

BAB III

METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah *Research and Development* (R&D), yang digunakan untuk merancang, membangun, dan mengevaluasi dalam Rancang Bangun Helm Pengaman Pintar untuk Konstruksi Berbasis IoT dengan Integrasi Platform ERP Odoo (Borg dan Gall, 1983;2003). Model penelitian ini menurut Borg dan Gall, (1983) terdiri dari 10 tahapan, diantaranya Penelitian dan Pengumpulan Data, Perencanaan, Pengembangan Produk Awal, Uji Coba Terbatas, Penyempurnaan Produk Awal, Uji Coba Sistem, Evaluasi Sistem Produk, Pengujian Lapangan Utama, Revisi Produk Akhir, dan Diseminasi dan Implementasi. Tahapan pendekatan R&D tersebut ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Tahapan Metode R&D

Pada Gambar 3.1 diatas menunjukkan tahapan metode *Research and Development* (R&D) dengan 10 tahapan. Namun, dalam penelitian ini hanya digunakan sampai tahap ke-7 (Evaluasi Sistem Produk) sesuai dengan ruang lingkup penelitian yang telah ditetapkan. Pembatasan ini dilakukan karena penelitian difokuskan pada tahap pengembangan prototipe hingga evaluasi setelah uji coba sistem terbatas, tanpa melanjutkan ke tahap diseminasi dan implementasi massal.

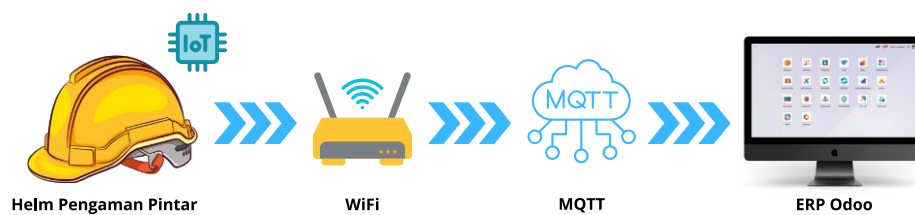
Metode ini dipilih karena mampu mendukung proses pengembangan produk secara sistematis dan terstruktur, mulai dari penelitian dan pengumpulan data hingga evaluasi sistem produk. Metode R&D ini juga relevan dalam pengembangan sistem teknologi serta perangkat berbasis IoT, karena memberikan fleksibilitas dalam menyesuaikan desain dengan tantangan kompleks di lapangan. Fokus utama terletak pada menghasilkan produk fungsional, sementara proses evaluasi dilakukan pada setiap tahap guna menjamin efektivitas dan kesesuaian produk dengan kebutuhan pengguna. Adapun penjelasan dari masing-masing tahapan dalam metode R&D adalah sebagai berikut.

3.1 Penelitian dan Pengumpulan Data

Pada tahap ini, dilakukan penelitian dan pengumpulan data untuk merancang solusi pengelolaan sumber daya dan keselamatan kerja di sektor konstruksi secara *real-time*. Tantangan utama meliputi tingginya angka kecelakaan kerja akibat rendahnya disiplin terhadap penerapan K3, minimnya penggunaan APD, dan sistem manajemen proyek yang terpisah. Helm keselamatan konvensional hanya menyediakan perlindungan fisik tanpa fitur pelacakan lokasi atau deteksi penggunaan. Solusi yang diusulkan adalah helm pintar berbasis IoT dengan fitur pelacakan lokasi dan deteksi penggunaan helm. Data dikirimkan secara *real-time* ke ERP Odoo. Implementasi melibatkan uji alat, integrasi helm pengaman pintar dengan ERP Odoo, dan skenario skala laboratorium dan lingkungan terbatas konstruksi. Evaluasi dilakukan untuk menilai efektivitas monitoring, serta kontribusi sistem terhadap pengurangan risiko kecelakaan kerja.

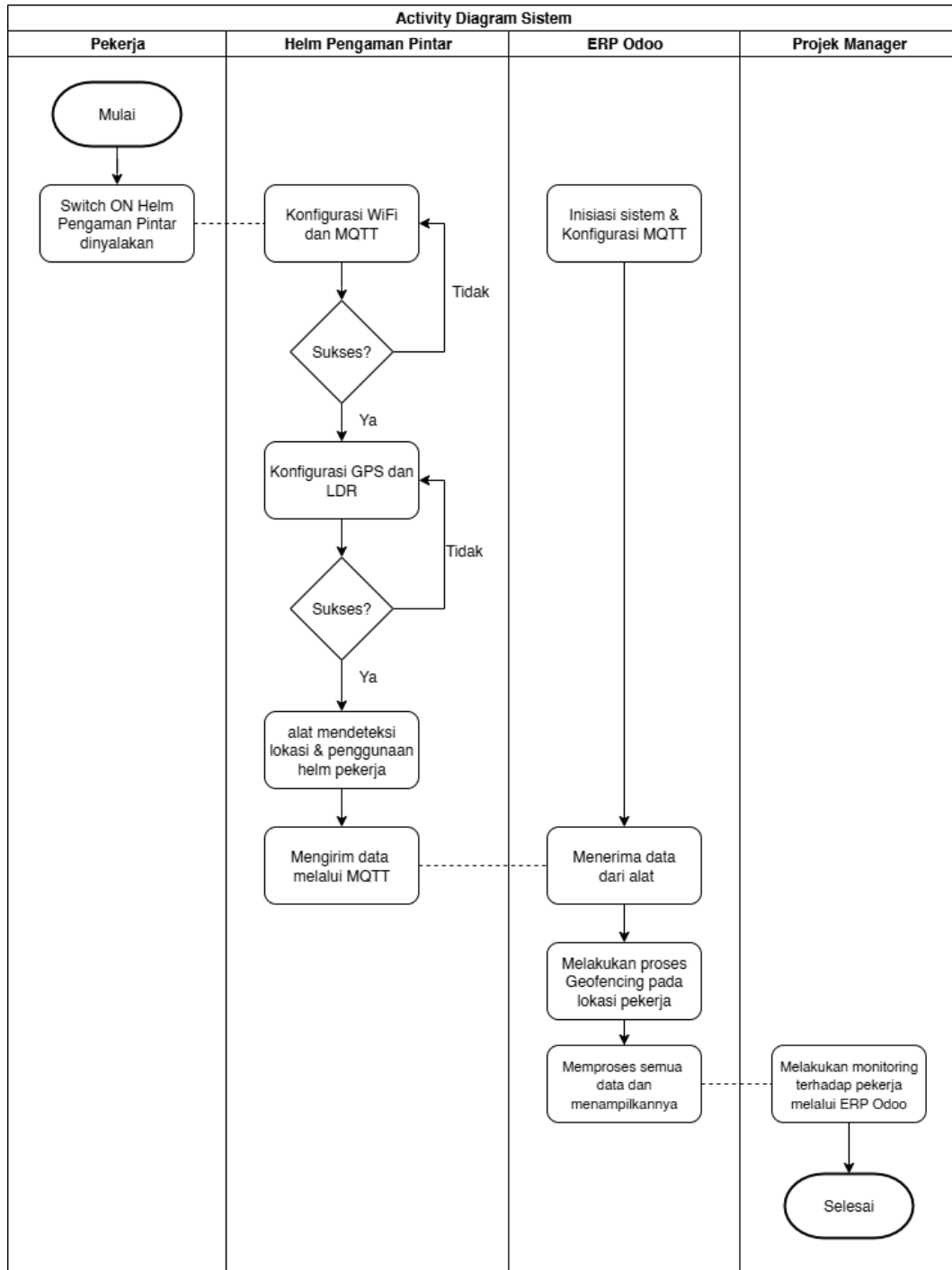
3.2 Perencanaan

Dalam tahap kedua, terdapat upaya perencanaan serta perancangan sistem yang berfokus pada desain perancangan. Desain ini mencakup prototype alat hingga perangkat lunak ERP Odoo yang akan digunakan, untuk memberikan gambaran yang lebih komprehensif. Peneliti menyertakan diagram arsitektur sistem secara menyeluruh, dimulai dari prototype helm pengaman pintar mengirim data melalui MQTT, hingga penyajian data monitoring pada ERP Odoo. Berikut adalah rancangan diagram arsitektur sistem yang telah dibuat.



Gambar 3. 2 Diagram Arsitektur Sistem

Pada Gambar 3.2 terdapat diagram arsitektur sistem yang menggambarkan alur kerja secara keseluruhan dari proses prototype alat helm pengaman pintar berbasis IoT yang dapat mengirimkan lokasi serta mendeteksi pekerja memakai helm sebagai APD atau tidak, kemudian data tersebut dikirim melalui MQTT ke platform ERP Odoo untuk divisualisasikan serta menyajikan data monitoring. Sistem ini dirancang untuk bekerja secara *real-time* dan terintegrasi.



Gambar 3. 3 Activity Diagram Sistem

Pada Gambar 3.3 menunjukkan activity diagram monitoring helm pengaman pintar yang terintegrasi dengan ERP Odoo untuk memantau lokasi dan penggunaan helm pekerja konstruksi. Diagram ini terdiri dari empat swimlane, yaitu Pekerja, Helm Pengaman Pintar, ERP Odoo, dan Project Manager, yang

masing-masing merepresentasikan peran atau komponen dalam sistem. Proses dimulai ketika pekerja menyalakan helm pengaman pintar (Switch ON). Helm kemudian melakukan konfigurasi WiFi dan MQTT untuk memastikan konektivitas dengan sistem ERP Odoo. Apabila konfigurasi ini berhasil, proses dilanjutkan ke tahap konfigurasi GPS dan sensor LDR. GPS digunakan untuk menentukan posisi pekerja, sedangkan sensor LDR digunakan untuk mendeteksi status penggunaan helm. Jika konfigurasi GPS dan LDR berhasil, perangkat akan mulai mendeteksi lokasi pekerja serta status pemakaian helm (dipakai atau tidak). Data hasil deteksi ini dikirimkan ke ERP Odoo melalui protokol MQTT. ERP Odoo akan menerima data tersebut, kemudian melakukan proses geofencing untuk memverifikasi apakah pekerja berada di dalam atau di luar area kerja yang telah ditentukan. Selanjutnya, sistem memproses seluruh data dan menampilkannya pada antarmuka ERP. Pada tahap akhir, Project Manager dapat melakukan monitoring pekerja secara *real-time* melalui ERP Odoo. Proses ini berakhir ketika seluruh data telah diproses dan ditampilkan untuk pengambilan keputusan terkait keselamatan dan kepatuhan pekerja terhadap prosedur K3.

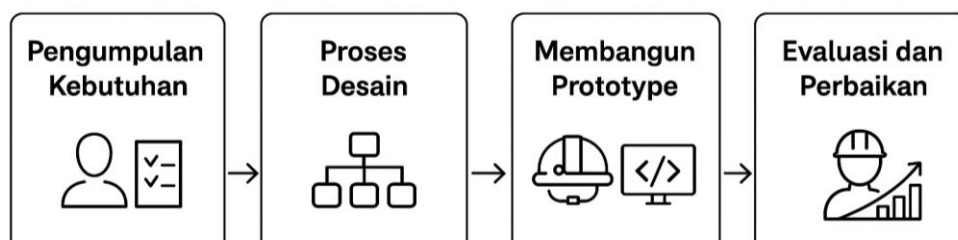
3.3 Pengembangan Produk Awal

Pada tahap ini berisi dengan metode pengembangan yang digunakan serta menjelaskan tahap-tahap pengembangan untuk prototipe helm pengaman pintar dan pengembangan perangkat lunak ERP Odoo. Tahap ini bertujuan untuk menghasilkan prototipe awal yang sesuai dengan rancangan.

3.3.1 Metode Pengembangan Helm Pengaman Pintar

Dalam penelitian ini, pengembangan alat menggunakan metode Prototype sebagai pendekatan utama. Metode Prototype dipilih karena penelitian ini didasarkan pada karakteristik pengembangan alat yang memungkinkan proses pengembangan yang bersifat iteratif, di mana prototype awal dapat segera diuji, dievaluasi, dan disempurnakan secara bertahap. Hal ini sangat penting terutama dalam pengembangan alat yang menggabungkan komponen perangkat keras dan perangkat lunak, karena memungkinkan identifikasi dan perbaikan kesalahan sejak tahap awal (Aziz dkk., 2024).

METODE PROTOTYPE



Gambar 3. 4 Tahapan Metode Prototype

Selain itu, metode prototype juga dinilai lebih fleksibel dan efisien dalam hal waktu dan biaya, karena fokus pada pembuatan model kerja awal yang dapat terus dikembangkan tanpa harus menunggu rancangan akhir yang sempurna. Dengan pendekatan ini, hasil akhir dari alat yang dikembangkan diharapkan lebih tepat guna, efektif, dan sesuai dengan kebutuhan pengguna di lapangan. Alur proses metode ini dimulai dari Pengumpulan kebutuhan, Proses desain, Membangun prototype, dan yang terakhir adalah Evaluasi dan perbaikan. Keempat proses ini saling terhubung, berikut merupakan penjelasan dari rangkaian proses secara rinci pada penelitian ini.

1. Pengumpulan Kebutuhan

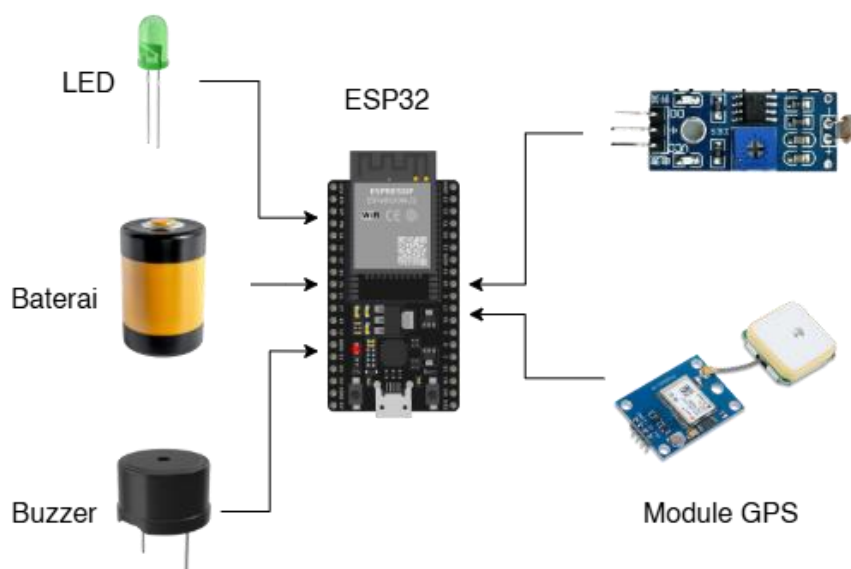
Pada tahap pertama ini, dilakukan proses yang bertujuan untuk mengidentifikasi kebutuhan pengguna dan menentukan spesifikasi teknis alat yang akan dikembangkan. Dimulai dengan studi literatur yang mencakup kajian terhadap penelitian terdahulu, standar keselamatan kerja (K3) di sektor konstruksi, dan pemanfaatan teknologi *Internet of Things* (IoT) dalam sistem monitoring keselamatan pekerja. Studi ini menitikberatkan pada penerapan penentuan lokasi pekerja, serta deteksi penggunaan helm, dengan integrasi data secara *real-time* ke dalam sistem ERP Odoo.

Hasil dari studi literatur tersebut menjadi dasar untuk merumuskan analisis kebutuhan fungsional dan non-fungsional sistem. Kebutuhan fungsional meliputi kemampuan sistem dalam mendeteksi posisi pekerja secara *real-time*, memverifikasi status penggunaan helm (dipakai atau tidak dipakai), mengirimkan data ke ERP Odoo, dan menampilkan informasi tersebut pada halaman monitoring. Sementara itu, kebutuhan non-fungsional mencakup ketahanan perangkat terhadap

kondisi lapangan yang ekstrem, efisiensi penggunaan daya baterai agar dapat beroperasi dalam durasi kerja yang panjang, stabilitas konektivitas data, serta kemudahan penggunaan antarmuka oleh manajer proyek untuk menunjang proses pengambilan keputusan yang cepat dan tepat.

2. Proses Desain

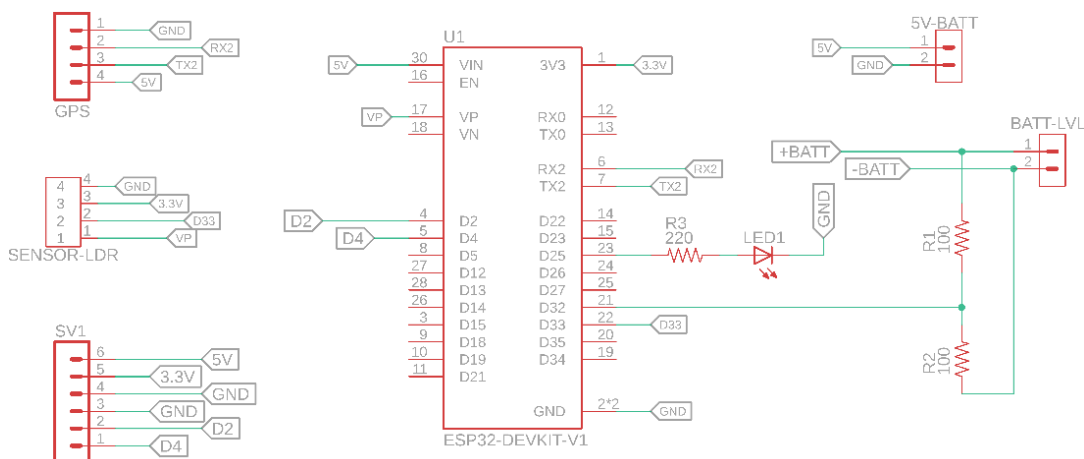
Desain dari pengembangan alat dibuat pada tahap ini, setelah menentukan kebutuhan perangkat keras dan kebutuhan perangkat lunak. Pembuatan desain disesuaikan dengan kebutuhan pengembangan alat helm pengaman pintar dengan mempertimbangkan proses dari awal hingga akhir. Pada penelitian ini, dibuat Diagram Blok, Diagram Arsitektur, Wiring Diagram, Flowchart Alat, Desain PCB, hingga Desain Alat. Berikut desain pada pengembangan helm pengaman pintar berbasis IoT secara lebih rinci dalam penelitian ini.



Gambar 3. 5 Diagram Arsitektur Alat

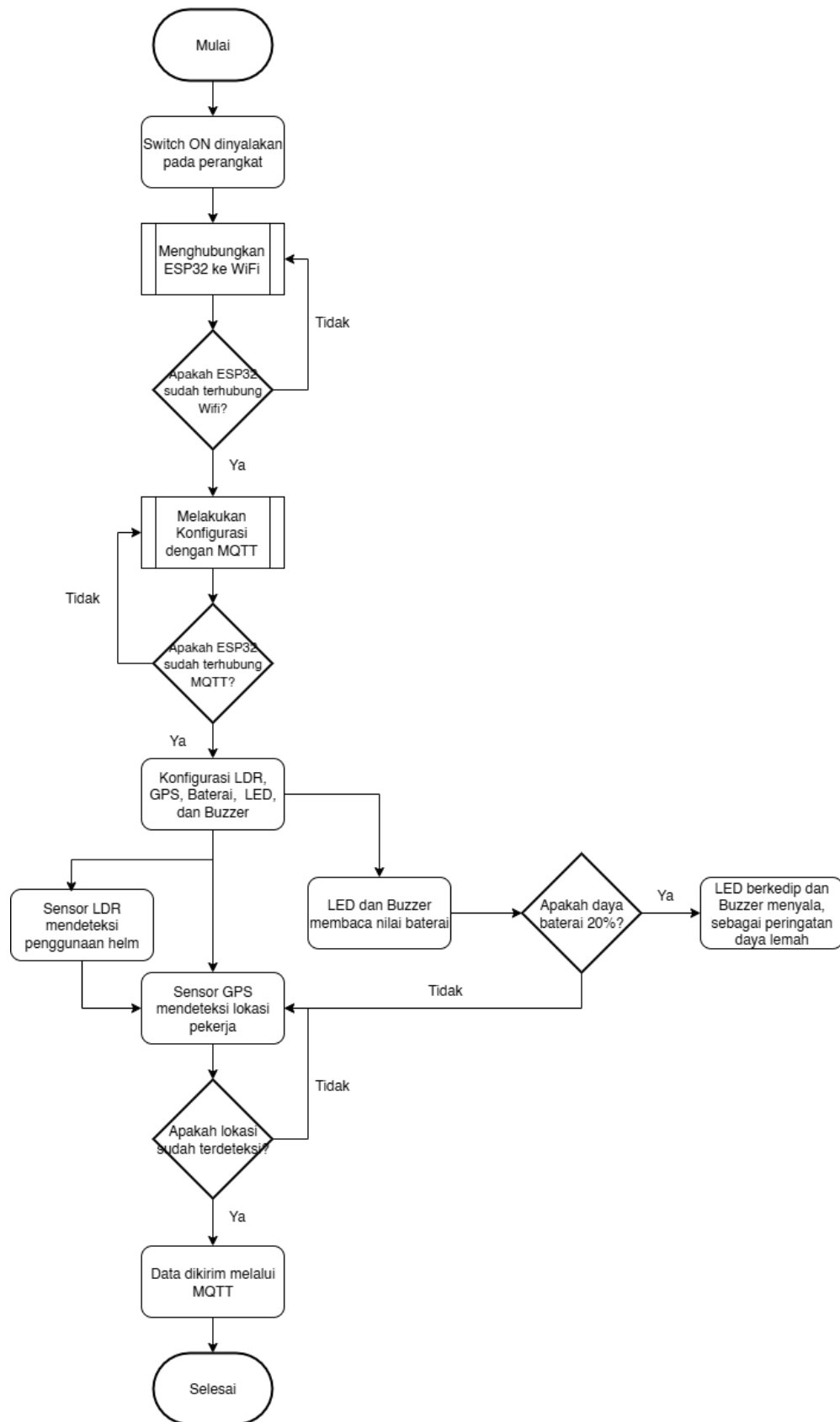
Pada Gambar 3.5 menampilkan diagram arsitektur helm pengaman pintar berbasis IoT yang menggambarkan alur kerja dan komponen utama yang digunakan. Sistem dimulai dari helm yang dilengkapi dengan mikrokontroler ESP32 sebagai inti pengolahan data. ESP32 menerima daya dari baterai dan terhubung dengan beberapa sensor utama, yaitu GPS Neo Ublox M8L untuk pelacakan lokasi pekerja secara *real-time*, sensor LDR (Light Dependent Resistor) untuk mendeteksi penggunaan helm, LED dan Buzzer untuk indikator penggunaan

daya baterai. Data yang diperoleh dari GPS dan LDR diproses oleh ESP32 dan dikirimkan melalui jaringan WiFi atau router ke server ERP Odoo.



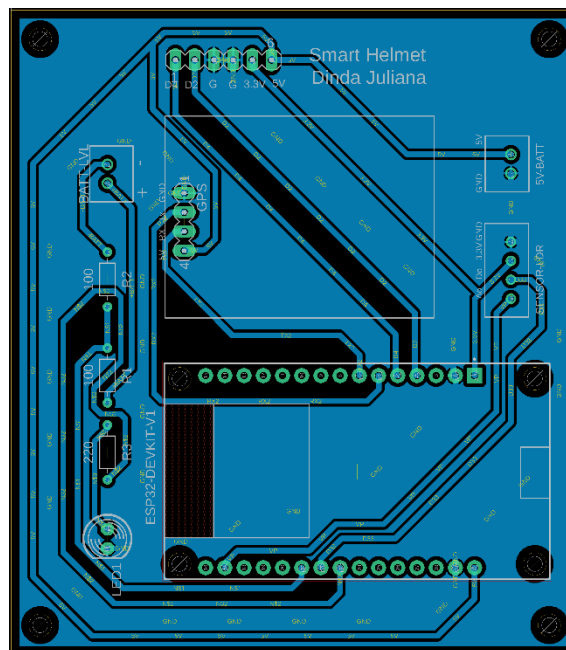
Gambar 3. 6 Schematic Printed Circuit Board (PCB)

Pada Gambar 3.6 menampilkan schematic PCB yang memperlihatkan hubungan antar komponen pada sistem helm pengaman pintar berbasis IoT, di mana ESP32 berperan sebagai pusat kendali untuk memproses data dan mengirimkannya melalui WiFi menggunakan protokol MQTT. Sensor LDR digunakan untuk mendeteksi penggunaan helm, sementara modul GPS Neo-M8L memberikan data koordinat lokasi pekerja secara *real-time*. LED indikator dan buzzer berfungsi sebagai peringatan visual dan audio, termasuk saat kapasitas baterai rendah yang dipantau melalui rangkaian pembaca tegangan. Seluruh komponen mendapat pasokan daya dari baterai lithium yang diatur melalui modul pengisian dan pengatur tegangan, sehingga memastikan perangkat dapat bekerja stabil mulai dari pengambilan data hingga pengiriman informasi ke sistem ERP Odoo.



Gambar 3. 7 Flowchart Alat

Pada Gambar 3.7 menunjukkan flowchart alat helm pengaman pintar berbasis IoT yang menggambarkan alur kerja perangkat helm pengaman pintar berbasis IoT sejak proses inisialisasi hingga pengiriman data ke server. Alur dimulai ketika sakelar (switch) pada perangkat diaktifkan, yang menandai penyalaan sistem dan memulai proses penghubungan mikrokontroler ESP32 ke jaringan WiFi. Setelah koneksi WiFi berhasil, perangkat melakukan konfigurasi koneksi dengan protokol MQTT untuk memastikan komunikasi data dapat berjalan. Tahap berikutnya adalah konfigurasi komponen utama, meliputi sensor LDR untuk mendeteksi kondisi pemakaian helm, modul GPS untuk menentukan lokasi pekerja, serta modul indikator berupa LED dan buzzer. Sistem juga melakukan pembacaan daya baterai, di mana jika kapasitas baterai berada di bawah 20%, LED akan berkedip dan buzzer akan berbunyi sebagai peringatan daya rendah. Sensor LDR akan memantau intensitas cahaya guna menentukan apakah helm sedang digunakan atau tidak, sedangkan modul GPS memantau koordinat lokasi pekerja secara *real-time*. Jika lokasi telah terdeteksi, data kondisi penggunaan helm dan lokasi pekerja akan dikirimkan ke server melalui protokol MQTT untuk kemudian diolah dan ditampilkan pada sistem ERP Odoo.



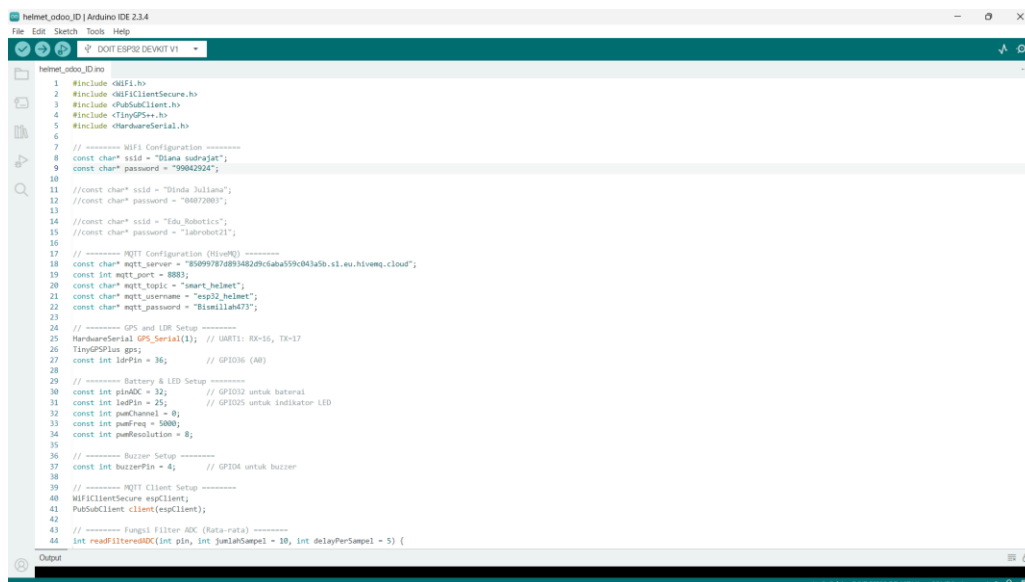
Gambar 3. 8 Desain *Printed Circuit Board* (PCB)

Pada Gambar 3.8 menampilkan desain PCB (Printed Circuit Board) yang akan digunakan sebagai board penghubung seluruh komponen dan sensor pada penelitian ini.



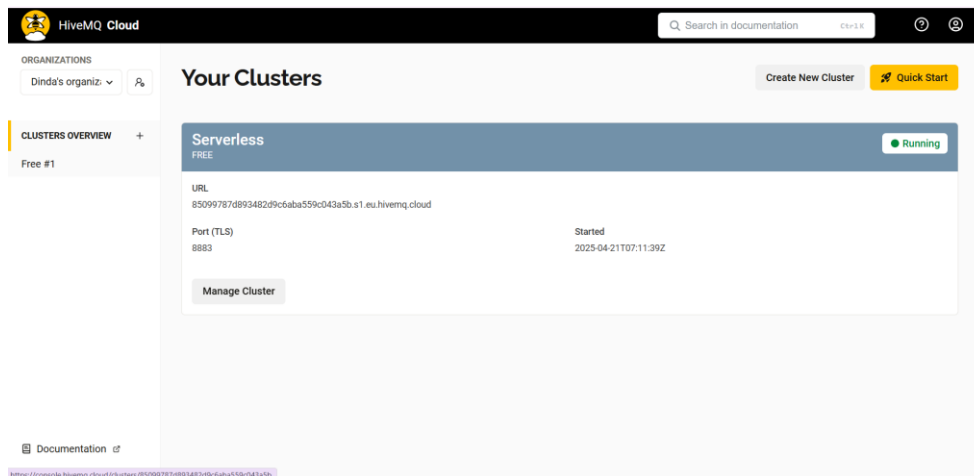
Gambar 3. 9 Desain Alat

Pada Gambar 3.9 menampilkan desain alat helm pengaman pintar berbasis IoT yang dirancang untuk mendukung implementasi keselamatan dan kesehatan kerja (K3) pada proyek konstruksi. Desain ini memperlihatkan struktur helm pelindung yang dimodifikasi dengan penambahan sebuah kotak casing di bagian belakang, berfungsi sebagai wadah untuk komponen elektronik utama seperti mikrokontroler ESP32, modul GPS, sensor LDR, baterai, serta rangkaian pendukung lainnya. Kotak casing yang terbuat dari akrilik tersebut dirancang dengan dimensi yang lebih lebar agar mampu menampung komponen secara aman, serta dilengkapi lubang akses untuk port pengisian daya dan sakelar utama perangkat. Selain itu, terdapat kebutuhan perangkat lunak untuk pengembangan alat yang akan ditunjukkan pada Gambar 3.10 dan Gambar 3.11.



Gambar 3. 10 Sketch Arduino IDE

Pada Gambar 3.10 ditunjukkan perangkat lunak Arduino IDE yang digunakan untuk menulis, mengkompile, dan mengupload sketch program pada ESP32 sebagai kebutuhan pengembangan alat helm pengaman pintar berbasis IoT pada penelitian ini.



Gambar 3. 11 HiveMQ Cloud

Pada Gambar 3.11 ditunjukkan perangkat lunak layanan cloud untuk protokol MQTT (Message Queuing Telemetry Transport). Dalam penelitian ini, HiveMQ Cloud digunakan untuk mentransfer data sensor secara *real-time* dari ESP32 ke server, sebelum data dikirim ke ERP Odoo.

3. Membangun Prototype

Pada tahap ini, komponen perangkat keras mulai dirakit sesuai skema rangkaian yang telah dirancang. Sensor dipasang pada helm dengan memperhatikan posisi yang optimal untuk akurasi deteksi. Mikrokontroler diprogram untuk mengelola data dari sensor, memprosesnya, dan mengirimkan hasilnya ke ERP Odoo melalui protokol komunikasi IoT yang dipilih. Integrasi dengan Odoo dilakukan menggunakan MQTT, sehingga data dapat ditampilkan secara *real-time* di modul monitoring pada ERP Odoo. Prototype yang dihasilkan pada tahap ini dapat mengalami perubahan sesuai hasil evaluasi.

4. Evaluasi dan Perbaikan

Prototype yang telah selesai dibangun kemudian diuji dengan skenario yang memungkinkan terjadi di lingkungan kerja nyata untuk memastikan semua fungsi berjalan sesuai dengan spesifikasi. Pengujian meliputi uji fungsionalitas alat serta uji ketahanan daya pada alat. Data hasil pengujian dianalisis untuk mengidentifikasi kekurangan atau potensi perbaikan. Masukan juga dijadikan acuan untuk modifikasi, baik pada perangkat keras (misalnya posisi sensor atau jenis modul komunikasi) maupun perangkat lunak (misalnya logika pemrosesan data). Proses evaluasi dan perbaikan dilakukan hingga alat mencapai performa optimal.

3.3.2 Metode Pengembangan ERP Odoo

Pengembangan ERP Odoo dalam penelitian ini menggunakan pendekatan Agile Software Development. Metode ini dipilih karena fleksibilitasnya dalam menyesuaikan perubahan kebutuhan secara cepat, efisien, dan terstruktur. Agile merupakan metodologi pengembangan perangkat lunak yang menekankan pada proses iteratif dan inkremental, di mana sistem dibangun melalui serangkaian siklus pengembangan (sprint) yang diikuti dengan evaluasi dan perbaikan secara berkala. Prinsip utama Agile berfokus pada kolaborasi aktif antar tim, penyelesaian masalah secara adaptif, serta pencapaian hasil yang optimal melalui komunikasi yang intensif (Raharja dkk., 2024).

Dalam praktiknya, terdapat beberapa model populer dalam metode Agile, seperti Extreme Programming (XP) dan Scrum. Model XP sangat sesuai untuk proyek yang kebutuhannya belum sepenuhnya terdefinisi sejak awal, karena

mengedepankan komunikasi berkelanjutan, pengujian secara konsisten, dan integrasi kode rutin. Sementara itu, Scrum berfokus pada pembagian kerja dalam sprint terstruktur dengan tujuan meningkatkan kecepatan pengembangan (Larasati dkk., 2021). Dengan penerapan metode Agile, pengembangan ERP Odoo pada penelitian ini dapat dilakukan secara terarah, adaptif, dan efisien, terutama dalam membangun perangkat lunak yang berfungsi menerima data serta memvisualisasikan informasi monitoring dari helm pengaman pintar berbasis IoT.



Gambar 3. 12 Tahapan Metode Agile
(sumber : BINAR)

Pada Gambar 3.12 ditunjukkan bahwa terdapat enam tahapan dalam metode Agile yang diterapkan dalam pengembangan perangkat lunak, dimulai dari Requirements, Design, Development, Testing, Deployment, dan Review. Keenam tahapan ini saling terhubung dan berkaitan, berikut merupakan penjelasan dari rangkaian proses secara rinci pengembangan ERP Odoo pada penelitian ini.

1. Requirements

Tahap awal pengembangan sistem difokuskan pada proses identifikasi kebutuhan yang diperlukan dalam membangun ERP Odoo yang terintegrasi dengan helm pengaman pintar berbasis IoT, baik dari sisi perangkat lunak maupun fungsi-fungsi yang harus dimiliki oleh sistem. Tahap ini dilakukan untuk memastikan seluruh kebutuhan teknis dan non-teknis telah terpenuhi sebelum proses implementasi dimulai. Kebutuhan tersebut meliputi instalasi *package* ERP Odoo,

penggunaan berbagai *library* pendukung, serta integrasi layanan pihak ketiga yang dapat meningkatkan kinerja sistem secara keseluruhan.

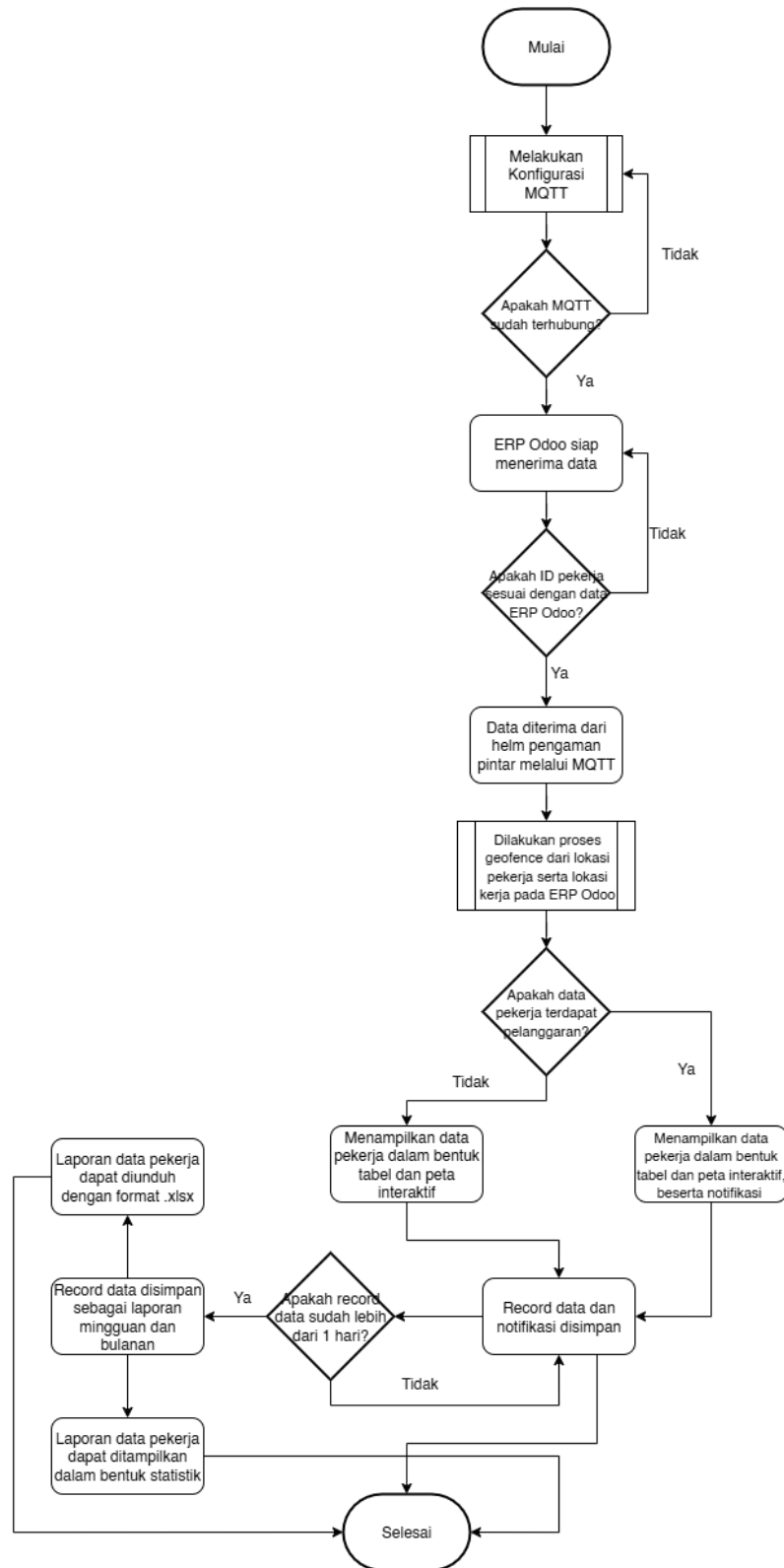
Dalam penelitian ini, pengembangan ERP Odoo menggunakan Odoo versi 17 Community Edition yang diintegrasikan dengan Python 3.10 dan menggunakan basis data bawaan Odoo, yaitu PostgreSQL, yang telah saling terintegrasi secara *default*. Pemilihan Odoo versi 17 didasarkan pada pertimbangan bahwa versi ini merupakan salah satu rilis terbaru yang kompatibel untuk pengembangan lanjutan, khususnya untuk pembuatan *add-ons* kustom yang terhubung dengan perangkat IoT. Selain itu, digunakan layanan pihak ketiga seperti Google Maps API untuk mendukung implementasi fitur *geofencing* dan *geocoding* yang diperlukan dalam sistem ini.

Integrasi data dilakukan melalui protokol MQTT, di mana data dari helm pintar dikirim ke *broker* kemudian diteruskan ke Odoo melalui *script listener* berbasis Python yang memanfaatkan XML-RPC API. Fungsi inti sistem mencakup pencatatan status helm (“dipakai” atau “tidak dipakai”), pelacakan lokasi pekerja, pengiriman notifikasi otomatis ketika pekerja tidak menggunakan helm atau berada di luar area proyek, serta visualisasi data secara *real-time* dalam bentuk tabel, grafik, dan peta interaktif pada modul ERP Odoo.

Selain fungsi inti tersebut, sistem ini juga memiliki fitur pendukung untuk mempermudah manajemen proyek. Data pekerja dan proyek diambil secara terintegrasi dari modul bawaan Odoo dengan penyesuaian (*customization*) sesuai kebutuhan penelitian. Pada *add-ons* baru bernama Smart Helmet, data pekerja diambil dari modul *Employees* dengan filter khusus untuk *Department* “Konstruksi” sehingga pengelolaan data menjadi lebih terstruktur dan efisien. Sementara itu, pada modul *Project* dilakukan penambahan fitur pemilihan lokasi proyek secara langsung melalui tampilan peta (*map view*) yang dilengkapi fungsi *geocoding* untuk penyimpanan data koordinat. Fitur ini mendukung mekanisme *geofencing* pada *add-ons* Smart Helmet untuk mendeteksi apakah pekerja berada di dalam atau di luar area kerja. Selain itu, sistem dilengkapi dengan fitur ekspor laporan kinerja pekerja dalam format .xlsx sehingga memudahkan pihak manajemen proyek dalam proses dokumentasi dan pelaporan.

2. Design

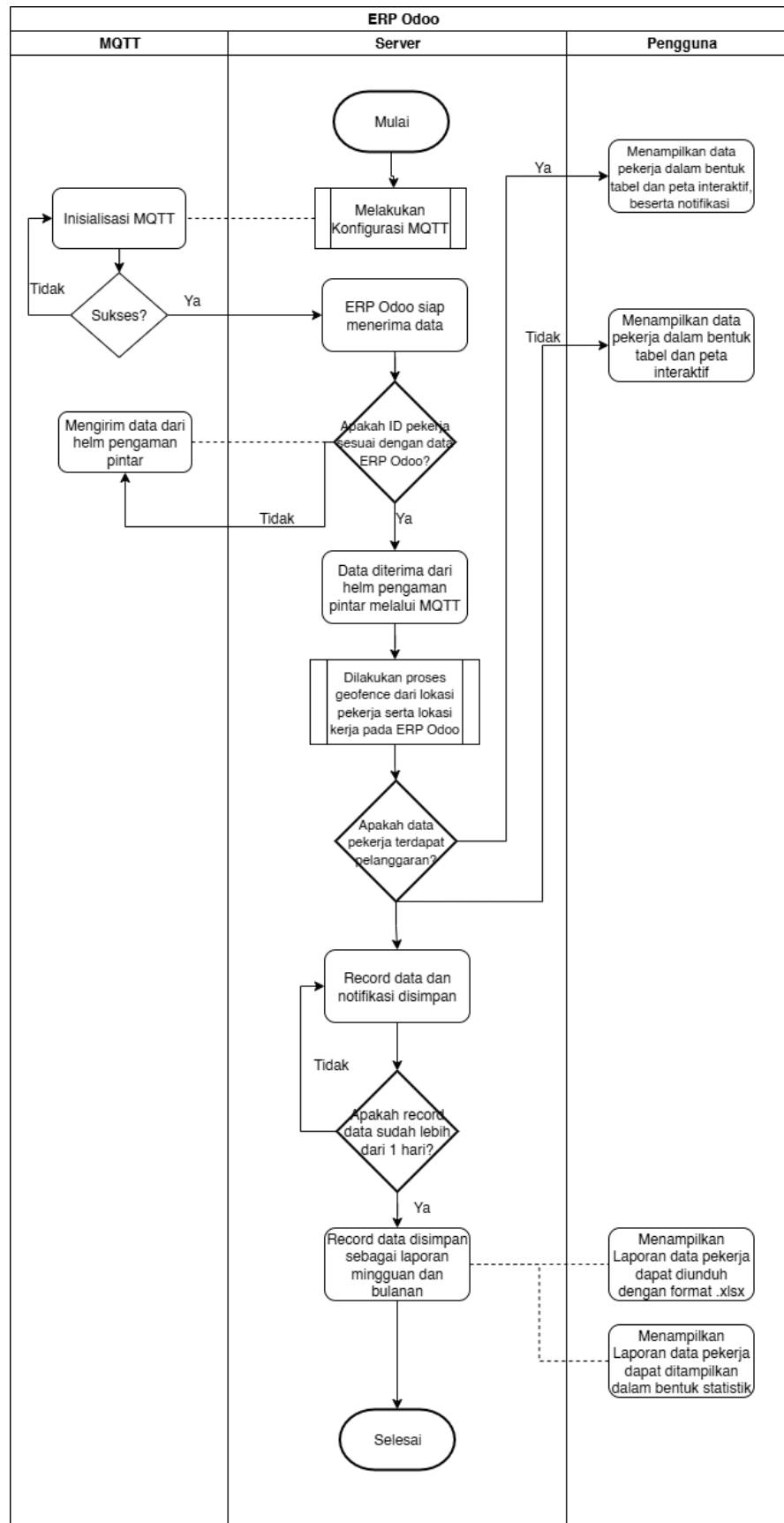
Pada tahap kedua, tahapan desain dilakukan untuk merumuskan rancangan sistem yang menjadi acuan dalam proses pengembangan. Desain ini befokus pada perancangan sistem ERP Odoo yang terintegrasi dengan helm pengaman pintar berbasis IoT, memenuhi kebutuhan fungsional, serta memberikan kemudahan bagi pengguna. Pada tahap ini menggambarkan alur proses sistem secara jelas sebelum masuk ke tahap implementasi. Rancangan sistem meliputi tiga aspek utama. Pertama, Flowchart Odoo yang menggambarkan alur proses mulai dari penerimaan data dari broker MQTT, pengolahan dan penyimpanan data, hingga penyajian informasi melalui antarmuka pengguna, ditunjukkan pada Gambar 3.14.



Gambar 3. 13 Flowchart ERP Odoo

Pada Gambar 3.13 menunjukkan flowchart proses kerja sistem ERP Odoo yang terintegrasi dengan helm pengaman pintar berbasis IoT melalui protokol komunikasi MQTT. Alur dimulai dengan proses konfigurasi koneksi MQTT pada sisi server Odoo untuk memastikan bahwa sistem dapat berkomunikasi dengan perangkat IoT secara *real-time*. Setelah konfigurasi dilakukan, sistem akan memverifikasi status koneksi MQTT sebelum melanjutkan ke tahap penerimaan data. Jika koneksi MQTT berhasil, ERP Odoo menyiapkan modul penerimaan data dan memeriksa kecocokan ID pekerja yang diterima dengan data master pekerja yang telah tersimpan di dalam sistem ERP Odoo. Apabila ID pekerja sesuai, data dari helm pengaman pintar yang mencakup informasi penggunaan helm dan koordinat lokasi pekerja akan diterima melalui MQTT. Selanjutnya, sistem ERP Odoo melakukan proses geofencing, yaitu membandingkan koordinat lokasi pekerja dengan lokasi proyek yang telah ditentukan di sistem untuk mendeteksi potensi pelanggaran seperti pekerja yang berada di luar area kerja atau tidak menggunakan helm. Jika pelanggaran terdeteksi, data pekerja akan ditampilkan dalam bentuk tabel dan peta interaktif, dilengkapi dengan notifikasi peringatan. Jika tidak ada pelanggaran, data tetap ditampilkan dalam tabel dan peta interaktif tanpa notifikasi tambahan. Data pekerja yang diterima dapat diunduh dalam format .xlsx, ditampilkan dalam bentuk statistik, atau disimpan sebagai catatan laporan harian, mingguan, dan bulanan. Sistem juga melakukan pengecekan apakah record data sudah lebih dari satu hari; jika ya, maka data akan diarsipkan untuk keperluan pelaporan jangka panjang. Dengan alur ini, flowchart menggambarkan bagaimana ERP Odoo berfungsi sebagai pusat pengolahan, penyimpanan, dan penyajian data keselamatan kerja secara efisien dan terstruktur.

Kedua, Activity Diagram yang menunjukkan interaksi broker MQTT, server Odoo, dan pengguna, beserta proses validasi status helm, pengecekan lokasi pekerja terhadap radius area proyek, dan pengiriman notifikasi otomatis, ditunjukkan pada Gambar 3.14.

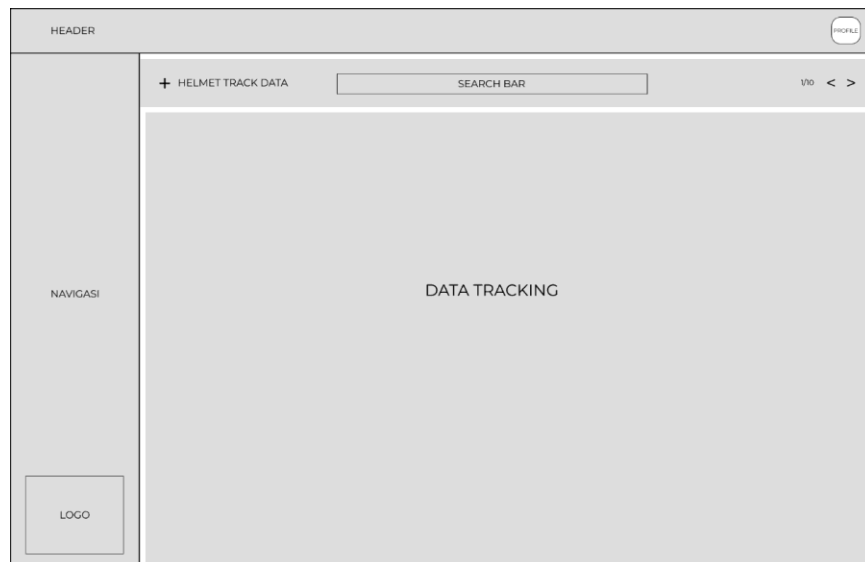


Gambar 3. 14 Activity Diagram ERP Odoo

Pada Gambar 3.14 menampilkan activity diagram ERP Odoo yang menggambarkan alur aktivitas sistem dalam menerima, memproses, dan menampilkan data dari helm pengaman pintar berbasis IoT melalui protokol MQTT. Diagram ini dibagi menjadi tiga swimlane utama, yaitu MQTT, Server Odoo, dan Pengguna, yang menunjukkan pembagian tanggung jawab proses pada masing-masing pihak. Alur dimulai pada swimlane MQTT dengan proses inisialisasi koneksi. Setelah itu, sistem melakukan pengecekan konektivitas untuk memastikan apakah proses inisialisasi berhasil. Jika berhasil, helm pengaman pintar akan mengirimkan data ke server ERP Odoo melalui MQTT. Pada swimlane Server Odoo, proses dimulai dengan konfigurasi MQTT. Setelah koneksi terhubung, server ERP Odoo akan mempersiapkan diri untuk menerima data. Tahap berikutnya adalah pengecekan kesesuaian ID pekerja dengan database di ERP Odoo. Jika ID sesuai, data dari helm pengaman pintar yang mencakup informasi penggunaan helm dan lokasi pekerja akan diterima. Selanjutnya, sistem melakukan proses geofencing, yaitu membandingkan lokasi pekerja dengan lokasi proyek yang sudah tersimpan di sistem ERP Odoo. Apabila terdeteksi pelanggaran, data akan disimpan sebagai record beserta notifikasi. Jika tidak ada pelanggaran, data tetap disimpan tanpa notifikasi tambahan. Setelah itu, dilakukan pengecekan apakah record data sudah berusia lebih dari satu hari. Jika iya, data akan disimpan sebagai laporan mingguan dan bulanan. Pada tahap akhir, data yang tersimpan dapat diunduh oleh pengguna dalam format .xlsx, ditampilkan dalam bentuk tabel dan peta interaktif, serta disajikan dalam format statistik untuk analisis lebih lanjut. Diagram ini memperlihatkan bahwa integrasi antara helm pengaman pintar, protokol MQTT, dan ERP Odoo memungkinkan alur kerja yang terstruktur dari pengambilan data lapangan hingga penyajian laporan yang siap digunakan untuk monitoring keselamatan kerja.

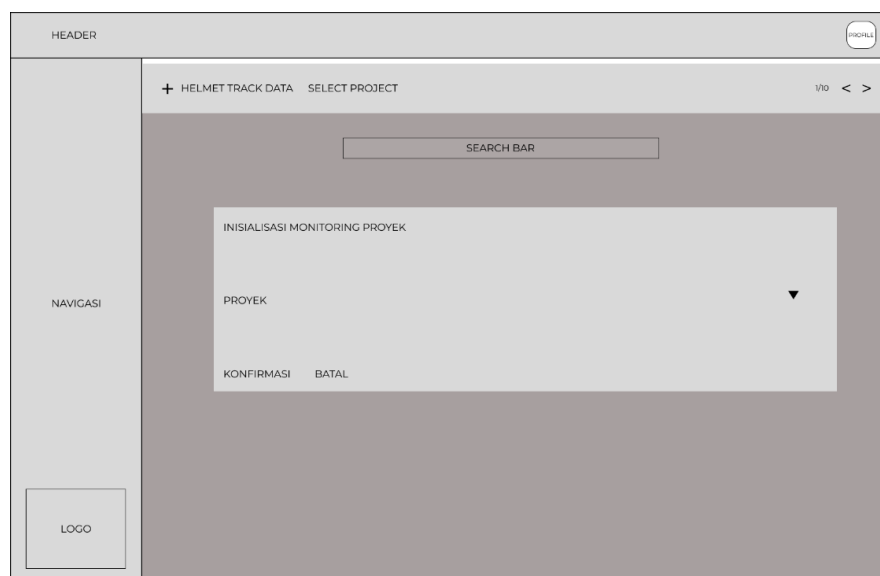
Ketiga, UI/UX Design yang memfokuskan pada penyusunan antarmuka yang sederhana, responsif, dan informatif, dimulai dari Tampilan Utama Odoo, dan berfokus pada add-ons Smart Helmet yang menjadi antarmuka data yang diterima dari helm pengaman pintar berbasis IoT, kemudian tampilan lainnya seperti tampilan table pemantauan pekerja pada halaman *Helmet Track Data*, integrasi peta interaktif untuk memantau posisi pekerja, tampilan laporan pekerja pada halaman

Usage Report, dan tampilan Usage Statistics untuk melihat laporan statistik dari laporan pekerja.



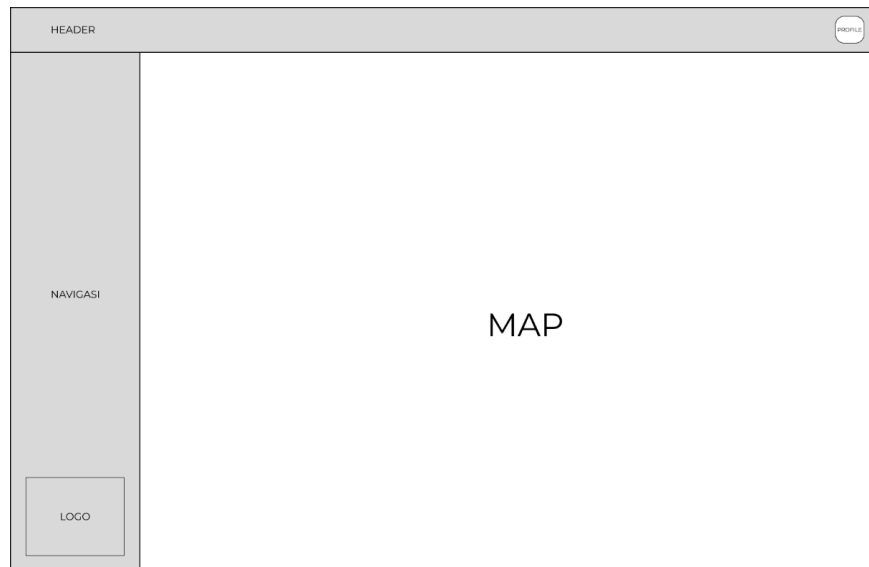
Gambar 3. 15 UI/UX Helmet Track Data

Pada Gambar 3.15 menampilkan desain UI/UX halaman Helmet Track Data pada modul Smart Helmet, halaman ini menampilkan tabel data tracking pekerja sehingga proyek manager dapat melakukan monitoring pada seluruh pekerja pada halaman ini.



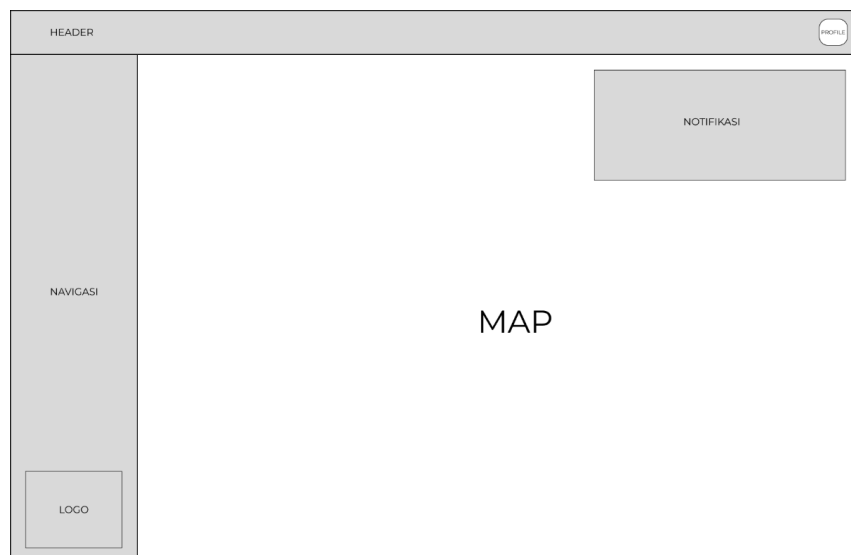
Gambar 3. 16 UI/UX Select Project

Pada Gambar 3.16 menampilkan desain UI/UX Inisialisasi Monitoring Proyek, pada Halaman Helmet Track Data, dimana pemilihan project ini digunakan untuk menentukan Location Work, yang sudah terintegrasi dengan modul Project.



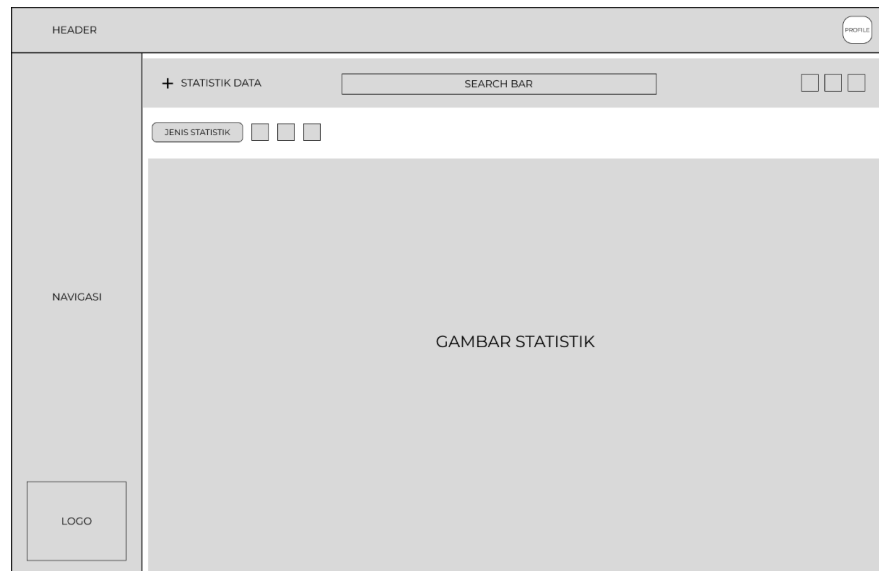
Gambar 3. 17 UI/UX Map Monitoring

Pada Gambar 3.17 menampilkan desain UI/UX halaman Map Monitoring berisi peta interaktif yang berguna untuk melakukan monitoring secara *real time* pekerja, dengan dapat melihat pergerakan lokasi nya ditandai dengan marker lokasi yang dapat dilihat, pada halaman ini juga terdapat notifikasi jika pekerja melakukan pelanggaran.



Gambar 3. 18 UI/UX Notifikasi Pada Map Monitoring

Pada Gambar 3.18 menampilkan desain UI/UX notifikasi pada halaman Map Monitoring, jika terjadi pelanggaran yang dilakukan pekerja. Maka, proyek manager dapat mengetahuinya lewat notifikasi, selagi melakukan monitoring pada tampilan peta interaktif.



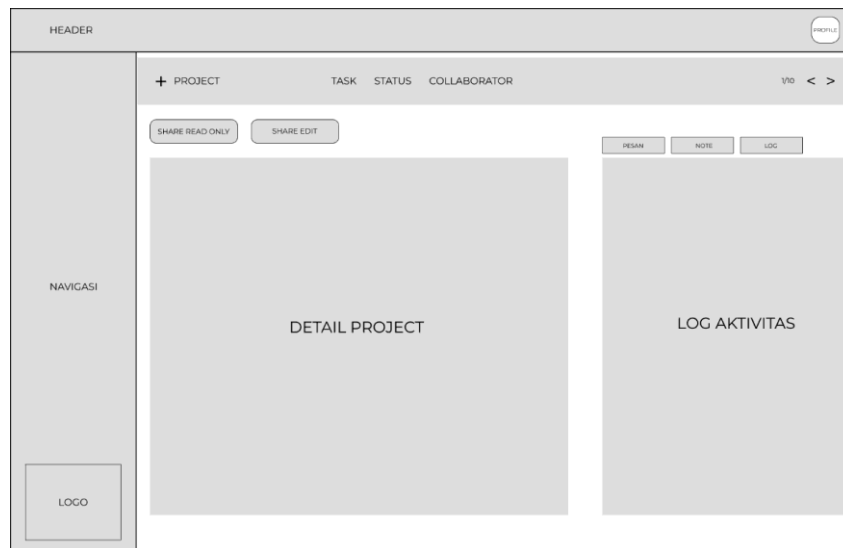
Gambar 3. 19 UI/UX Helmet Statistics

Pada Gambar 3.19 menampilkan desain UI/UX pada halaman Helmet Statistics yang menampilkan grafik statistik terkait penggunaan helm pengaman pintar pada seluruh pekerja berdasarkan status helm dipakai atau tidak, dengan lokasi berada di dalam area pekerjaan.



Gambar 3. 20 UI/UX Halaman Riwayat Notifikasi

Pada Gambar 3.20 menampilkan desain UI/UX pada halaman Riwayat Notifikasi dimana menampilkan riwayat notifikasi pelanggaran yang sudah muncul di halaman Map Monitoring kemudian riwayat nya masuk pada halaman ini.



Gambar 3. 21 UI/UX Settings Project

Pada Gambar 3.21 menampilkan desain UI/UX halaman Settings pada modul Project untuk menentukan lokasi kerja proyek dan radius jarak untuk geofencing, sesudah membuat project baru dalam modul Project. Modul ini merupakan default Odoo tetapi dilakukan kustomisasi untuk menentukan lokasi proyek dan radius jarak untuk kebutuhan geofencing pada Modul Smart Helmet.



Gambar 3. 22 UI/UX Pilih Lokasi Project

Pada Gambar 3.22 menampilkan desain UI/UX pada halaman Map Project di modul Project untuk memilih lokasi proyek yang akan dilaksanakan pada modul Project. Hal ini berkaitan dengan modul Smart Helmet.

3. Development

Setelah melewati tahap desain, tahap selanjutnya adalah tahap pengembangan. Pada tahap ini, seluruh rancangan yang telah dibuat sebelumnya mulai diimplementasikan dalam bentuk kode program dan komponen fungsional lainnya. Tahap ini mencakup proses pengembangan antarmuka pengguna (UI), integrasi dengan *script listener* berbasis Python yang berfungsi mengambil data dari broker MQTT, kemudian mengirimkannya ke Odoo menggunakan XML-RPC API. Di Odoo, dibuat *add-ons* kustom bernama Smart Helmet yang berisi model *fits.monitoring* untuk menyimpan data status helm, koordinat lokasi, waktu pencatatan, dan identitas pekerja. Modul ini juga dilengkapi logika *geofencing* untuk memeriksa posisi pekerja terhadap lokasi proyek.

Selama tahap pengembangan ini, dilakukan juga serangkaian uji coba untuk memastikan setiap fitur berfungsi dengan baik. Uji coba dilakukan terhadap koneksi integrasi dengan MQTT, data yang diterima sesuai dengan data yang dikirim, fungsionalitas geofencing, fungsionalitas export data, hingga tampilan yang sesuai untuk mudah dipahami oleh pengguna. Semua proses tersebut dilakukan secara bertahap sesuai dengan prinsip pengembangan Agile, sehingga memudahkan identifikasi serta perbaikan bug atau error yang muncul selama proses pengembangan berlangsung.

4. Testing

Tahapan selanjutnya setelah pengembangan adalah testing, yang bertujuan untuk memastikan bahwa semua fitur dan komponen yang telah dikembangkan berjalan sesuai dengan fungsi yang telah dirancang. Pada tahap ini dilakukan serangkaian pengujian sistem secara menyeluruh, baik secara fungsional maupun integratif, guna mendeteksi adanya bug, kesalahan logika, atau kendala kompatibilitas pada ERP Odoo. Pengujian dilakukan menggunakan skenario uji yang telah disusun sebelumnya, dengan mengacu pada tabel uji coba yang mencakup setiap fitur utama, integrasi antar modul yang berkaitan dengan add-ons Smart Helmet, melakukan fungsionalitas geofencing, serta menyajikan dan memvisualisasikan semua data sesuai dan mudah dipahami oleh pengguna.

Pengujian dilakukan dalam dua bentuk yaitu pengujian unit dan pengujian integrasi. Pengujian unit difokuskan pada masing-masing fitur secara terpisah,

sementara itu pengujian integrasi dilakukan dengan memastikan bahwa seluruh modul dapat bekerja secara terkoordinasi. Apabila seluruh fitur utama berhasil dijalankan dengan status Berhasil, yang berarti ERP Odoo telah memenuhi ekspektasi yang ditentukan pada masing-masing skenario uji. Hasil pengujian ini juga akan ditampilkan dalam bentuk tabel untuk menunjukkan input, proses, dan output yang diharapkan serta status keberhasilannya. Tahapan testing ini sangat penting untuk memastikan bahwa sistem siap digunakan dengan performa yang baik dan stabil.

5. Deployment

Setelah tahapan testing, tahapan selanjutnya adalah deployment. Tahapan deployment ini merupakan tahapan lanjutan setelah sistem berhasil dikembangkan dan diuji. Pada tahap ini, dilakukan untuk memindahkan sistem dari lingkungan pengembangan ke lingkungan operasional. Instalasi Odoo 17 dilakukan pada server yang telah dikonfigurasi agar dapat berjalan sebagai layanan otomatis (*service*) sehingga tetap aktif meskipun sistem di-*restart*. Broker MQTT di HiveMQ Cloud disiapkan untuk menerima koneksi dari helm pengaman pintar. *Add-ons* Smart Helmet diunggah dan diaktifkan pada server Odoo, kemudian dilakukan uji koneksi untuk memastikan aliran data dari perangkat IoT ke ERP Odoo berjalan dengan baik. Setelah seluruh konfigurasi selesai, sistem mulai digunakan oleh pengguna akhir untuk memantau kondisi lapangan secara *real-time*.

Meskipun ERP Odoo telah berhasil dijalankan di lingkungan nyata, tahap deployment ini bukan merupakan tahap akhir dalam pengembangan sistem. Hasil dari tahap ini akan menjadi dasar untuk memasuki tahap review, di mana sistem akan dievaluasi kembali secara menyeluruh untuk melihat kekurangan, potensi pengembangan lanjutan, serta perbaikan berdasarkan hasil dari pengujian lapangan. Dengan demikian, tahap deployment menjadi momen penting dalam mengukur kesiapan sistem sebelum dilakukan penyempurnaan lebih lanjut.

6. Review

Tahapan terakhir dalam proses pengembangan ERP Odoo dengan metode Agile pada penelitian ini adalah tahapan review, yaitu tahapan evaluasi menyeluruh terhadap seluruh proses dan hasil yang telah dicapai selama siklus pengembangan.

Tujuan dari tahap ini adalah untuk menilai performa sistem secara keseluruhan, mengevaluasi apakah sistem telah memenuhi semua kebutuhan yang telah ditetapkan di awal, serta mengidentifikasi kekurangan yang masih ada, baik dari sisi teknis maupun fungsional.

Pada tahap review, hasil dari implementasi dan pengujian di lingkungan nyata dianalisis kembali. Evaluasi dilakukan terhadap beberapa aspek penting seperti menilai aspek keakuratan data, kecepatan notifikasi, kemudahan penggunaan antarmuka, dan kestabilan koneksi sistem. Hasil dari proses deployment menjadi acuan untuk mengetahui apakah sistem berjalan sesuai ekspektasi masih memerlukan penyempurnaan lebih lanjut. Selain evaluasi teknis, tahap ini juga digunakan untuk meninjau ulang dokumentasi pengembangan, efektivitas metode Agile yang digunakan, serta efisiensi alur kerja selama pengembangan aplikasi berlangsung. Jika ditemukan kekurangan atau potensi pengembangan tambahan, maka sistem dapat dimasukkan kembali ke dalam siklus pengembangan selanjutnya, sesuai prinsip pengembangan Agile yang fleksibel dan adaptif terhadap perubahan. Dengan dilaksanakannya tahap review ini, memberikan ruang untuk perbaikan berkelanjutan demi meningkatkan kualitas, skalabilitas, dan keberlanjutan sistem di masa mendatang.

3.4 Uji Coba Terbatas

Setelah melakukan pengembangan produk awal, berupa pengembangan helm pengaman pintar dan ERP Odoo. Tahap selanjutnya adalah melakukan uji coba terbatas pada hasil pengembangan yang telah dilakukan.

3.4.1 Metode Pengujian Helm Pengaman Pintar

Peneliti melakukan pengujian pada perangkat keras dengan menggunakan metode *functional testing*. *Functional testing* merupakan metode pengujian yang bertujuan untuk mengevaluasi fungsionalitas setiap komponen dan perangkat keras yang digunakan (Pratama, L. I.M., 2022). Pengujian ini memastikan bahwa semua elemen dalam sistem beroperasi sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Pengujian dilakukan dengan berbagai skenario yang telah disiapkan berdasarkan kemungkinan kejadian nyata di lapangan. Berikut Tabel 3.1 menjelaskan skema pengujian fungsionalitas alat, dan Tabel 3.2 yang menjelaskan skema pengujian alat berdasarkan ketahanan daya pada alat.

Tabel 3. 1 Skema Pengujian Alat

No	Skenario Alat	Input	Output yang Diharapkan
1.	Pekerja menyalakan dan memakai alat	Saklar ON ditekan	Alat menyala dan seluruh komponen menyala.
2.	Akurasi lokasi pada alat	Longitude dan Latitude	Longitude dan Latitude yang didapatkan sesuai dengan posisi aktual
3.	Perubahan posisi saat alat berpindah tempat	Waktu update koordinat (refresh rate)	Update posisi perpindahan alat ± 10 detik
4.	Cahaya dalam keadaan gelap, menandakan status alat sedang dipakai	Nilai LDR	Status menunjukkan bahwa alat sedang dipakai
5.	Cahaya dalam keadaan terang, menandakan status alat sedang tidak dipakai	Nilai LDR	Status menunjukkan bahwa alat sedang tidak dipakai
6.	Daya pada alat dalam keadaan lemah	Tidak ada input	Buzzer menyala dan LED berkedip pada saat daya baterai pada alat $\pm 20\%$

Tabel 3. 2 Skema Pengujian Alat Berdasarkan Ketahanan Daya

No	Jenis Baterai	Waktu Mulai	Waktu Selesai	Kesimpulan
1	Lithium 18650			
2	Lithium NCR18650B			
3	Lithium IMR18650			

3.4.2 Metode Pengujian Odoo

Pada implementasi *software* dilakukan pengujian sistem dan keseluruhan fungsi dilakukan dengan menggunakan metode *black-box testing*, di mana sistem diuji berdasarkan keluaran yang dihasilkan tanpa melihat kode sumber yang mendasarinya (Yulianti, Y., dkk. 2023). Beberapa aspek yang diuji dalam tahap ini meliputi validasi input, keakuratan pemrosesan data, respons sistem terhadap berbagai skenario penggunaan, serta kompatibilitas dengan perangkat keras yang digunakan. Pengujian juga mencakup evaluasi terhadap antarmuka pengguna (*user interface*) untuk memastikan kemudahan penggunaan dan efektivitas interaksi dengan sistem.

Hasil dari pengujian perangkat lunak akan dianalisis untuk mengidentifikasi potensi kesalahan atau kekurangan yang perlu diperbaiki. Jika ditemukan kendala atau ketidaksesuaian, maka dilakukan perbaikan dan pengujian ulang hingga sistem berfungsi dengan optimal sesuai dengan kebutuhan. Berikut merupakan tabel skema pengujian ERP Odoo dengan metode *black-box testing* ditunjukkan pada Tabel 3.3.

Tabel 3. 3 Skema Pengujian ERP Odoo dengan Black-box testing

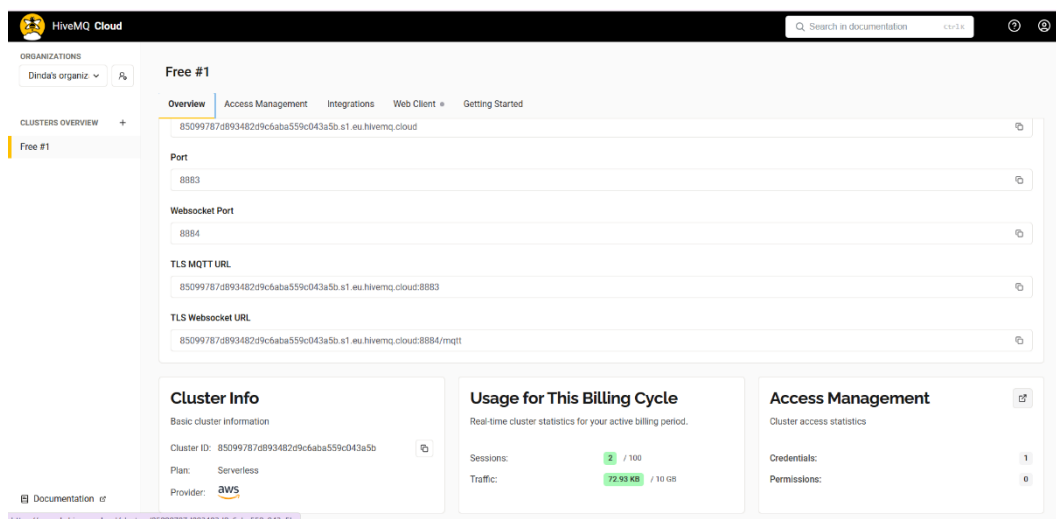
No	Fitur yang diuji	Skenario Pengujian	Input	Output
1.	Inisiasi terhubung dengan Alat	ERP Odoo terhubung dengan MQTT Broker	Kredensial MQTT	MQTT Terhubung
2.	Menampilkan halaman utama Odoo	Mengakses URL ERP Odoo melalui browser	URL sistem ERP Odoo dan kredensial login yang valid	Halaman utama ERP Odoo tampil dengan semua modul yang terpasang
3.	Menampilkan Halaman “Home”	Memilih modul Smart Helmet	Klik modul Smart Helmet pada	Halaman Helmet Track Data tampil

No	Fitur yang diuji	Skenario Pengujian	Input	Output
	submenu “Helmet Track Data”		Halaman Utama	dengan filter dan tabel data helm
4.	Memilih proyek pada “Select Project”	Menguji pemilihan proyek untuk mengisi data “Location Work” dan Geofencing	Pilihan proyek dari “Select Project”	Data location work yang tampil sesuai dengan proyek yang dipilih
5.	Menampilkan tabel data pada submenu “Helmet Track Data”	Mengakses tabel yang menampilkan seluruh informasi data helm pekerja	Data helm tersimpan di database	Tabel menampilkan data pekerja, status helm, lokasi, dan waktu secara <i>real-time</i>
6.	Menampilkan Halaman “Map Monitoring”	Mengakses halaman peta pemantauan lokasi pekerja secara <i>real-time</i>	Klik menu “Map Monitoring”	Peta interaktif tampil, menunjukkan radius geofencing dan posisi pekerja sesuai data GPS helm
7.	Menampilkan Halaman “Home” submenu “Usage Report”	Mengakses halaman laporan penggunaan helm	Klik menu “Home” → “Usage Report”	Halaman Usage Report tampil dengan tabel rekap penggunaan helm
8.	Mengunduh file Laporan pada	Menguji fitur ekspor laporan	Klik tombol “Export”	File laporan dalam format .xlsx berhasil

No	Fitur yang diuji	Skenario Pengujian	Input	Output
	Halaman “Usage Report”	penggunaan helm		diunduh dengan data yang sesuai
9.	Menampilkan Halaman “Helmet Statistics”	Mengakses statistik penggunaan helm	Klik menu “Helmet Statistics”	Grafik dan chart statistik penggunaan helm tampil sesuai data yang tersimpan
10.	Menampilkan halaman “Settings” untuk memilih proyek pada modul Projects	Menguji akses ke pengaturan proyek	Klik menu “Projects” → “Settings”	Halaman Settings Project tampil dengan daftar proyek
11.	Memilih lokasi proyek di halaman “Map Project” pada modul Projects	Menguji fitur pemilihan lokasi proyek	Klik peta pada Map Project dan simpan lokasi	Lokasi proyek berhasil tersimpan dan ditampilkan di sistem

3.5 Penyempurnaan Produk Awal

Setelah melakukan uji coba terbatas pada helm pengaman pintar dan ERP Odoo secara terpisah. Tahap selanjutnya adalah melakukan penyempurnaan produk awal. Pada tahap ini dilakukan penerapan sistem yang telah dirancang, baik dari sisi perangkat keras (*hardware*) maupun perangkat lunak (*software*). Penyempurnaan produk awal ini bertujuan untuk mengintegrasikan sistem yang telah dirancang. Sistem dikembangkan dengan mengacu pada rancangan arsitektur yang telah ditetapkan, dimulai dari mengintegrasikan prototype helm pengaman pintar dengan ERP Odoo menggunakan MQTT sebagai protokol yang menjembatani mengirim data dari ESP32 pada prototype untuk dapat diterima oleh ERP Odoo. Ditunjukkan pada Gambar 3.23 konfigurasi koneksi MQTT yang digunakan pada penelitian ini.



Gambar 3. 23 Konfigurasi Koneksi MQTT

Pada Gambar 3.23 memperlihatkan halaman konfigurasi MQTT pada layanan *cloud* berbasis web HiveMQ, yang digunakan sebagai *broker* untuk mengelola komunikasi data antara perangkat IoT dan sistem ERP Odoo. Pada bagian atas halaman, ditampilkan informasi detail *cluster*, termasuk Cluster ID yang menjadi identitas unik, alamat URL untuk koneksi MQTT (*TLS MQTT URL* dan *TLS Websocket URL*), serta port yang digunakan (Port 8883 untuk TLS MQTT dan Port 8884 untuk TLS Websocket). Bagian Cluster Info memuat informasi penyedia layanan (*provider*) yang dalam kasus ini menggunakan AWS, serta jenis paket yang digunakan (*Serverless*). Konfigurasi ini menjadi komponen penting dalam integrasi helm pengaman pintar berbasis IoT ini, karena memastikan data dari perangkat dapat ditransmisikan secara aman, andal, dan terstruktur ke dalam sistem ERP untuk keperluan pemantauan dan analisis. Selain itu, pada pengembangan sistem ini terdapat fitur geofence untuk menentukan status pekerja berada didalam area kerja atau berada diluar area kerja. Dengan satu lokasi yang didapatkan dari ERP Odoo sebagai titik acuan (Lokasi Kerja) dan satu lokasi yang didapatkan dari alat melalui modul GPS yang menandakan posisi lokasi pekerja berada. Berikut konfigurasi geofencing pada ERP Odoo yang berada dalam konfigurasi yang sama dengan MQTT, ditunjukkan pada Gambar 3.24.


```

110 # === Ambil lokasi kerja & radius dari project ===
111 geofence_radius = 0.0
112 latitude_work = 0.0
113 longitude_work = 0.0
114 if project_id:
115     proj = models.execute_kw(ODOO_DB, uid, ODOO_PASSWORD,
116                             ['project.project', 'read',
117                             [project_id], {'fields': ['geofence_radius', 'latitude', 'longitude']}]
118     )
119     if proj:
120         geofence_radius = proj[0].get('geofence_radius') or 0.0
121         latitude_work = proj[0].get('latitude') or 0.0
122         longitude_work = proj[0].get('longitude') or 0.0
123
124 # --- Hitung jarak geofence ---
125 distance = 0.0
126 out_of_bounds = False
127 if lat and lon and latitude_work and longitude_work:
128     R = 6371000
129     lat1, lon1 = radians(float(latitude_work)), radians(float(longitude_work))
130     lat2, lon2 = radians(float(lat)), radians(float(lon))
131     dlat, dlon = lat2-lat1, lon2-lon1
132     a = sin(dlat/2)**2 + cos(lat1)*cos(lat2)*sin(dlon/2)**2
133     c = 2*atan2(sqrt(a), sqrt(1-a))
134     distance = R*c
135     if geofence_radius > 0:
136         out_of_bounds = distance > geofence_radius
137
138 # === Simpan monitoring ===
139 update_data = {
140     'employee_id': employee_id,
141     'project_id': project_id,
142     'time': now.strftime('%Y-%m-%d %H:%M:%S'),
143     'location_now': alamat,
144     'latitude_now': lat or 0.0,
145     'longitude_now': lon or 0.0,
146     'status': status,
147     'battery_percentage': batt_percent,
148     'distance_to_work': distance,
149     'out_of_bounds': out_of_bounds
150 }
151 print(f"DEBUG: project_id = {project_id}, radius = {geofence_radius}, distance = {distance}")
152 models.execute_kw(ODOO_DB, uid, ODOO_PASSWORD,
153                 ['fits.monitoring', 'create', [update_data]]
154                 log_event(f"Data masuk: {update_data}")

```

Gambar 3. 24 Konfigurasi Geofencing sistem ERP Odoo

Pada Gambar 3.24 menampilkan kode program yang digunakan untuk melakukan konfigurasi geofencing pada sistem *Smart Helmet*. Bagian awal kode berfungsi untuk mengambil data lokasi kerja yang tersimpan pada *record monitoring*, meliputi koordinat lintang (*latitude_work*), bujur (*longitude_work*), dan radius area kerja (*geofence_radius*). Selanjutnya, dilakukan validasi apakah data koordinat valid, kemudian dilakukan perhitungan jarak antara lokasi pekerja (*lat*, *lon*) dengan titik pusat area kerja (*latitude_work*, *longitude_work*) menggunakan rumus Haversine. Rumus ini mempertimbangkan kelengkungan bumi dengan radius bumi yang diatur sebesar 6.371.000 meter, sehingga hasil perhitungan jarak menjadi lebih akurat. Hasil perhitungan disimpan dalam variabel *distance*, dan status *out_of_bounds* akan bernilai True jika jarak melebihi *geofence_radius*, yang mengindikasikan pekerja berada di luar area kerja yang ditentukan. Sebaliknya, jika berada dalam radius, nilai *out_of_bounds* akan menjadi False. Implementasi logika ini memungkinkan sistem untuk memantau posisi pekerja secara real-time dan memberikan peringatan apabila pekerja keluar dari area kerja yang aman.

3.6 Uji Coba Sistem

Setelah melakukan penyempurnaan produk awal, kemudian tahap selanjutnya dilakukan kembali uji coba dengan keseluruhan sistem pada penelitian

ini. Tujuan dari pengujian ini adalah untuk mengevaluasi keseluruhan sistem dapat menjalankan tugasnya dengan baik dalam skenario yang dirancang, sehingga memungkinkan pendeteksian potensi masalah atau ketidaksesuaian. Pengujian akan dilakukan dengan beberapa skenario sebagai bahan uji fungsionalitas. Skenario pengujian meliputi skema pengujian sistem berdasarkan intensitas cahaya untuk mengetahui intensitas cahaya dapat berpengaruh terhadap akurasi status penggunaan helm pengaman pintar yang ditunjukkan pada Tabel 3.4, serta skema pengujian sistem berdasarkan kondisi yang bisa terjadi di lapangan yang ditunjukkan dalam Tabel 3.5.

Tabel 3. 4 Skema Pengujian Sistem Berdasarkan Intensitas Cahaya

No	Waktu Pengujian	Skenario	Ekspetasi Hasil
1.	Pagi	Pekerja memakai helm dan berada di area kerja	Odoo menyimpan status = dipakai, dan lokasi valid dalam area
		Pekerja tidak memakai helm tapi berada di area kerja	Odoo menyimpan status = tidak dipakai, muncul notifikasi helm tidak dipakai
		Pekerja memakai helm tapi berada diluar area kerja	Odoo menyimpan lokasi tidak valid karena berada diluar area, muncul notifikasi pekerja berada di luar area
		Pekerja tidak memakai helm dan berada diluar area kerja	Odoo menyimpan status tidak dipakai dan lokasi tidak valid, muncul notifikasi helm tidak dipakai dan lokasi diluar area
2.	Siang	Pekerja memakai helm dan berada di area kerja	Odoo menyimpan status = dipakai, dan lokasi valid dalam area
		Pekerja tidak memakai helm tapi berada di area kerja	Odoo menyimpan status = tidak dipakai, muncul notifikasi helm tidak dipakai

No	Waktu Pengujian	Skenario	Ekspetasi Hasil
		Pekerja memakai helm tapi berada diluar area kerja	Odoo menyimpan lokasi tidak valid karena berada diluar area, muncul notifikasi pekerja berada di luar area
		Pekerja tidak memakai helm dan berada diluar area kerja	Odoo menyimpan status tidak dipakai dan lokasi tidak valid, muncul notifikasi helm tidak dipakai dan lokasi diluar area
3.	Malam	Pekerja memakai helm dan berada di area kerja	Odoo menyimpan status = dipakai, dan lokasi valid dalam area
		Pekerja tidak memakai helm tapi berada di area kerja	Odoo menyimpan status = tidak dipakai, muncul notifikasi helm tidak dipakai
		Pekerja memakai helm tapi berada diluar area kerja	Odoo menyimpan lokasi tidak valid karena berada diluar area, muncul notifikasi pekerja berada di luar area
		Pekerja tidak memakai helm dan berada diluar area kerja	Odoo menyimpan status tidak dipakai dan lokasi tidak valid, muncul notifikasi helm tidak dipakai dan lokasi diluar area

Tabel 3. 5 Skema Pengujian Sistem Berdasarkan Kondisi

No	Skenario Uji Lapang	Input	Output yang Diharapkan
1.	Pekerja memakai helm dan berada di area kerja	LDR = gelap, GPS = Geofencing dalam area	Odoo menyimpan status = dipakai, dan lokasi valid dalam area

No	Skenario Uji Lapang	Input	Output yang Diharapkan
2.	Pekerja tidak memakai helm tapi berada di area kerja	LDR= terang, GPS = Geofencing dalam area	Odoo menyimpan status = tidak dipakai, muncul notifikasi helm tidak dipakai
3.	Pekerja memakai helm tapi berada diluar area kerja	LDR = gelap, GPS = Geofencing diluar area	Odoo menyimpan lokasi tidak valid karena berada diluar area, muncul notifikasi pekerja berada di luar area
4.	Pekerja tidak memakai helm dan berada diluar area kerja	LDR= terang, GPS = Geofencing diluar area	Odoo menyimpan status tidak dipakai dan lokasi tidak valid, muncul notifikasi helm tidak dipakai dan lokasi diluar area
6.	Pekerja menyimpan helm dengan kondisi telungkup dan berada didalam area	LDR= terang, GPS = Geofencing dalam area	Odoo menyimpan status = tidak dipakai, muncul notifikasi helm tidak dipakai
7.	Pekerja menyimpan helm dengan kondisi telungkup dan berada diluar area	LDR= terang, GPS = Geofencing diluar area	Odoo menyimpan status tidak dipakai dan lokasi tidak valid, muncul notifikasi helm tidak dipakai dan lokasi diluar area
8.	Pekerja menyimpan helm dengan kondisi digantung dan berada didalam area	LDR= terang, GPS = Geofencing dalam area	Odoo menyimpan status = tidak dipakai, muncul notifikasi helm tidak dipakai
9.	Pekerja menyimpan helm dengan kondisi digantung dan berada diluar area	LDR= terang, GPS = Geofencing diluar area	Odoo menyimpan status tidak dipakai dan lokasi tidak valid, muncul notifikasi helm tidak dipakai dan lokasi diluar area

3.7 Evaluasi Sistem Produk

Pada tahap terakhir penelitian ini, dilakukan evaluasi sistem produk dilakukan setelah seluruh tahapan pengujian selesai untuk mengidentifikasi berbagai kendala yang muncul selama proses pengujian dan menentukan langkah perbaikan yang diperlukan. Proses evaluasi bertujuan untuk menilai kinerja helm pengaman pintar dan integrasinya dengan sistem ERP Odoo, sekaligus mengidentifikasi kendala yang mungkin muncul selama pengujian. Evaluasi dilakukan dengan menelaah aspek teknis, integrasi sistem, serta aspek operasional penggunaan helm.

Pada aspek teknis, evaluasi difokuskan pada kinerja sensor, kestabilan komunikasi data melalui protokol MQTT, serta efisiensi mikrokontroler dalam mengolah dan mengirimkan data. Pada aspek integrasi sistem, evaluasi diarahkan untuk menilai sejauh mana data dari helm dapat tersimpan dan ditampilkan di dalam modul Odoo, termasuk fungsi notifikasi. Sementara itu, aspek operasional dievaluasi dari sisi kenyamanan penggunaan helm oleh pekerja, daya tahan perangkat keras, serta adaptabilitasnya di lingkungan kerja konstruksi.

Hasil dari evaluasi ini akan dijadikan dasar untuk melakukan revisi operasional produk serta sebagai dasar dalam memberikan rekomendasi perbaikan pada versi sistem selanjutnya. Dengan demikian, tahap evaluasi sistem produk ini tidak hanya berfungsi untuk menilai performa, tetapi juga sebagai umpan balik penting dan refleksi untuk pengembangan di masa depan agar sistem semakin optimal dan adaptif sesuai dengan kebutuhan pengguna untuk dapat dikembangkan lebih lanjut pada penelitian selanjutnya.