dengan elemen antarmuka bawaan yang sesuai dengan prinsip

Material Design dan Pedoman Antarmuka Pengguna iOS, sehingga tampilan aplikasi menjadi adaptif, responsif, dan konsisten di berbagai jenis perangkat.

Kerangka kerja Ionic menyediakan berbagai komponen antarmuka, seperti `<ion-header>`, `<ion-toolbar>`, `<ion-content>`, `<ion-card>`, dan `<ion-modal>`, yang memudahkan dalam membuat antarmuka pengguna yang efektif. Selain itu, Ionic juga menawarkan CSS Utilities untuk mengelola tata letak, beserta sistem grid dan flexbox yang mendukung desain adaptif di berbagai resolusi layar. Melalui kemampuan ini, pengembang dapat membuat antarmuka modern yang sesuai dengan tuntutan operasional di lapangan.

Dalam proyek pengembangan ini, pengembangan difokuskan pada tiga fitur utama, yaitu *Login* dengan CAPTCHA, Panic Button, dan Notifikasi *Real-Time*. Ketiga fitur ini memainkan peran penting dalam mendorong transformasi digital pada sistem pelaporan dan keselamatan kerja di PT Sumber Segara Primadaya. Semua fitur dikembangkan menggunakan Ionic Framework, didukung oleh *Angular* untuk pengelolaan data dinamis dan *Capacitor* untuk terhubung ke fungsi hardware seperti GPS dan penyimpanan lokal.

3.4.1 Halaman *Login*

Fitur CAPTCHA Login dibuat untuk meningkatkan tingkat keamanan dalam proses verifikasi pengguna. CAPTCHA, yang merupakan singkatan dari Completely Automated Public Turing Test to Tell Computers and Humans Apart, dirancang untuk memverifikasi bahwa akses *login* berasal dari manusia, bukan program otomatis. Aplikasi ini efektif dalam mencegah serangan brute-force dan upaya *login* oleh bot pada platform aplikasi.

Antarmuka halaman *login* dirancang secara ringkas, menyediakan bidang input untuk nama pengguna, kata sandi, dan CAPTCHA, yang berisi kombinasi acak huruf dan angka. Komponen `<ion-input>` digunakan untuk membentuk bidang input, sementara `<ion-button>` digunakan sebagai tombol untuk menjalankan aksi *login*. Berikut adalah potongan kode yang diterapkan pada halaman *login*:

```
<ion-card>
  <ion-card-header>
    <ion-title class="text-center font-bold">Login S2P-Mobile</ion-title>
  </ion-card-header>
  <ion-card-content>
    <ion-input placeholder="Masukkan Username"
[ (ngModel) ]="username"></ion-input>
    <ion-input type="password" placeholder="Masukkan Password"
[ (ngModel) ]="password"></ion-input>
    <div class="captcha-container">
```

```

    <p class="captcha-text">{{ captchaCode }}</p>
    <ion-input placeholder="Masukkan CAPTCHA"
  [(ngModel)]="inputCaptcha"></ion-input>
  </div>
  <ion-button expand="block" color="primary"
  (click)="verifyCaptcha()">Login</ion-button>
</ion-card-content>
</ion-card>

```

Gambar 3. 20 Kode Program Halaman *Login*

Pada Gambar 3.20, komponen `<ion-card>` digunakan untuk membungkus seluruh konten, sehingga tampilan keseluruhan menjadi lebih terstruktur. Elemen CAPTCHA ditampilkan melalui teks dengan gaya font acak yang dibuat secara dinamis. Proses verifikasi CAPTCHA dilakukan menggunakan logika sederhana dalam berkas TypeScript, sebagai berikut:

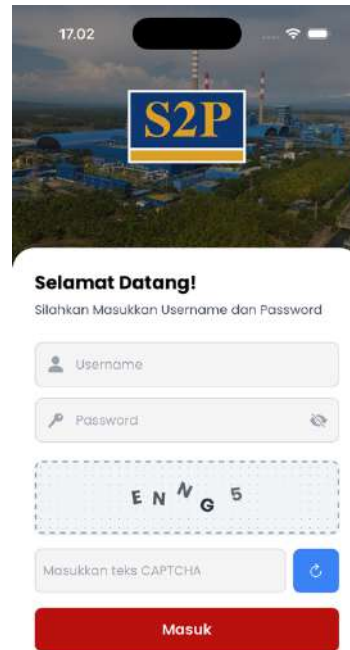
```

generateCaptcha() {
  const chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
  this.captchaCode = Array.from({ length: 6 }, () =>
    chars.charAt(Math.floor(Math.random() * chars.length))
  ).join('');
}

verifyCaptcha() {
  if (this.inputCaptcha !== this.captchaCode) {
    alert('Kode CAPTCHA tidak sesuai, silakan coba lagi.');
```

Gambar 3. 21 Potongan Kode Fungsi `generateCaptcha()` dan `verifyCaptcha()`.

Tampilan akhir halaman *login* ditunjukkan pada Gambar 3.22. Pengujian fungsional menunjukkan bahwa sistem dapat memverifikasi input CAPTCHA dengan benar serta menolak akses jika input salah.



Gambar 3. 22 Tampilan Akhir Halaman *Login* dengan CAPTCHA

3.4.2 Halaman *Panic Button*

Fitur Panic Button dirancang untuk memudahkan pelaporan insiden darurat secara cepat dan terorganisir. Melalui fitur ini, pengguna dapat melaporkan insiden seperti kecelakaan, situasi berbahaya, atau ancaman potensial langsung ke tim HSE, disertai dengan detail penting seperti jenis insiden, lokasi, dan deskripsi singkat.

Desain antarmuka halaman *Panic Button* mencakup elemen-elemen seperti `<ion-input>`, `<ion-textarea>`, dan `<ion-button>`. Pendekatan desain ini memudahkan pengguna untuk menyusun laporan dengan efisien. Berikut adalah potongan kode untuk halaman *Panic Button*:

```
<ion-input formControlName="jenisKejadian" label="Jenis Kejadian"
fill="outline"></ion-input>
<ion-input formControlName="unit" label="Unit" fill="outline"></ion-
input>
<ion-input formControlName="lokasi" label="Lokasi Kejadian"
fill="outline"></ion-input>
<ion-textarea formControlName="keterangan" label="Keterangan"
fill="outline"></ion-textarea>
<ion-button expand="block" color="danger" (click)=" kirimLaporan()">Kirim
Laporan Darurat</ion-button>
```

Gambar 3. 23 Potongan Kode Tampilan Formulir Pelaporan Darurat pada Fitur *Panic Button*.

Fitur ini diintegrasikan dengan *Capacitor Geolocation Plugin* untuk mengambil koordinat lokasi pengguna. Berikut potongan kode untuk mendapatkan lokasi otomatis saat laporan dikirim:

```
async kirimLaporan() {
  const position = await Geolocation.getCurrentPosition();
  const data = {
    jenisKejadian: this.form.value.jenisKejadian,
    lokasi: position.coords,
    waktu: new Date(),
  };
  this.http.post('https://<domain-internal>/laporan', data).subscribe(()
=> {
  alert('Laporan darurat berhasil dikirim.');
```

Gambar 3. 24 Potongan Kode Fungsi kirimLaporan()

Tampilan halaman *Panic Button* ditunjukkan pada Gambar 3.25. Fitur ini telah diuji menggunakan simulator dan perangkat nyata, dengan hasil bahwa laporan dapat dikirim lengkap beserta informasi lokasi pengguna secara otomatis.

Gambar 3. 25 Tampilan Akhir Halaman *Panic Button*

3.4.3 Halaman Notifikasi

Fitur Notifikasi *Real-Time* dirancang untuk mendistribusikan informasi keselamatan kerja, peringatan darurat, dan pengumuman penting kepada semua pengguna aplikasi. Fitur ini menggunakan mekanisme notifikasi pada Ionic serta penyimpanan lokal berbasis Capacitor Preferences untuk menyimpan data pesan, sehingga tetap dapat diakses bahkan ketika perangkat tidak terhubung ke internet (pendekatan offline-first).

Berikut potongan kode untuk menampilkan daftar notifikasi:

```
<ion-card *ngFor="let notif of notifications" [ngClass]="{'bg-light': !notif.isRead}">
  <ion-card-header>
    <ion-title>{{ notif.title }}</ion-title>
    <ion-note>{{ notif.timeAgo }}</ion-note>
  </ion-card-header>
  <ion-card-content>
    <p>{{ notif.message }}</p>
```

```

<ion-button fill="clear" color="primary" (click)="markAsRead(notif)">
  Tandai Telah Dibaca
</ion-button>
</ion-card-content>
</ion-card>

```

Gambar 3. 26 Potongan Kode Tampilan Daftar Notifikasi.

Kode berikut menangani pengambilan data notifikasi dari server dan menyimpannya ke penyimpanan lokal menggunakan Capacitor Preferences agar data tetap tersedia ketika aplikasi dijalankan secara *offline*:

```

async fetchNotifications() {
  const response = await this.http.get('https://<domain-internal>/notifications').toPromise();
  this.notifications = response as Notification[];
  await Preferences.set({
    key: 'notifications',
    value: JSON.stringify(this.notifications),
  });
}

```

Gambar 3. 27 Potongan Kode Fungsi fetchNotifications().

Selain itu, fungsi markAsRead() memperbarui status pesan menjadi telah dibaca:

```

async markAsRead(notif: any) {
  notif.isRead = true;
  await Preferences.set({
    key: 'notifications',
    value: JSON.stringify(this.notifications),
  });
}

```

Gambar 3. 28 Potongan Kode Fungsi markAsRead().

Fitur ini diuji menggunakan simulator iOS serta perangkat fisik iPhone XR dengan iOS 18.7.2 untuk memastikan sinkronisasi data dan responsivitas antarmuka berjalan dengan baik untuk memastikan sinkronisasi berjalan lancar dan tampilan tetap responsif. Tampilan akhir halaman Notifikasi *Real-Time* dapat dilihat pada Gambar 3.29.



Gambar 3. 29 Tampilan Akhir Halaman Notifikasi *Real-Time*

3.5 Pembahasan

Pengembangan aplikasi S2P-Mobile menunjukkan bahwa pendekatan teknologi mobile lintas platform menggunakan Ionic Framework mampu menjawab kebutuhan digitalisasi sistem *Health, Safety, and Environment* (HSE) di PT Sumber Segara Primadaya (S2P). Hasil implementasi dan pengujian yang dilakukan selama magang memberikan bukti bahwa teknologi dan metode yang dipilih selaras dengan teori pada Bab II sekaligus relevan dengan kondisi operasional di lapangan.

Pemilihan Ionic Framework terbukti efektif karena memungkinkan pengembangan aplikasi Android dan iOS dari satu basis kode, sehingga proses implementasi dapat dilakukan lebih cepat dan efisien. Temuan ini sejalan dengan penelitian Akbar et al. (2024) dan Bidah & Noor Islami (2024), yang menyatakan bahwa Ionic mampu mengurangi beban pengembangan tanpa mengorbankan responsivitas tampilan antarmuka. Hal ini terasa nyata pada pengerjaan tiga fitur utama Login dengan CAPTCHA, Panic Button, dan Notifikasi *Real-Time* yang dapat dikembangkan dalam *sprint* terpisah tetapi tetap konsisten dari sisi tampilan dan logika.

Kontribusi *Angular* sebagai kerangka kerja *frontend* juga sangat signifikan. Berdasarkan implementasi pada subbab 3.2, *Angular* mempermudah pengelolaan data dinamis, validasi formulir, serta pengaturan status UI. Sebagai contoh, mekanisme data binding pada halaman *login*

memastikan input pengguna langsung terhubung dengan proses validasi CAPTCHA. Sementara itu, event binding seperti (click)="markAsRead(notification)" pada fitur notifikasi berfungsi memastikan interaksi pengguna tercatat dan tersimpan ke dalam storage lokal. Pendekatan ini sejalan dengan panduan desain antarmuka yang menekankan kesederhanaan navigasi dan efisiensi interaksi dalam aplikasi keselamatan kerja.

Selain itu, penggunaan *Capacitor* terbukti membantu menghubungkan aplikasi berbasis web dengan kemampuan perangkat asli. Pada fitur Panic Button, integrasi Geolocation memungkinkan aplikasi mengambil koordinat pengguna secara *real-Time* tanpa kendala berarti selama proses pengujian. Pada halaman Notifikasi *Real-Time*, penggunaan penyimpanan lokal berbasis *Capacitor Preferences* memastikan data notifikasi tetap tersimpan meskipun perangkat kehilangan koneksi internet, sehingga mendukung konsep *offline-first*.

3.5.1 Keterkaitan Hasil Pengujian dengan Tujuan Magang

Hasil pengujian fungsional yang dilakukan melalui simulator iOS dan perangkat fisik iPhone XR menunjukkan bahwa fitur yang dikembangkan telah bekerja sesuai dengan rancangan. Hasil ini secara langsung mendukung tujuan magang, yaitu menghasilkan antarmuka pengguna yang responsif, stabil, dan mudah digunakan dalam konteks operasional HSE. Beberapa poin keterkaitan yang terlihat dari hasil pengujian antara lain:

1. Validasi CAPTCHA berjalan baik dan menolak input yang salah, sehingga mendukung tujuan peningkatan keamanan autentikasi pengguna.
2. Formulir *Panic Button* menampilkan data dengan benar dan mengambil lokasi perangkat secara akurat, sesuai kebutuhan pelaporan keadaan darurat yang cepat dan tepat.
3. Fitur Notifikasi *Real-Time* mampu menampilkan pesan, memperbarui status terbaca, dan menyimpan data secara lokal, mendukung akses informasi di area dengan koneksi internet terbatas.
4. Responsivitas UI konsisten di berbagai ukuran layar, membuktikan bahwa desain antarmuka yang dikembangkan telah memenuhi kebutuhan pekerja lapangan yang menggunakan beragam perangkat.

Dengan demikian, hasil pengujian bukan hanya membuktikan kelayakan teknis fitur yang dibuat, tetapi juga memastikan bahwa kontribusi pengembangan antarmuka yang dilakukan mendukung tujuan proyek secara keseluruhan.

3.5.2 Temuan dan Tantangan Selama Magang

Selama pelaksanaan magang dan proses pengembangan aplikasi, ditemukan beberapa hal penting yang menjadi bagian dari pembelajaran teknis maupun non-teknis, yaitu:

1. Perbedaan perilaku tampilan antara simulator dan perangkat fisik Beberapa elemen UI seperti margin dan proporsi komponen tidak tampil sama, sehingga memerlukan iterasi desain dan debugging tambahan.
2. Stabilitas pengambilan lokasi perangkat Pada pengujian awal, integrasi Geolocation tidak selalu memberikan hasil yang konsisten, terutama ketika perangkat belum memberikan izin lokasi atau koneksi GPS kurang stabil.
3. Penyesuaian UI dengan desain Figma Tidak semua desain dapat langsung diterapkan melalui komponen Ionic sehingga diperlukan beberapa improvisasi UI agar tampilan tetap konsisten.
4. Pengelolaan penyimpanan lokal Pada fitur Notifikasi, pengaturan Preferences untuk menyimpan status “sudah dibaca” memerlukan penyesuaian ulang agar data tetap konsisten ketika aplikasi dibuka kembali.
5. Koordinasi dalam Agile *Sprint* Penyesuaian terhadap ritme evaluasi sprint dan umpan balik dari supervisor diperlukan, terutama ketika perubahan antarmuka harus dilakukan dalam waktu singkat.

Tantangan-tantangan tersebut membantu pemahaman terhadap konteks pengembangan aplikasi nyata yang tidak hanya bergantung pada teori, tetapi juga pada kondisi lingkungan teknis dan kebutuhan pengguna di lapangan.

3.5.3 Kesimpulan Pembahasan

Secara umum, pengembangan S2P-Mobile selama magang menunjukkan bahwa kombinasi Ionic Framework, Angular, Capacitor, serta metode *Agile* mampu menghasilkan aplikasi yang relevan dan mendukung kebutuhan digitalisasi HSE di PT Sumber Segara Primadaya. Implementasi fitur utama, keberhasilan pengujian, serta temuan selama proses pengembangan memberikan bukti bahwa tujuan magang telah tercapai dan hasilnya dapat dijadikan dasar pengembangan lebih lanjut pada tahap pengujian industri.

BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Refleksi Pelaksanaan Magang

Pelaksanaan magang di PT Nocola IoT Solution memberikan pengalaman yang signifikan dalam memahami penerapan teknologi modern di industri perangkat lunak. Selama enam bulan pelaksanaan magang, pengembangan aplikasi S2P-Mobile dilakukan dengan peran Frontend Developer yang terlibat langsung dalam proses perancangan dan implementasi antarmuka pengguna. Aplikasi S2P-Mobile dikembangkan untuk mendukung proses pelaporan dan pemantauan *Health, Safety, and Environment* (HSE) di PT Sumber Segara Primadaya (S2P). Keterlibatan dalam proyek ini menunjukkan bahwa pengembangan aplikasi tidak hanya berfokus pada penulisan kode, tetapi juga mencakup analisis kebutuhan pengguna, penyesuaian dengan proses bisnis perusahaan, serta perancangan solusi digital yang mampu memberikan dampak nyata terhadap operasional industri.

Dalam proses pengembangannya, aplikasi S2P-Mobile dibangun menggunakan Ionic Framework dan Angular sebagai teknologi utama yang memungkinkan pengembangan aplikasi lintas platform dengan satu basis kode. Fokus pengembangan diarahkan pada implementasi antarmuka pengguna dan pengalaman pengguna untuk tiga fitur inti, yaitu halaman Login dengan CAPTCHA, fitur Panic Button, serta halaman Notifikasi *Real-Time*. Desain antarmuka yang disediakan oleh tim UI/UX melalui Figma kemudian diimplementasikan ke dalam kode *frontend* menggunakan Ionic Framework.

PT Nocola IoT Solution menerapkan metode *Agile Development* dalam proses pengembangan aplikasi dengan membagi pekerjaan ke dalam beberapa *sprint*. Selama pelaksanaan magang, pengembangan dilakukan melalui tiga *sprint* utama yang masing-masing berfokus pada satu fitur inti aplikasi. Melalui metode iteratif ini, diperoleh pemahaman mengenai pentingnya perencanaan, komunikasi tim, evaluasi rutin, serta kemampuan menyesuaikan pengembangan dengan perubahan kebutuhan proyek. Pengalaman praktis tersebut memperlihatkan perbedaan yang cukup signifikan antara teori yang diperoleh selama perkuliahan dan praktik nyata di lingkungan industri, baik dari sisi penggunaan framework modern, manajemen waktu, maupun standar teknis yang diterapkan. Dengan demikian, kegiatan magang ini berperan sebagai jembatan yang memperkuat integrasi antara teori akademik dan praktik profesional di industri teknologi.

Pelaksanaan magang memiliki keterkaitan yang kuat dengan beberapa mata kuliah yang ditempuh dalam Program Studi Informatika Universitas Islam Indonesia. Mata kuliah Rekayasa Perangkat Lunak (RPL) memberikan landasan dalam memahami penyusunan kebutuhan sistem, alur pengembangan perangkat lunak, serta pentingnya dokumentasi, yang seluruhnya diterapkan dalam proses *sprint* selama magang. Selain itu, mata kuliah Pengembangan Aplikasi Berbasis Web (PABW) memberikan pemahaman dasar mengenai HTML, CSS, JavaScript, dan konsep routing, yang relevan dalam proses implementasi antarmuka menggunakan Ionic dan Angular.

Pengembangan fitur dalam aplikasi S2P-Mobile juga berkaitan dengan materi pada mata kuliah Sistem Jaringan Komputer (SJK), khususnya dalam memahami konsep komunikasi *client-server*, struktur *request* dan *response* API, serta mekanisme pertukaran data melalui protokol HTTP. Prinsip-prinsip dari mata kuliah Algoritma dan Struktur Data turut mendukung pengelolaan state aplikasi, pemrosesan input, serta perancangan logika komponen agar lebih efisien. Dari sisi etika kerja, mata kuliah Etika Profesi memberikan pemahaman mengenai profesionalitas, seperti menjaga kerahasiaan data perusahaan, disiplin dalam menyelesaikan tugas, serta sikap kooperatif dalam kerja tim. Secara keseluruhan, pengalaman magang menunjukkan bahwa teori perkuliahan menjadi bekal fundamental dalam beradaptasi dengan ritme kerja industri perangkat lunak yang dinamis.

4.2 Pembelajaran Magang

Kegiatan magang memberikan kesempatan untuk mengembangkan kemampuan teknis dan profesional secara lebih mendalam. Melalui keterlibatan dalam proyek S2P-Mobile, proses pengembangan aplikasi mobile dipahami secara menyeluruh, mulai dari interpretasi penerapan desain ke dalam kode, penerjemahan desain ke dalam komponen antarmuka, integrasi logika aplikasi, hingga pengujian fitur menggunakan simulator dan perangkat fisik. Pengalaman ini memperluas pemahaman bahwa pengembangan aplikasi tidak hanya menitikberatkan pada aspek visual, tetapi juga harus memperhatikan fungsionalitas, keandalan, dan kenyamanan pengguna.

Penerapan metode *Agile Development* membentuk pola kerja yang terstruktur melalui pembagian *sprint*, penentuan prioritas pekerjaan, pelaporan progres kepada supervisor, serta pelaksanaan revisi berdasarkan evaluasi rutin. Proses ini menunjukkan bahwa keberhasilan proyek tidak hanya ditentukan oleh kemampuan teknis, tetapi juga oleh komunikasi yang efektif, kemampuan memecahkan masalah, manajemen waktu, serta adaptasi terhadap perubahan kebutuhan. Dengan demikian, pelaksanaan magang membentuk pola pikir yang lebih sistematis dan profesional dalam menghadapi tantangan pekerjaan di industri teknologi.

4.2.1 Manfaat Magang

Pelaksanaan magang di PT Nocola IoT Solution memberikan manfaat yang signifikan baik dari sisi teknis maupun non-teknis. Dari sisi teknis, pengalaman langsung dalam menggunakan Ionic Framework berbasis Angular memperluas wawasan mengenai teknologi lintas platform yang belum dipelajari secara mendalam selama perkuliahan, namun sangat relevan di industri. Proses konversi desain dari Figma ke dalam antarmuka responsif memberikan pemahaman mengenai pentingnya konsistensi tampilan pada berbagai ukuran layar. Selain itu, pemanfaatan Capacitor untuk mengakses fitur perangkat seperti GPS dan penyimpanan lokal menunjukkan bagaimana teknologi berbasis web dapat terintegrasi dengan fungsi *native* perangkat *mobile*.

Dari sisi non-teknis, magang ini melatih disiplin kerja, komunikasi yang efektif, kemampuan menerima revisi secara profesional, serta penyelesaian masalah secara kolaboratif. Lingkungan kerja yang menekankan tanggung jawab, ketepatan waktu, dan kualitas hasil pekerjaan memberikan pemahaman bahwa keberhasilan di dunia profesional tidak hanya bergantung pada kemampuan teknis, tetapi juga pada sikap, etika kerja, dan kemampuan beradaptasi.

4.2.2 Kendala, Hambatan, dan Tantangan

Selama pelaksanaan magang, terdapat beberapa kendala dan tantangan yang dihadapi sebagai bagian dari proses pembelajaran. Tantangan utama yang dialami adalah adaptasi terhadap teknologi baru, khususnya Ionic Framework dan Angular. Pengalaman sebelumnya yang lebih berfokus pada pengembangan aplikasi native Android menggunakan Kotlin dan XML menyebabkan perlunya penyesuaian terhadap arsitektur berbasis komponen, mekanisme *routing*, serta konsep data binding yang diterapkan dalam Angular. Kendala tersebut secara bertahap dapat diatasi melalui pemanfaatan dokumentasi resmi, eksplorasi kode sumber, serta arahan dan bimbingan dari supervisor selama proses pengembangan.

Tantangan berikutnya berkaitan dengan konsistensi tampilan antarmuka pada perangkat Android dan iOS. Perbedaan perilaku beberapa komponen Ionic menyebabkan sejumlah elemen antarmuka tidak sepenuhnya sesuai dengan desain awal, sehingga diperlukan proses pengujian dan penyesuaian berulang menggunakan simulator maupun perangkat fisik untuk memastikan tampilan aplikasi tetap konsisten di berbagai platform. Selain itu, lokasi kerja yang terpisah dari kantor pusat juga menjadi kendala dalam koordinasi tim. Untuk mengatasi hal tersebut, komunikasi daring melalui Google Meet, WhatsApp, dan Trello dimanfaatkan secara optimal agar kolaborasi dan pemantauan progres pekerjaan tetap berjalan efektif.

Selain itu, beberapa revisi signifikan dilakukan pada halaman Login agar selaras dengan standar UI/UX yang ditetapkan oleh perusahaan. Meskipun proses revisi tersebut memerlukan penyesuaian dan ketelitian yang tinggi, hal ini memberikan pembelajaran penting mengenai fleksibilitas, kedisiplinan, serta kesiapan untuk terus meningkatkan kualitas hasil kerja. Secara keseluruhan, berbagai kendala dan tantangan yang dihadapi selama magang memberikan wawasan mendalam mengenai dinamika kerja di industri teknologi sekaligus membentuk sikap profesional dalam menghadapi permasalahan pengembangan perangkat lunak.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil magang di PT Nocola IoT Solution, dapat disimpulkan bahwa pengalaman ini memberikan manfaat yang sangat besar, baik secara teknis maupun profesional. Selama enam bulan pelaksanaan magang, pengembangan antarmuka pengguna dilakukan secara aktif pada tiga fitur utama aplikasi S2P-Mobile, yaitu *Login* dengan CAPTCHA, *Panic Button*, dan Notifikasi *Real-Time*. Ketiga fitur ini dikembangkan menggunakan kerangka kerja Ionic berbasis *Angular*, dengan pendekatan Agile Development yang mengharuskan kemampuan beradaptasi dengan kebutuhan yang berubah dan berkolaborasi dengan tim proyek.

Melalui kegiatan ini, diperoleh pemahaman mendalam mengenai penerapan konsep desain UI/UX di industri, mulai dari mengonversi desain Figma menjadi kode, menggunakan komponen Ionic, hingga menguji antarmuka lintas platform di Android dan iOS. Pengalaman ini juga memperluas pemahaman mengenai ekosistem pengembangan lintas platform yang efisien, responsif, dan mudah dipelihara. Dalam praktiknya, desain antarmuka yang efektif tidak hanya menekankan estetika, tetapi juga kecepatan, keterbacaan, dan kenyamanan pengguna di lapangan, terutama dalam implementasi HSE di lingkungan berisiko tinggi seperti pembangkit listrik Sumber Segara Primadaya (S2P).

Namun, magang ini juga menghadirkan beberapa tantangan. Perbedaan mendasar antara pengalaman akademis yang berfokus pada pengembangan native menggunakan Kotlin dan praktik industri yang mengharuskan penguasaan Ionic Framework berbasis JavaScript-TypeScript membutuhkan waktu yang cukup lama untuk beradaptasi. Penulis juga menghadapi keterbatasan dalam bimbingan langsung akibat jarak dengan supervisor. Hambatan-hambatan ini diatasi melalui pembelajaran mandiri, eksplorasi dokumentasi resmi, dan diskusi *online* dengan supervisor dan rekan magang. Dari proses tersebut, kemandirian, disiplin, dan keterampilan pemecahan masalah sangat penting di dunia profesional.

Secara keseluruhan, magang ini berkontribusi secara signifikan dalam meningkatkan keterampilan teknis, memperkuat etos kerja, dan memahami dinamika proyek perangkat lunak di industri. Pengalaman ini berfungsi sebagai jembatan antara teori akademik dan praktik industri, serta menjadi landasan penting dalam mempersiapkan pengembangan karier di bidang pengembangan aplikasi *mobile* lintas platform.

5.2 Saran

Berdasarkan hasil pelaksanaan magang di PT Nocola IoT Solution, beberapa saran dapat disampaikan sebagai berikut:

PT Nocola IoT Solution perlu terus meningkatkan efektivitas program magangnya dengan menerapkan mekanisme pembimbingan yang lebih terorganisir dan terencana. Mengingat beberapa magang ditempatkan di lokasi yang berbeda dari supervisor, koordinasi rutin melalui sesi tinjauan *online* atau pertemuan *sprint* mingguan akan sangat bermanfaat bagi mahasiswa dalam memahami tugas mereka, mengevaluasi pekerjaan mereka, dan mempercepat proses penyesuaian. Selain itu, melibatkan mahasiswa dalam seluruh siklus proyek mulai dari tahap perencanaan dan pengembangan hingga tahap evaluasi akan memberikan pengalaman yang lebih komprehensif dan meningkatkan kualitas hasil magang.

Bagi mahasiswa yang akan mengikuti magang dalam pengembangan aplikasi *mobile*, sangat disarankan untuk mempersiapkan diri dengan pemahaman dasar tentang HTML, CSS, JavaScript, dan TypeScript, karena keterampilan ini sangat penting dalam pengembangan berbasis Ionic Framework. Selain keterampilan teknis, mahasiswa juga perlu mengasah keterampilan komunikasi, kerja tim, dan manajemen waktu agar dapat bekerja secara efektif meskipun dengan pengawasan yang terbatas. Mengambil pendekatan proaktif dalam mencari solusi dan bersedia menjelajahi dokumentasi atau sumber referensi tambahan akan sangat membantu mahasiswa dalam menghadapi dinamika industri yang cepat dan menantang.

Secara keseluruhan, magang ini memberikan pelajaran penting bahwa penguasaan teknologi dan kemampuan beradaptasi harus saling melengkapi. Oleh karena itu, mahasiswa tidak hanya akan siap secara akademis, tetapi juga kompeten dan profesional dalam menghadapi realitas dunia kerja yang sesungguhnya. Dengan demikian, hasil dari penelitian ini diharapkan dapat menjadi acuan bagi pengembangan aplikasi HSE berbasis *mobile* di industri serupa.

DAFTAR PUSTAKA

- Adesina, A. O., Ayobioloja, P. S., Obagbuwa, I. C., Odule, T. J., Afolorunso, A. A., & Ajagbe, S. A. (2023). An Improved Text-Based and Image-Based CAPTCHA Based on Solving and Response Time. *Computers, Materials & Continua*, 74(2), 2661–2675. <https://doi.org/10.32604/cmc.2023.031245>
- Akbar, M. I., Fahlevvi, M. R., & Muaafii, D. A. (2024). Rancang Bangun Sistem Perpustakaan Berbasis Web Menggunakan Ionic Framework di SMAN 1 Purwokerto. *Jurnal Teknologi Informasi*, 10. <https://doi.org/10.52643/jti.v10i1.4344>
- Akwilla, H., & Jollyta, D. (2024). Implementasi Agile Development Berbasis Web Based Pada Sistem Inventory. *Jurnal Manajemen Informatika Jayakarta*, 4(1), 138. <https://doi.org/10.52362/jmijayakarta.v4i1.1317>
- Alamsyah, R., Nugroho, I. M., & Alam, S. (2022). Redesign User Interface Dan User Experience Aplikasi Wastu Mobile Menggunakan Metode Design Thinking. *Jurnal Ilmiah Betrik*, 13(2), 152–159. <https://doi.org/10.36050/betrik.v13i2.506>
- Ali, O. V., Siboro, I., & Ramdan, M. (2024). Efektivitas Pelaporan Bahaya Pada Pt Petrosea Di Balikpapan. <https://doi.org/10.36277/identifikasi.v10i2.391>
- Amir, F., Hanafiyah, A. A., Maulana, F. R., Nursaidah, N., & Ahsan, A. M. (2025). Rencana dan Implementasi Manajemen Risiko Berbasis HSE dalam Proyek Seismik Migas di PT XYZ. *Jurnal Inovasi Global*, 3(6), 990–1001. <https://doi.org/10.58344/jig.v3i6.365>
- Andarini, R. Y., Hendradi, P., & Nugroho, S. (2023). Meningkatkan Keamanan Terhadap Sql Injection Studi Kasus Sistem Kepegawaian Bnn. *Indonesian Journal of Business Intelligence (IJUBI)*, 6(1). <https://doi.org/10.21927/ijubi.v6i1.3161>
- Ariesta, A., Wijaya, A. H., & Atmaja, M. A. (2024). Metode Agile Development Dalam Perancangan Undangan Digital Menggunakan Aplikasi Ionic Mobile Dan Angular Frameworks. *INFOTECH Journal*, 10(1), 161–165. <https://doi.org/10.31949/infotech.v10i1.9872>
- Arifin, A., & Yunus, M. (2025). Penerapan Algoritma Naive Bayes untuk Menganalisis Kesadaran Gen-Z terhadap Ancaman Cyber melalui CAPTCHA di Kota Mataram. <https://journalsaktipublishingcenter.org/>
- Arno, M., Saputra, Y., Fadillah, M. N., & Setiadi, D. (2014). Aplikasi Checklist Safety Berbasis Mobile Menggunakan Metode Job Safety Analysis (Jsa) Untuk Audit Kesehatan Dan Keselamatan Kerja (K3) Di Pt.Matahari Putra Prima,Tbk. *JURNAL SISTEM INFORMASI UNIVERSITAS SURYADARMA*, 8(2). <https://doi.org/10.35968/jsi.v8i2.740>

- Askar, N. F., Herawanti, D., Susilawati, & Pratomo, D. N. (2021). Sistem Pelaporan Berbasis Web Pada Pelaksanaan Kegiatan Keselamatan Dan Kesehatan Kerja (K3) Di Puskesmas. *Universitas Muhammadiyah Tangerang*, 6, 1–8. <https://doi.org/10.31000/jkft.v6i2.5615.g2884>
- Bidah, & Noor Islami, R. (2024). Penerapan Aplikasi Mobile E-Card Employee Menggunakan Ionic Framework (Studi Kasus : PT. Darma Henwa, Tbk- Acp). *Journal Information Technology Trends (JITRENDS)*, 1(02), 70–78. <https://doi.org/10.51817/jitrends.v1i2.21>
- Dewantara, R. A., Irwansyah, M. A., & Pratiwi, H. S. (2022). *Aplikasi Pelaporan Darurat Berbasis Android*. <https://doi.org/10.26418/jari.v1i1.53213>
- Dinh, N. T., & Hoang, V. T. (2023). Recent advances of Captcha security analysis: a short literature review. *Procedia Computer Science*, 218, 2550–2562. <https://doi.org/10.1016/j.procs.2023.01.229>
- Iqbal Sain, M., Asep Rizkiawan, M., Maulana Rahmat, M. A., & Sidik, M. (2025). Optimalisasi Pengalaman Pengguna: Redesign Ui/Ux Website Simakip Uhamka Dengan Metode Design Thinking. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(1). <https://doi.org/10.23960/jitet.v13i1.5479>
- Mirza, M. N., Lourinx, E., Nugroho, R. A., & Syaifudin, A. (2024). Perancangan Sistem Peringatan Darurat Bagi Penyandang Gangguan Pendengaran. *JKM (Jurnal Kesehatan Masyarakat) Cendekia Utama*, 12(1), 90. <https://doi.org/10.31596/jkm.v12i1.2015>
- Mouza Salsabila, A., Setyo Wahyuningsih, A., & Penerapan Aplikasi Pelaporan Bahaya Sebagai Upaya Pencegahan Kecelakaan Kerja, E. (2023). *Efektivitas Penerapan Aplikasi Pelaporan Bahaya Sebagai Upaya Pencegahan Kecelakaan Kerja Pada PT X*. 8(5). <https://doi.org/10.36418/syntax>
- Pratama, R. E., & Wahyuni, E. G. (2024). Perancangan Desain Antarmuka Pengguna Dan Pengalaman Pengguna Pada Aplikasi Yada. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 9(2), 841–851. <https://doi.org/10.29100/jupi.v9i2.4742>
- Purnawati, E., Syafa'at, A. Y., Febianto, D. D., Sari, D., & Adhimah, L. F. (2023). Perancangan Aplikasi Digital Bidang Keselamatan dan Kesehatan Kerja dengan User Interface Berbasis Android. *SISTEMASI*, 12(1), 257. <https://doi.org/10.32520/stmsi.v12i1.2446>
- Suhari, Faqih, A., & Basysyar, F. M. (n.d.). Sistem Informasi Kepegawaian Menggunakan Metode Agile Development di CV. *Jurnal Teknologi Dan Informasi*. <https://doi.org/10.34010/jati.v12i1>
- Sultan, M. (2021). Persepsi Karyawan Terhadap Sistem Pelaporan Kecelakaan Kerja Dan Potensi Bahaya Di Pertambangan Batubara Pt. Putra Kajang Kalimantan Timur. *Jurnal Sehat Mandiri*, 16(1), 18–28. <https://doi.org/10.33761/jsm.v16i1.339>
- Susanto, D., Wardana, R. A. C., Aditia, D. A., & Haryono, W. (2024). Pengembangan Aplikasi Berbasis Web untuk Monitoring dan pengelolaan Stok ATK dengan Notifikasi Otomatis dan Sistem

Barcode di Bank Mandiri, Commercial Banking 5&6. *Journal of Informatics and Business*, 2(3), 493–496. <https://doi.org/10.47233/jibs.v2i3.1873>

Yogama, C. D., Djunaidi, Z., & Rahmawati, F. F. (2022). *Implementasi Program Pelaporan Unsafe Action & Unsafe Condition Di Pt Xyz*.

Zulkarnain, M. N., Dzakwan, R. H., & Arrahman, T. (2025). *Survivorbox: Sistem Pemantauan Dan Peringatan Pekerja Berbasis Aiot Dengan Komunikasi Lora Untuk Mencegah Kecelakaan Di Industri Pertambangan*.

LAMPIRAN

Lampiran A: Gambar Dokumentasi Kegiatan Magang



Lampirann B: Surat Permohonan Magang Kampus



FAKULTAS
TEKNOLOGI INDUSTRI

Gedung KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
Jl. Kaliurang km 14,5 Yogyakarta 55584
T. (0274) 898444 ext. 4100, 4101
F. (0274) 895007
E. fti@uii.ac.id
W. fti.uii.ac.id

Nomor : 142/Kaprodi.IF.S1/10/Prodi.IF.S1/VII/2024
Lamp. : 1 berkas portofolio
1 lembar form keputusan permohonan magang
Hal : Permohonan magang

Kepada Yth.

Pimpinan PT Nocola IOT Solution
Di 117 Ir. Hj Djuanda St, Central Cilacap, 53223, Central Java, Indonesia

Assalamu' alaykum wr wb.

Berdasarkan kurikulum yang berlaku di Program Studi Informatika – Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia, mahasiswa dapat mengambil jalur Magang mulai semester 7 dan 8 sebagai bagian dari Tugas Akhir untuk menyelesaikan studinya.

Berkenaan dengan itu, kami mohon kesediaan Bapak/Ibu menerima mahasiswa berikut untuk dapat melaksanakan magang di institusi yang Bapak/Ibu pimpin:

NIM : 21523031
Nama : Ikhwan Alfi Mushaffa Asadilah

Portofolio mahasiswa terlampir.

Apabila memungkinkan, kami berharap mahasiswa dapat memulai magang pada **15 September 2024 untuk masa sedikitnya 6 bulan**. Kami mengikuti sepenuhnya prosedur penerimaan dan pelaksanaan magang yang berlaku di institusi Bapak/Ibu.

Kami lampirkan pula form keputusan permohonan magang yang dapat digunakan untuk menyampaikan persetujuan atau penolakan terhadap permohonan kami ini. Meski demikian, surat dapat juga menggunakan format dari institusi Bapak/Ibu.

Demikian permohonan kami. Atas perhatian dan kerja sama Bapak/Ibu, kami ucapkan terima kasih.

Wassalamu' alaykum wr wb.

Yogyakarta, 25 Juli 2024
Ketua Program Studi Informatika – Program Sarjana


Thomas Hatta Fudholi

Lampiran C: Surat Keputusan Pemohonan Magang

FORM KEPUTUSAN PERMOHONAN MAGANG

Kepada

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri Universitas Islam Indonesia
Di Yogyakarta

Referensi : Ya

Kami Pimpinan PT Nocola IOT Solution

menyatakan bahwa permohonan magang mahasiswa

NIM : 21523031

Nama : Ikhwan Alfi Mushaffa Asadilah


DISETUJUI/DITOLAK (*coret yang tidak perlu).

Tanggal mulai magang: 15 September 2024

dengan masa magang: 6 bulan.

Demikian keputusan kami.

Tanggal: 31 Juli 2024


(M. Kover Susastra)

Tanda tangan dan nama terang
Cap stempel institusi