

**IMPLEMENTASI KONSEP *BLOCKCHAIN* PADA PEMBUATAN
PLATFORM *TRACEABILITY* UNTUK INDUSTRI KELAPA SAWIT**

(Skripsi)

Oleh

Mumtaz Farid Wijaya

1915031043



JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2023

ABSTRAK

IMPLEMENTASI KONSEP *BLOCKCHAIN* PADA PEMBUATAN PLATFORM *TRACEABILITY* UNTUK INDUSTRI KELAPA SAWIT

Oleh

MUMTAZ FARID WIJAYA

Kondisi umum yang terjadi di industri kelapa sawit adalah kurangnya visibilitas dan akuntabilitas dalam transfer pasokan bahan baku dari petani ke pabrik, ke produsen, ke gerai ritel dan akhirnya ke konsumen. Sistem rantai pasok (*supply chain*) konvensional ini menyebabkan tingkat keterlacakan produk CPO dan produk turunannya menjadi rendah. Dalam bisnis modern hal ini dapat menyebabkan rendahnya tingkat kepercayaan (*trust*) konsumen dan *stakeholder* terhadap produk CPO sawit. Rendahnya *trust* dan transparansi produksi menyebabkan industri sawit rentan oleh kampanye hitam (*black campaign*) oleh pihak-pihak yang kontra, yang pada akhirnya dapat berpengaruh pada keberlanjutan industri sawit. Oleh sebab itu, industri sawit memerlukan suatu sistem ketelusuran yang berkelanjutan. Dalam penelitian ini, platform *traceability* informasi telah dibuat dan diuji sehingga hasil CPO telah dapat ditelusuri ketelusurannya menggunakan kode unik yang disediakan ketika admin *stakeholder* menginputkan data pada aplikasi serta masing-masing *stakeholder* telah menjadi satu blok sesuai konsep *blockchain* dalam proses pendistribusian dari TBS hingga CPO. Pengujian dilakukan menggunakan *assessment* proses pengembangan dan *assessment* kepuasan user yang diperoleh dari kuesioner *stakeholder* terkait. *Assessment* proses pengembangan menghasilkan nilai rata-rata skala 3,9. Sedangkan untuk *Assessment* kepuasan *user* memiliki 3 aspek penilaian yaitu aspek pengetahuan, aspek fungsionalitas *traceability* aplikasi, dan aspek fungsionalitas *blockchain* aplikasi. Melalui aplikasi ini, model manajemen rantai pasok kelapa sawit dari tradisional ditransformasikan menjadi model *blockchain* melalui fitur *traceability* yang berfungsi sebagai ERP dan media untuk memastikan ketelusuran produk CPO. Dengan konsep *blockchain*, data dan informasi bisnis tersimpan dan terakses oleh *stakeholder*, memastikan *visibility*, transparansi, dan akuntabilitas. Data yang tersimpan juga bisa dijadikan sebagai dukungan pengambilan keputusan bagi manajemen. Aplikasi ini juga memastikan kepercayaan dan keberlanjutan produk sawit serta meminimalisasi dampak kampanye hitam.

Kata kunci: **Kelapa Sawit, Blockchain, Traceability, Supply-Chain, Sustainable Oil Palm**

ABSTRACT**IMPLEMENTATION OF BLOCKCHAIN CONCEPT IN DEVELOPING
TRACEABILITY PLATFORM FOR PALM OIL INDUSTRY****By****MUMTAZ FARID WIJAYA**

The general condition in the palm oil industry is the lack of visibility and accountability in the supply chain from farmers to factories, to producers, to retail outlets, and finally to consumers. This conventional supply chain system results in low traceability of CPO and derivative products. In modern business, this can result in low consumer and stakeholder trust in palm oil products. The low trust and production transparency make the palm oil industry vulnerable to black campaigns by anti-parties, which ultimately affects the sustainability of the palm oil industry. Therefore, the palm oil industry requires a sustainable traceability system. In this research, a traceability information platform has been created and tested so that CPO results can be traced using a unique code provided when stakeholders input data into the application and each stakeholder becomes a block according to the blockchain concept in the distribution process from Fresh Fruit Bunches to CPO. Testing was carried out using a development process assessment and a user satisfaction assessment obtained from stakeholder questionnaires. The development process assessment resulted in an average score of 3.9 on a scale. Meanwhile, the user satisfaction assessment has 3 evaluation aspects, namely knowledge aspects, traceability application functionality aspects, and blockchain application functionality aspects. Through this application, the traditional palm oil supply chain management model is transformed into a blockchain model through the traceability feature, which serves as an ERP and a media to ensure product traceability. With the blockchain concept, business data and information are stored and accessible by stakeholders, ensuring visibility, transparency, and accountability. Stored data can also be used as a support for decision-making by management. This application also ensures trust and sustainability of palm oil products and minimizes the impact of black campaigns.

Keywords: Palm Oil, Blockchain, Traceability, Supply-Chain, Sustainable Oil Palm

**IMPLEMENTASI KONSEP *BLOCKCHAIN* PADA PEMBUATAN
PLATFORM *TRACEABILITY* UNTUK INDUSTRI KELAPA SAWIT**

Oleh

Mumtaz Farid Wijaya

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

pada

Jurusan Teknik Elektro

Fakultas Teknik Universitas Lampung



JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2023

Judul Skripsi : **IMPLEMENTASI KONSEP *BLOCKCHAIN*
PADA PEMBUATAN PLATFORM
TRACEABILITY UNTUK INDUSTRI
KELAPA SAWIT**

Nama Mahasiswa : **Mumtaq Farid Wijaya**

Nomor Pokok Mahasiswa : 1915031043

Jurusan : Teknik Elektro

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing



Dr. Ing. Ardian Ulvan, S.T., M.Sc.
NIP 19731128 199903 1 005



Aryanto, S.T., M.T.
NIP 19900621 201903 1 001

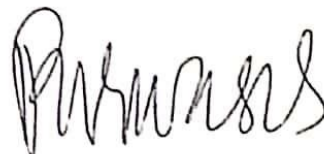
2. Mengetahui

Ketua Jurusan Teknik Elektro



Herlinawati, S.T., M.T.
NIP 19710314 199903 2 001

Ketua Program Studi Teknik Elektro



Dr. Eng. Nining Purwasih, S.T., M.T.
NIP 19740422 200012 2 001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. Ing. Ardian Ulvan, S.T., M.Sc.**



.....

Sekretaris : **Aryanto, S.T., M.T.**



.....

Penguji : **Dr. Ing. Melvi, S.T., M.T.**



.....

2. Dekan Fakultas Teknik



Dr. Eng. H. Helmy Fitriawan, S.T., M.Sc.
NIP 19750928/200112 1 002

Tanggal Lulus Ujian Skripsi : **01 Februari 2023**

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepanjang pengetahuan saya tidak terdapat atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam daftar pustaka. Selain itu, saya menyatakan bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi sesuai dengan hukum yang berlaku

Bandarlampung, 03 Februari 2023



Mumtaz Farid Wijaya

NPM. 1915031043

RIWAYAT HIDUP



Penulis dilahirkan di kota Tangerang, provinsi Banten pada tanggal 13 Maret 2001. Penulis merupakan anak terakhir dari pasangan Bapak Agus Wijaya dan Ibu Hendrawati. Penulis memulai pendidikan di SDN Kebon Besar 2 pada tahun 2007 hingga 2013, SMPN 5 Kota Tangerang pada tahun 2013 hingga 2016, dan SMAN 6 Kota Tangerang pada tahun 2016 hingga 2019.

Penulis menjadi mahasiswa Jurusan Teknik Elektro, Universitas Lampung pada tahun 2019 melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis berkesempatan menjadi asisten praktikum mata kuliah Dasar Sistem dan Jaringan Telekomunikasi dan mata kuliah Sistem Komunikasi mulai tahun 2021 hingga 2022. Selain itu, penulis tergabung dalam lembaga kemahasiswaan yang ada di Jurusan Teknik Elektro (HIMATRO) sebagai sekretaris Departemen Kaderisasi pada tahun 2020 dan menjadi kepala Departemen Kaderisasi pada tahun 2021. Penulis juga aktif dalam mengikuti lomba yang diselenggarakan oleh Belmawa maupun lomba yang diselenggarakan oleh non Belmawa. Penulis mendapatkan pendanaan dari kegiatan Program Kreativitas Mahasiswa pada tahun 2021 yang diselenggarakan oleh Belmawa. Selain itu, penulis juga mendapat pendanaan riset melalui kegiatan Lomba Riset Sawit yang diselenggarakan oleh Badan Pengelola Dana Perkebunan Kelapa Sawit (BPDPKS) pada tahun 2022. Penulis mengikuti kegiatan Merdeka Belajar Kampus Merdeka (MBKM) pada tahun 2021 di mitra Orbit Future Academy yang berfokus pada pengembangan *Artificial Intelligence*. Kemudian, pada tahun 2022 penulis mengikuti kegiatan (MBKM) di mitra Binar Academy yang berfokus pada pengembangan aplikasi berbasis android

PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah, Atas Berkat Rahmat Allah yang Maha Kuasa

KUPERSEMBAHKAN KARYA INI UNTUK

Bapak dan Ibu Tercinta

Agus Wijaya dan Hendrawati

Keluarga Besar, Dosen, Teman, dan Almamater

MOTTO

**“TUGAS KITA HANYA BERIKHTIAR SEMAMPU YANG BISA
DILAKUKAN, SELANJUTNYA BIAR ALLAH YANG MENETAPKAN
APA YANG MENJADI BAGIAN MASLAHAT DARI KEHIDUPAN KITA”**

**“TETAPLAH SEMANGAT SAMPAI BISMILLAH MU MENJADI
ALHAMDULILLAH”**

SANWACANA

Atas berkat rahmat Allah yang Maha Kuasa dan limpahan nikmat Nya yang diberikan kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini. Shalawat senantiasa dicurahkan kepada Nabi Muhammad Shallahu Alaihi Wassalam suri tauladan yang mampu membuka sesuatu yang terkunci, penutup dari semua yang terdahulu, penolong kebenaran dengan jalan yang benar, dan petunjuk kepada jalan Mu yang lurus.

Tugas Akhir dengan judul “Implementasi Konsep *Blockchain* Pada Pembuatan Platform *Traceability* Untuk Industri Kelapa Sawit” ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung. Pada kesempatan ini, penulis mengucapkan terimakasih kepada bapak dan ibu dosen yang telah memberikan bimbingan, saran, nasehat, dan motivasi kepada penulis

1. Bapak Dr. Ing. Ardian Ulvan, S.T., M.Sc selaku dosen pembimbing penelitian yang selalu memberikan bimbingan, nasehat, dan motivasi serta saran yang membangun kepada penulis
2. Ibu Dr. Ing. Melvi, S.T., M.T selaku dosen pembimbing penelitian yang selalu memberikan pengajaran untuk taat pada aturan penulisan yang berlaku untuk meningkatkan kedisiplinan kepada penulis
3. Bapak Aryanto, S.T., M.T selaku dosen pembimbing penelitian yang selalu memberikan saran dan masukan yang membangun dalam pengembangan aplikasi yang dibuat oleh penulis

Semoga bapak dan ibu dosen selalu dalam lindungan Allah SWT.

Selanjutnya, penulis mengucapkan terimakasih kepada

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, DEA, IPM selaku Rektor Universitas Lampung
2. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc selaku Dekan Fakultas Teknik Universitas Lampung
3. Ibu Herlinawati, S.T., M.T selaku Kepala Jurusan Teknik Elektro Universitas Lampung

4. Bapak Meizano Ardhi Muhammad, S.T., M.T selaku Sekretaris Jurusan Teknik Elektro Universitas Lampung
5. Ibu Dr. Eng. Nining Purwasih, S.T., M.T selaku Kepala Prodi Teknik Elektro Universitas Lampung
6. Bapak Sumadi, S.T., M.T selaku dosen Pembimbing Akademik
7. Seluruh dosen di Jurusan Teknik Elektro yang telah memberikan ilmu yang bermanfaat, wawasan dan pengalaman bagi penulis
8. Seluruh staff di Jurusan Teknik Elektro dan Fakultas Teknik yang telah membantu penulis dalam hal administrasi dan hal-hal lainnya
9. Kepada kedua orang tua yang sangat saya cintai dan saya sayangi, aa, tetch, neng yang telah memberikan motivasi dalam hidup
10. Teman seperjuangan keluarga besar telti angkatan 2018, 2019,2020

Bandarlampung, 03 Febaruari 2023

A handwritten signature in black ink, consisting of a large, stylized letter 'F' followed by several loops and a horizontal line at the end.

Mumtaz Farid Wijaya

DAFTAR ISI

| | |
|--|-------------|
| ABSTRAK | ii |
| HALAMAN JUDUL | iv |
| LEMBAR PERSETUJUAN | v |
| LEMBAR PENGESAHAN | vi |
| SURAT PERNYATAAN | vii |
| RIWAYAT HIDUP | viii |
| PERSEMBAHAN..... | ix |
| MOTTO | x |
| SANWACANA | xi |
| DAFTAR ISI..... | xiii |
| DAFTAR GAMBAR..... | xv |
| DAFTAR TABEL..... | xvi |
| BAB I..... | 1 |
| PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Tujuan Penelitian..... | 3 |
| 1.3 Rumusan Masalah | 3 |
| 1.4 Batasan Masalah | 3 |
| 1.5 Manfaat Penelitian..... | 4 |
| 1.6 Sistematika Penulisan..... | 4 |
| BAB II | 6 |
| TINJAUAN PUSTAKA | 6 |
| 2.1 Kajian Pustaka Penelitian Sebelumnya | 6 |
| 2.2 <i>Blockchain</i> Pada Manajemen Rantai Pasok Perusahaan Kelapa Sawit | 8 |
| 2.3 <i>Traceability</i> Pada Manajemen Rantai Pasok Perusahaan Kelapa Sawit | 9 |
| 2.4 Laravel..... | 9 |
| 2.4.1 Struktur Folder Laravel..... | 10 |
| 2.5 PHP..... | 15 |
| 2.6 Visual Studio Code..... | 17 |
| 2.7 XAMPP | 17 |
| 2.8 CPanel Hosting..... | 18 |

| | |
|--|----|
| 2.9 <i>Software Development Life Cycle</i> | 20 |
| 2.10 Scrum | 21 |
| 2.11 <i>Bussines Model Canvas</i> | 23 |
| 2.12 <i>Stakeholder</i> | 24 |
| 2.12Indikator Keberhasilan | 31 |
| BAB III | 32 |
| METODOLOGI PENELITIAN | 32 |
| 3.1 Waktu dan Tempat Penelitian..... | 32 |
| 3.2 Perangkat Penelitian..... | 32 |
| 3.2.1 Perangkat Keras..... | 32 |
| 3.2.2 Perangkat Lunak..... | 32 |
| 3.3 Metode Penelitian..... | 32 |
| BAB IV | 42 |
| KESIMPULAN | 42 |
| 4.1 Kesimpulan..... | 42 |
| 4.2 Saran | 42 |
| DAFTAR PUSTAKA | 42 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Struktur Folder Laravel | 10 |
| Gambar 2.2 Perbandingan Laravel dan CodeIgniter..... | 14 |
| Gambar 2.3 Roadmap PHP | 16 |
| Gambar 2.4 XAMPP Control Panel | 18 |
| Gambar 2.5 <i>Dashboard</i> CPanel | 19 |
| Gambar 2.6 <i>Bussines Model Canvas</i> | 23 |
| Gambar 3.1 Diagram Alir Penelitian | 33 |
| Gambar 3.2 Titik Area <i>Loading Point</i> | 34 |
| Gambar 3.2 Sistem <i>Dashboard Stakeholder</i> | 35 |
| Gambar 3.3 Sistem Portal <i>Traceability</i> | 36 |
| Gambar 3.4 <i>Development</i> Aplikasi | 38 |
| Gambar 1 <i>User Story</i> | 54 |
| Gambar 2 <i>Low Fidelity</i> | 59 |
| Gambar 3 <i>High Fidelity</i> | 61 |
| Gambar 4 <i>Traceability</i> Aplikasi..... | 68 |
| Gambar 5 <i>Blockchain</i> Aplikasi | 69 |
| Gambar 6 Koneksi <i>Database</i> | 69 |
| Gambar 7 <i>Library</i> | 70 |
| Gambar 8 <i>Layout Header</i> | 71 |
| Gambar 9 <i>Layout Footer</i> | 72 |
| Gambar 10 <i>Authentication Login</i> | 73 |
| Gambar 11 <i>Method Create</i> | 73 |
| Gambar 12 <i>Method Read</i> | 74 |
| Gambar 13 <i>Method Update</i> | 76 |
| Gambar 14 <i>Method Delete</i> | 77 |
| Gambar 15 <i>Code Traceability</i> | 77 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 <i>Otentifikasi</i> Pengguna | 24 |
| Tabel 2.2 <i>Fields Item Otentifikasi</i> Pengguna | 25 |
| Tabel 2.3 Master Data Petani | 25 |
| Tabel 2.4 <i>Fields Item</i> Master Data Petani | 26 |
| Tabel 2.5 Master Data Kebun | 26 |
| Tabel 2.6 <i>Fields Item</i> Master Data Kebun | 27 |
| Tabel 2.7 Master Data Logistik..... | 27 |
| Tabel 2.8 <i>Fields Item</i> Master Data Logistik..... | 28 |
| Tabel 2.9 Master Data Pabrik..... | 28 |
| Tabel 2.10 <i>Fields Item</i> Master Data Pabrik | 29 |
| Tabel 2.11 <i>Fields Item</i> Master Data Pesanan..... | 29 |
| Tabel 2.12 <i>Roles dan Permissions</i> | 30 |
| Tabel 4.1 Pengujian <i>Blacbox</i> | 45 |
| Tabel 4.2 <i>Assesment</i> Proses Pengembangan | 47 |
| Tabel 1 <i>Scrum Daily Sprint</i> | 55 |
| Tabel 2.10 <i>Fields Item</i> Master Data Pabrik | 29 |
| Tabel 2.11 <i>Fields Item</i> Master Data Pesanan..... | 29 |
| Tabel 2.8 <i>Fields Item</i> Master Data Logistik..... | 28 |
| Tabel 2.9 Master Data Pabrik..... | 28 |
| Tabel 2.10 <i>Fields Item</i> Master Data Pabrik | 29 |

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Indonesia menjadi negara penghasil sawit dengan lebih dari 90% dari total luas lahan kelapa sawit yang tersebar di 190 kabupaten di Pulau Sumatera dan Pulau Kalimantan. Komposisi kepemilikan kebun sawit, 53% swasta dan 41% rakyat, menciptakan momentum lepas landas bagi pertumbuhan ekonomi sektor perkebunan dan perekonomian Indonesia [1].

Berdasarkan data dari Badan Pengelola Dana Perkebunan Kelapa Sawit (BPDPKS), rantai pasok (*supply chain*) produk sawit saat ini beragam di berbagai daerah. Salah satu contoh rantai pasok yang ada saat ini, Tandan Buah Segar (TBS) dipanen dari kebun petani kemudian dikumpulkan koperasi/kelompok petani. TBS yang sudah dikumpulkan, kemudian dijemput oleh agen kecil untuk didistribusikan ke agen besar, selanjutnya dikirim ke pemegang *delivery order* (DO). Pemegang DO kemudian mengirimkan TBS ke pabrik kelapa sawit. Rantai pasok yang sudah ada ini cukup panjang sehingga juga merugikan petani, karena setiap titik/pihak turut mengambil profit sepanjang rantai pasok [2]. Inefisiensi dalam manajemen rantai pasok dapat mempengaruhi keberlanjutan industri minyak sawit.

Kondisi umum yang terjadi di industri minyak sawit adalah kurangnya visibilitas dan akuntabilitas dalam transfer pasokan bahan baku dari petani ke pabrik, ke produsen, ke gerai ritel dan akhirnya ke konsumen. Beberapa pendekatan untuk mengatasi kekhawatiran yang berkembang tentang keberlanjutan dalam industri minyak sawit telah diadopsi. Misalnya, *Roundtable on Sustainable Palm Oil* (RSPO) telah mengembangkan serangkaian kriteria lingkungan dan sosial yang harus dipenuhi perusahaan untuk menghasilkan *Certified Sustainable Palm Oil* [3].

Indonesia Sustainable Palm Oil (ISPO) juga telah mengatur regulasi dengan dikeluarkannya Peraturan Menteri Pertanian (Permentan) Nomor 38 Tahun 2020 mengenai pembinaan dan pengawasan terhadap pelaksanaan sertifikasi ISPO sebagaimana diatur dalam pasal 25 Peraturan Presiden Nomor 44 Tahun 2020 tentang Sistem Sertifikasi Perkebunan Kelapa Sawit Berkelanjutan Indonesia. Salah satu hasil penetapan pada Permentan yakni pada bagian keenam penilaian rantai pasok pasal 28, menetapkan bahwa Lembaga Swadaya (LS) ISPO diharuskan melakukan penilaian terhadap rantai pasok dalam rangka menjamin ketelusuran bahan baku yang diolah menjadi minyak sawit, serta menjamin ketertelusuran rantai pasok produk kelapa sawit untuk ruang lingkup kebun, pengolahan kelapa sawit dan pengumpulan hasil *crude palm oil (CPO)* atau *bulking*. Dengan demikian, dikeluarkannya Permentan 38/2020 juga berupaya untuk memperbaharui Permentan 11/2015 yang mengatur hal – hal serupa. Mengingat kekhawatiran ini, maka sangat penting untuk mendata seluruh rantai nilai dan rantai pasokan produk sawit, dan memahami elemen-elemen yang terlibat dalam mobilisasi pasokan. Dengan adanya regulasi tersebut, perlu dilakukan pembaharuan dalam manajemen rantai pasok industri kelapa sawit dari yang bersifat konvensional kearah penggunaan konsep *blockchain*. Hal ini dilakukan agar ketelusuran bahan baku yang diolah menjadi kelapa sawit dapat ditelusuri oleh para pemangku kepentingan. Konsep *blockchain* yang dirancang dalam manajemen rantai pasok industri kelapa sawit adalah dengan menjadikan masing–masing *stakeholder* yang terkait dalam proses pengolahan TBS menjadi CPO menjadi satu blok dalam sistem rantai pasok.

Pada penelitian ini dilakukan rancangan manajemen rantai pasok industri kelapa sawit dengan konsep *blockchain*. Implementasi konsep *blockchain* dilakukan melalui pembuatan aplikasi yang berfungsi seperti miniatur *Enterprise Resource Planning (ERP)*, sekaligus sebagai portal sistem *traceability* untuk produk CPO yang diproduksi oleh perusahaan. Selain itu, aplikasi ini dapat membantu dalam meningkatkan transparansi dan akuntabilitas dalam sebuah perusahaan kelapa sawit. Dengan menggunakan aplikasi ini, segala macam transaksi atau data yang terkait dengan produk kelapa sawit bisa terekam secara digital, dapat diverifikasi, namun tidak dapat diubah, sehingga risiko penipuan, manipulasi data atau kecurangan lainnya dapat diminimalisasi.

1.2 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang manajemen rantai pasok untuk industri perusahaan kelapa sawit dengan konsep *blockchain* melalui pembuatan aplikasi yang berfungsi sebagai *platform traceability* dengan menggunakan *framework* Laravel PHP. Aplikasi ini dapat digunakan sebagai media perekaman dan pencatatan data perusahaan kelapa sawit, serta dapat digunakan untuk menelusuri asal-usul produk CPO.

1.3 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Apakah konsep *blockchain* dapat diimplementasikan pada manajemen rantai pasok di sebuah perusahaan kelapa sawit?
2. Apakah dengan penggunaan konsep *blockchain* pada manajemen rantai pasok ketertelusuran produk CPO pada perusahaan kelapa sawit dapat dilakukan?

1.4 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah sebagai berikut :

1. Konsep *blockchain* hanya diimplementasikan pada manajemen rantai pasok utama pada sebuah perusahaan kelapa sawit yang terdiri dari 4 *stakeholder* yaitu *stakeholder* petani, *stakeholder* kebun, *stakeholder* logistik dan *stakeholder* pabrik
2. *Platform traceability* yang dibangun menggunakan pemrograman PHP melalui *framework* Laravel
3. Sistem *database* yang dibangun menggunakan MySQL
4. Proses *traceability* hanya sampai pada produk CPO
5. *Stakeholder* kebun dan *stakeholder* pabrik hanya diwakili oleh satu akun

1.5 Manfaat Penelitian

Pada penelitian ini dapat diambil beberapa manfaat yang mencakup hal-hal sebagai berikut:

1. Terciptanya transparansi, visibilitas, dan akuntabilitas dalam manajemen rantai pasok pada perusahaan kelapa sawit.
2. Produk CPO dari sebuah perusahaan kelapa sawit yang menerapkan aplikasi ini dapat ditelusuri sesuai dengan regulasi Peraturan Presiden Nomor 44 Tahun 2020 dan Permentan Nomor 38 Tahun 2020
3. Penelitian ini dapat dijadikan referensi bagi peneliti dan pengembang *software development* serta pelaku di sektor perkebunan.
4. Penelitian ini dapat menambah pengalaman serta pembelajaran penting bagi penulis mengenai sebuah pembuatan sistem pengelolaan rantai pasok industri sawit.

1.6 Sistematika Penulisan

Dalam rangka mempermudah pemahaman pembaca terhadap topik penelitian ini maka sistematika penulisan laporan skripsi ini dibagi kedalam 5 bab, yaitu:

BAB 1. PENDAHULUAN

Bab pendahuluan berisi uraian yang menjelaskan tentang latar belakang, tujuan penelitian, manfaat penelitian, rumusan masalah, batasan masalah, dan sistematika penulisan.

BAB 2. TINJAUAN PUSTAKA

Tinjauan pustaka berisi uraian pemaparan dari beberapa teori pendukung dan referensi materi yang diperoleh dari berbagai sumber buku, jurnal dan penelitian ilmiah yang digunakan untuk penulisan laporan tugas akhir ini.

BAB 3. METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan tentang waktu dan tempat penelitian, perangkat penelitian yang digunakan, metode penelitian, dan diagram alir penelitian.

BAB 4. HASIL DAN PEMBAHASAN

Hasil penelitian yang dilakukan diuraikan pada bab hasil dan pembahasan dengan menganalisis dan menjelaskan hasil pembuatan aplikasi yang dirancang untuk manajemen rantai pasok perusahaan kelapa sawit.

BAB 5. PENUTUP

Bab penutup berisikan ungkapan penulis mengenai hal – hal penting dalam penelitian sehingga dapat menambah perspektif baru mengenai subjek tema yang dibahas.

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Sehubungan dengan penelitian yang dilakukan, kajian terhadap penelitian-penelitian sebelumnya sangat penting dilakukan untuk mendapat wawasan penulis terkait dengan topik yang diteliti. Selain itu, kajian pustaka dilakukan untuk menghindari plagiarisme atau duplikasi penelitian sebelumnya. Kajian pustaka juga dimaksudkan untuk mengukur posisi penelitian ini terhadap penelitian sejenis dan diharapkan dapat menjadi kontributor agar tema penelitian ini terus berkembang. Berikut beberapa ulasan penelitian terlebih dahulu yang telah dilakukan mengenai metode yang digunakan dan hasil yang diperoleh.

Pada kajian Tsolakis [4] diusulkan kerangka kerja pengambilan keputusan hierarkis yang komprehensif dan taksonomi kritis yang berlaku untuk semua pemangku kepentingan yang terlibat dalam desain dan pengelolaan rantai pasok tanaman pangan (*Agri Food Supply Chain-AFSC*). Hierarki alami dari proses pengambilan keputusan ditentukan untuk perancangan dan perencanaan AFSC. Analisis kritis yang dilakukan menunjukkan bahwa manajemen rantai pasok tanaman pangan adalah bidang riset yang berkembang pesat, namun terlalu fokus pada penanganan masalah yang bergantung pada kasus yang terkait dengan level rantai pasokan yang berbeda dan belum serius pada pengembangan pendekatan metodologis terintegrasi untuk optimasi seluruh rantai pasokan.

Kajian yang lebih spesifik tentang kelapa sawit telah dilakukan oleh (L.P. Tan, C.O. Fong [5] yang menyajikan model pemrograman linear dinamis untuk koneksi keputusan strategis. Model ini memungkinkan untuk mengidentifikasi kombinasi produksi tanaman yang optimal untuk perkebunan tahunan dalam kondisi harga yang tidak pasti. Ini dimaksudkan untuk memaksimalkan pendapatan dalam kondisi berisiko dengan menerapkan penalti saat neraca negatif. Penerapan penalti ini membuka kemungkinan dilakukannya evaluasi untuk menggulirkan proyek

budidaya alternatif sesuai dengan tingkat risiko dan keuntungan diestimasi berdasarkan nilai bersih (*net value*) saat itu. Selain itu, kebijakan ini juga membuka peluang dilakukannya asesmen terhadap faktor – faktor produksi dan biaya ekspor yang harus dibayar perusahaan. Model ini diujicobakan di perkebunan kelapa sawit Malaysia.

García-Cáceres [6] menyajikan fungsionalitas dan nilai terkait dari hubungan antara koneksi dan level jabatan di fase hulu rantai pasokan kelapa sawit di Kolombia. Metodologi yang diterapkan adalah karakterisasi rantai pasok (*characterized supply chain*) dengan cara memberikan deskripsi terorganisir dan rinci di setiap tahapan – tahapan proses dimana produk dan turunannya berada sehingga mengidentifikasi nilai tambah di setiap fase pemrosesan. Dalam kesimpulannya, kualitas tanaman dan produktivitas tanaman adalah aspek kunci untuk menghasilkan kesejahteraan bagi setiap elemen yang terlibat dalam rantai khususnya di fase hulu.

Dari uraian diatas terlihat bahwa pengembangan model rantai pasok sebagian masih menggunakan pendekatan konvensional dengan metode pemrograman linear/integer, model matematis, atau menggunakan *tools* tertentu khususnya untuk sistem pengambilan keputusan. Pada dasarnya rantai pasokan adalah urutan atau jaringan pemasok, produsen, pengangkut, gudang, pengecer, dan pelanggan. Manajemen rantai pasok berusaha untuk mengelola aliran dana, informasi dan produk dalam rantai pasokan untuk memastikan ketersediaan produk dan layanan tingkat tinggi kepada pelanggan dengan biaya serendah mungkin. Saat ini banyak sekali pencatatan dihasilkan dari transaksi antara pemasok dan pembeli [7].

Hingga sejauh ini, tema pengembangan model rantai pasok pada perkebunan kelapa sawit dengan kata kunci pendekatan ketertelusuran informasi produk, mekanisme *blockchain*, atau analisa *big data*, belum banyak diangkat dan dipublikasikan. Literatur-literatur tentang topik ini dan *state-of-the-art* nya telah dibahas mengungkapkan sejumlah kajian bermakna yang berhubungan dengan pengoptimalan taktis dalam rantai pasokan agribisnis.

2.2 *Blockchain* Pada Manajemen Rantai Pasok Perusahaan Kelapa Sawit

Blockchain adalah buku besar kas transaksi bersifat digital dan tersebar, terekam dan terefleksi secara langsung (*real-time*) di jaringan komputer atau *node*. Tidak dibutuhkan sebuah otoritas pusat untuk mengesahkan transaksinya, maka dari itu *blockchain* disebut sebagai mekanisme *trustless peer-to-peer*. Pada hakikatnya, teknologi *blockchain* memberikan cara yang lebih aman bagi buku besar kas untuk menyimpan arsip, dan *database* tanpa perlunya intervensi manual terpusat. Sehingga biaya verifikasi pengarsipan transaksi dalam sistem berbasis *blockchain* jauh lebih murah dibandingkan sistem verifikasi terpusat oleh manusia. Hasilnya adalah sebuah model operasi untuk transaksi dalam sistem berdasarkan kepercayaan yang berbasis sistem (*system-based-trust*) bukan kepercayaan yang basis pihak lain (*counterparty-based-trust*) [7].

Pada dasarnya, yang dibutuhkan adalah sistem *database* yang terintegrasi dan tidak mudah dimanipulasi untuk menyimpan identitas digital produk kelapa sawit, yang kemudian dapat digunakan oleh semua pemangku kepentingan dalam rantai pemasok produk kelapa sawit dari penyuling hingga konsumen akhir. Saat ini, sistem untuk memastikan minyak kelapa sawit yang berkelanjutan masih memakan biaya yang tinggi karena harus sering ada audit, monitor dan pelaporan di keseluruhan rantai nilai produk kelapa sawit (dari kebun hingga proses pembuatan).

Dengan pembuatan aplikasi dengan konsep *blockchain* ini audit, monitor dan pelaporan dilakukan pada tiap-tiap *stakeholder* untuk menciptakan identitas digital produk kelapa sawit dan informasi ini kemudian tidak dimanipulasi di sepanjang rantai nilai produk kelapa sawit. Hal ini akan menjadi solusi yang berbasis teknologi terintegrasi dan dapat diimplementasikan di seluruh sektor produk kelapa sawit, mulai dari sumber (petani minyak kelapa sawit) hingga konsumen akhir. Hal tersebut juga akan berdampak langsung pada peningkatan nilai dan kesadaran mengenai minyak kelapa sawit berkelanjutan [8].

2.3 Traceability Pada Manajemen Rantai Pasok Perusahaan Kelapa Sawit

Industri kelapa sawit di dunia terus diserang isu-isu negatif salah satunya adalah isu deforestasi. Pembangunan kebun kelapa sawit dinilai menjadi salah satu penyebab terjadinya deforestasi dan pemanasan global dunia. Selain itu, maraknya gerakan labelisasi “*Palm Oil Free*” (bebas minyak sawit) telah memberikan tekanan terhadap industri kelapa sawit di Indonesia. Isu–isu yang terjadi dapat dilawan dengan membuat kebijakan bahwa seluruh produk akan menggunakan bahan baku yang berkelanjutan dan tidak melakukan deforestasi [9].

Ketelusuran merupakan kunci untuk mencapai transformasi industri kelapa sawit. Seluruh *retailer, brand, refinery* dan pabrik bekerjasama dengan pemasok tandan buah segar (TBS) untuk mengidentifikasi dari mana asal bahan mentah produk, siapa yang memproduksinya dan dari mana bahan baku berasal. Ketelusuran minyak kelapa sawit memiliki tujuan untuk memahami praktik para pemasok dari proses produksi, pengadaan, penanganan, hingga pengolahan baik dari TBS hingga minyak sawit mentah atau *crude palm oil* (CPO).

2.4 Laravel

Laravel adalah kerangka kerja aplikasi web berbasis PHP yang *open source*, menggunakan konsep *Model-View-Controller* (MVC). Laravel berada dibawah lisensi MIT, dengan menggunakan *GitHub* sebagai tempat berbagi kode. Laravel berfokus pada bagian *end-user* seperti tampilan ataupun penulisan yang menghasilkan fungsionalitas aplikasi web bekerja sebagaimana mestinya. Hal ini membuat *developer* maupun perusahaan menggunakan *framework* ini untuk membangun apapun, mulai dari proyek kecil hingga skala perusahaan kelas atas [10].

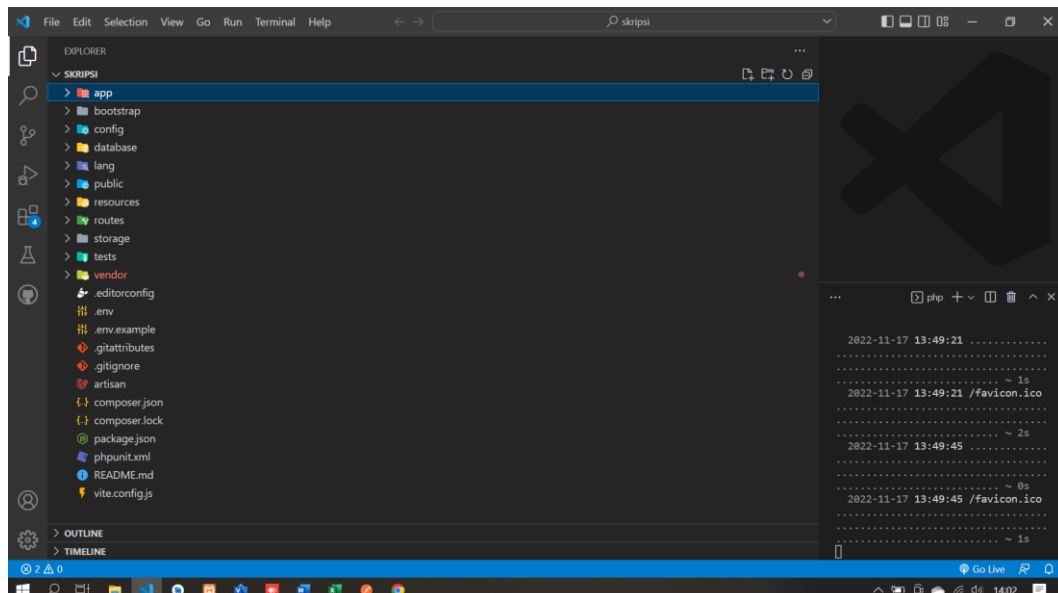
Laravel mengubah pengembangan website menjadi lebih elegan, ekspresif, dan menyenangkan. Selain itu, Laravel juga mempermudah proses pengembangan website dengan bantuan beberapa fitur unggulan, seperti *Template Engine, Routing,* dan *Modularity* [10].

Laravel menawarkan beberapa keuntungan untuk mengembangkan website menggunakan dasar *framework* :

1. Website menjadi lebih *scalable* (mudah dikembangkan)
2. Terdapat *namespace* dan tampilan untuk mengorganisir dan mengatur sumber daya website.
3. Proses pengembangan menjadi lebih cepat sehingga menghemat waktu, karena Laravel dapat dikombinasikan dengan beberapa komponen dari *framework* lain untuk mengembangkan website.

2.4.1 Struktur *Folder* Laravel

Laravel memiliki sebuah struktur berupa susunan aneka folder dengan beragam fungsi atau biasa disebut dengan *direktori*. Laravel terdiri dari beberapa versi, hingga saat ini, versi terbaru Laravel adalah Laravel 9. Meskipun terdiri dari beraneka versi, susunan struktur Laravel tidak jauh berbeda satu sama lain. Perbedaan dari versi sebelumnya terletak pada susunan *sub direktori*. Susunan folder Laravel dapat dilihat pada Gambar 2.1.



Gambar 2.1 Struktur Folder Laravel

Dalam memahami susunan folder direktori Laravel akan mempermudah saat menjalankan aneka tugas dengan Laravel seperti *Create, Read, Update, dan Delete* (CRUD) atau sekedar membedakan dan memilih *folder* yang dibutuhkan untuk kerja membangun *software*.

1. *App/Request/Http*

Folder *direktori app* yang juga dikenal dengan nama *request* atau *http*. Penamaan yang beragam ini muncul karena berkaitan dengan fungsi folder *app* yakni untuk menyimpan semua file *response Http* dan proses *request*. Selain itu, folder ini juga menyimpan semua fungsi *controller*. Folder utama *app* memiliki sejumlah *sub folder/ subdirektori* yang dapat digunakan untuk mempermudah pekerjaan. Beberapa *sub folder direktori* yang dimiliki oleh *app* adalah *middleware, controllers/auth, dan request*.

2. *Middleware*

Middleware adalah *sub direktori* yang berfungsi untuk menampung dan menyimpan semua *class middleware* PHP. *Class middleware* akan dijalankan sebelum *HTTP request* dapat masuk ke *system controller*. Dengan kata lain, *middleware* adalah penghubung dari *request* dan *controller* di Laravel.

3. *Controllers/Auth*

Controller/Auth berfungsi untuk menyimpan seluruh *class* milik *controller* seperti *KebunController.php, LogistikController.php* dan sebagainya. *Folder* ini juga merupakan *folder* Laravel untuk menyimpan semua *sistem autentikasi* Laravel yang dibuat.

4. *Request*

Sub direktori request khusus menyimpan satu tipe *class* yakni *Request.php*. Fungsi *request* digunakan untuk memperoleh data *form request* dari *browser web*. Namun terkadang *folder* ini juga bisa digunakan untuk menyimpan *validator* buatan *user* dengan memasukkan *source code* di *terminal*: `php artisan make:request`.

5. *Test*

Folder test menyimpan beragam komponen *library* untuk kebutuhan *testing framework* yang akan dijalankan oleh *user (programmer)*. Seluruh *package testing* pada Laravel tersimpan dalam bentuk *bundling PHPUnit* sehingga *user* dapat menjalankan *testing* lebih mudah. Jenis *test* yang bisa dijalankan dengan Laravel contohnya adalah *Test Driven Development (TDD)*.

6. *Database/Seeds*

Sub folder database/seeds membuat aneka *seeder database* untuk proses input data awal atau inisialisasi data pada tabel *database* menggunakan Laravel. *Source code* yang digunakan untuk *generate* fitur ini yaitu: `php artisan make:seeder`.

7. *Database/Factories*

Database/factories digunakan untuk membuat data–data *fake* atau data *dummy*. Data *dummy* biasa digunakan saat mengembangkan aplikasi tahap awal dengan Laravel.

8. *Database/Migrations*

Sub folder database/migrations digunakan untuk mengelola aneka tabel di *database*, mulai dari membuat tabel, mengedit table menambah kolom, menghapus kolom, menambah table, hingga menghapus tabel itu sendiri. Fitur ini bisa dijalankan dengan perintah yaitu: `php artisan make:migration`.

9. *Public dan Private*

Direktori publik berfungsi untuk memungkinkan semua *resource styling* (seperti koding JavaScript, gambar, CSS, dan kebutuhan lain untuk *styling*) dapat diakses melalui *browser web*. Sehingga otomatis, *direktori publik* terletak pada *web server*. Sementara *private* adalah *direktori* bawaan Laravel yang memuat seluruh kode PHP bawaan *system* dan buatan *user* untuk digunakan dalam pekerjaan yang bersifat lebih rahasia.

10. Storage

Storage adalah *folder* penyimpanan di Laravel. Hal yang disimpan pada *storage* adalah aneka *progress* dan proses yang dilakukan dengan Laravel. *Storage* digunakan oleh *user* untuk menyimpan *backup* dari *database*, *backup file* Laravel membentuk index dari Laravel Scout dan menyimpan aneka *file logs error*.

11. Resources/View

Folder view merupakan bagian dari konsep MWC. Seluruh *file* yang dibuat di Laravel, seperti *template blade* dan *file HTML* dapat dilihat memakai *sub folder view*.

12. Resources/Lang

Lang adalah *sub folder direktori* yang *mensupport* implementasi *localization* dan menyimpan semua definisi bahasa yang dibuat. Dengan adanya *subfolder* ini, *user* dapat membuat berbagai web atau aplikasi yang di *support* dengan fasilitas multi bahasa.

13. Resources/Assets

Assets digunakan untuk menyimpan *resources* yang dikerjakan *user* pada proses *compile less file*, *coffeescript* dan *saas*. Ketika *resources* disimpan di *assets*, maka *file* dapat *decompile* otomatis oleh *system* Laravel memakai *gulp*. Terakhir, *file* yang sudah diproses akan disimpan pada sebuah *direktori publik* untuk digunakan. *Sub folder* ini umumnya digunakan oleh *user* yang memakai Laravel mix.

14. Config

Config berfungsi untuk menyimpan aneka konfigurasi core Laravel dan tambahan *package* untuk Laravel. Misalnya adalah file *app.php* untuk mendaftarkan *package* tambahan di Laravel yang tersimpan pada folder *config*.

15. Routes dan Vendor

Routes berfungsi untuk menyimpan seluruh jalur *route* yang dibuat dari *request* ke *controller* disimpan bersama – sama dengan *API routes*. Sedangkan *vendor* berisi seluruh *file composer package* yang digunakan saat proses *development web*. Kedua *folder routes* dan *vendor* tidak memiliki *sub folder direktori*.

Selain framework Laravel yang berbasis bahasa pemrograman PHP, ada framework CodeIgniter yang menggunakan bahasa pemrograman PHP. Perbedaan keduanya seperti pada gambar 2.2 Perbandingan Laravel dan CodeIgniter.

| FEATURES | | |
|------------------------|---------|-------------|
| | LARAVEL | CODEIGNITER |
| ORM | ✓ | ✗ |
| ACTIVE RECORD | ✓ | ✓ |
| PACKAGED MODULE | ✓ | ✗ |
| ROUTING | ✓ | ✓ |
| INBUILT USER AUTH | ✓ | ✓ |
| INBUILT CAPTCHA | ✗ | ✓ |
| TEMPLATE ENGINE | ✓ | ✗ |
| CODE GENERATION | ✓ | ✗ |
| NAMESPACES | ✓ | ✗ |
| INBUILT CLI SUPPORT | ✓ | ✗ |
| LOG MANAGEMENT | ✓ | ✗ |
| INCLUDED JS LIBRARY | ✗ | ✓ |
| HHVM SUPPORT | ✓ | ✗ |
| HMVC SUPPORT (BONFIRE) | ✗ | ✓ |
| COMMUNITY SUPPORT | ✓ | ✓ |

Gambar 2.2 Perbandingan Laravel dan CodeIgniter

(Sumber : <https://www.codeclouds.com/blog/laravel-vs-codeigniter-a-difficult-choice/>)

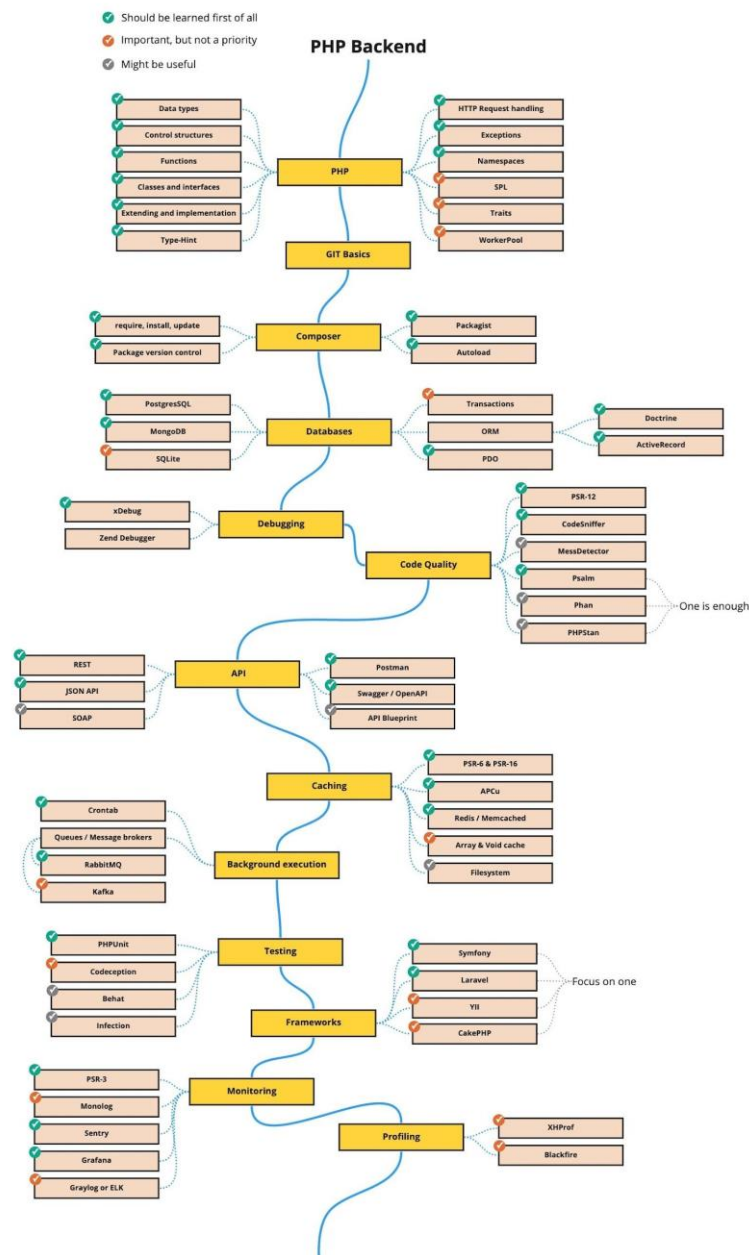
2.5 PHP

Pada penelitian ini, dalam membangun sebuah aplikasi untuk perusahaan kelapa sawit menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP). PHP atau singkatan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman yang *open source* dalam pengembangan web yang terdapat dalam *script Hypertext Markup Language* (HTML). PHP adalah bahasa *script* sisi server yang pemrosesan data berlangsung di server, sederhananya adalah server yang mengkompilasi ulang *script* program kemudian hasilnya dikirim ke klien yang melakukannya. Konsep lain dari PHP adalah singkatan dari kata tersebut *Hypertext Preprocessor*, bahasa pemrograman berdasarkan kode (*script*) yang digunakan. memproses data dan mengirimkannya kembali ke jaringan ke dalam kode HTML *browser*.

Pada dasarnya server berfungsi jika ada spesifikasi pelanggan. Dalam hal ini, *client* menggunakan Kode PHP untuk mengirim permintaan ke *server*. Sistem kerja PHP diawali dengan permintaan *request* berasal dari web *browser* berdasarkan URL atau alamat situs web di internet. Kemudian web *server* mencari file yang diminta dan menampilkan kontennya di *browser*. Pada Prinsipnya sama seperti saat memanggil kode *Hypertext Markup Language* (HTML), tetapi ketika permintaan dikirim ke *server* web, *server web* mengirimkannya dan memeriksa jenis file yang diminta oleh pengguna. Jika tipe file yang diminta PHP adalah, memeriksa isi *script* dari sisi PHP. Jika file tersebut tidak berisi *script* PHP, permintaan pengguna ditampilkan langsung di browser, tetapi jika file tersebut berisi *script* PHP maka proses dilanjutkan dengan modul PHP sebagai engine ini kompilasi *script* PHP dan proses *script*, sehingga kemudian dapat diubah menjadi kode-kode HTML ditampilkan di browser pengguna.

Sebelum membuat program aplikasi, dapat dibuat sebuah roadmap dalam menentukan langkah awal dalam meng-*coding* sebuah program yang dapat di *execute* pada web *browser* seperti pada Gambar 2.3 Roadmap PHP. Langkah awal yaitu membuat *database*, konfigurasi *database* dengan halaman *user*, *authentication user*, *debugging*, *background extraction*, *testing*, *monitoring*, *profiling*, dan mengambil data dari *database* yang tersedia untuk ditampilkan pada

halaman *user* yang bersifat informatif sesuai dengan tujuannya yaitu *user* dapat men – *tracking* atau melacak atau menelusuri asal – usul dari *crude palm oil* (CPO). Data CPO didapatkan dari *stakeholder* yang melakukan proses komunikasi atau menginput data atau melakukan proses order pada tiap – tiap rantai yang menjadi satu blok dalam sistem. Sehingga data yang ditampilkan pada web *browser* pengguna bersifat *real-time* dan akan bertambah seiring data tersebut masuk ke dalam sistem.



Gambar 2.3 Roadmap PHP

(Sumber : <https://dev.to/he110/php-backend-roadmap-2j1a>)

2.6 Visual Studio Code

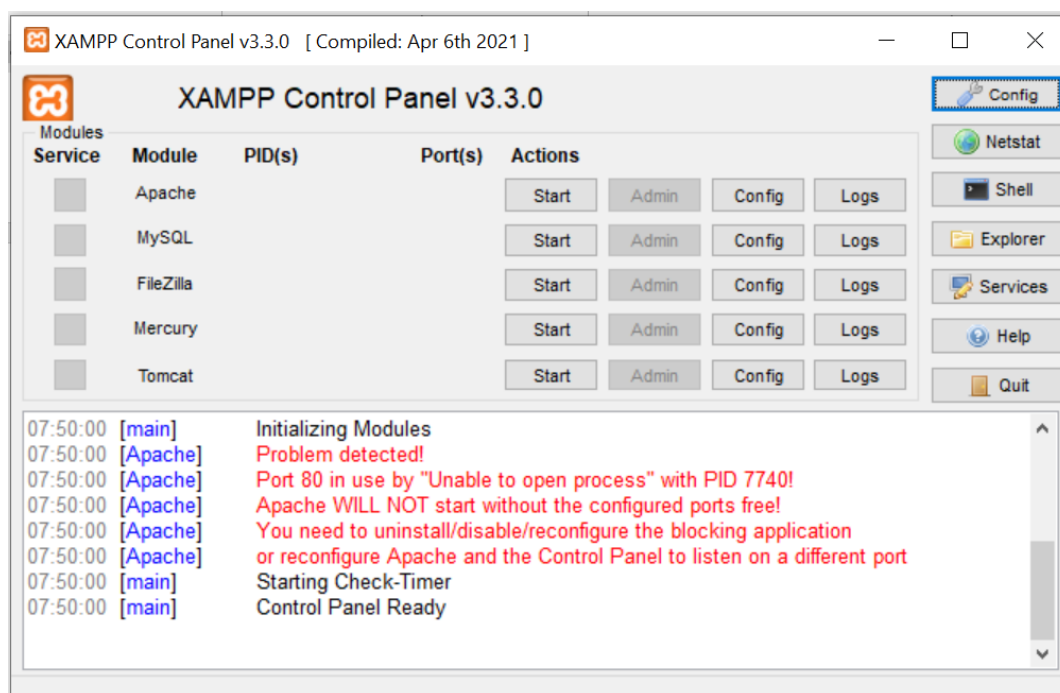
Visual Studio Code merupakan *text editor* yang digunakan penulis untuk membangun aplikasi. *Visual Studio Code* berbasis *opensource* yang dapat dijalankan pada *Operating System* (OS) Windows, MacOS, dan Linux. *Text editor* ini juga memudahkan dalam penulisan *code* yang mendukung beberapa jenis pemrograman, seperti C++, C#, Java, Python, PHP, GO. *Visual Studio Code* memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian *code* tersebut. Selain itu, *Visual Studio Code* sudah terintegrasi dengan *github* dan juga memiliki fitur untuk menambah *ekstensi* pada *text editor* tersebut.

2.7 XAMPP

XAMPP adalah sebuah *local server* yang digunakan penulis sebagai *tools* untuk mengeksekusi code program aplikasi secara *run-time* pada web *browser*. XAMPP bersifat *open source* berbasis web *server* yang berisi berbagai program. Aplikasi ini mendukung berbagai sistem operasi seperti Linux, Windows, MacOS dan Solaris. XAMPP bekerja sebagai *server/host* lokal yang sudah menyertakan program Apache, MySQL dan PHP. Kelahiran XAMPP dimulai dengan kesulitan menginstal Apache dan menambahkan dukungan PHP dan MySQL. Hal inilah yang kemudian menyebabkan lahirnya XAMPP sebagai aplikasi yang memudahkan pekerjaan developer yang membutuhkan web server di localhost hanya dengan satu aplikasi. Xampp terdiri dari singkatan program yang dikandungnya, antara lain yaitu X adalah tanda kode untuk perangkat lunak lintas *platform*. Ini berarti XAMPP berjalan di banyak sistem operasi umum. Seperti Windows, Linux dan Mac OS. A atau Apache adalah aplikasi *server* web yang bebas digunakan (bersifat *open source*). File dengan kode (bahasa pemrograman PHP) dapat dieksekusi di *localhost* oleh web server. M atau MySQL adalah aplikasi *database server* yang menggunakan bahasa pemrograman SQL (*Structured Query Language*). Tugasnya adalah mengelola data secara terstruktur dan sistematis. MySQL dapat berjalan di

localhost tanpa koneksi internet, memungkinkan pengembang dan pemrograman membuat aplikasi berbasis web di komputer mereka.

P atau PHP adalah bahasa pemrograman sisi *server* yang dapat digunakan untuk mengelola konten dinamis dan database. PHP membuat halaman lebih dinamis. P atau Perl adalah bahasa pemrograman tingkat tinggi yang dapat digunakan untuk semua kebutuhan (*platform*). Saat ini, Perl banyak digunakan untuk mengembangkan aplikasi untuk server web.



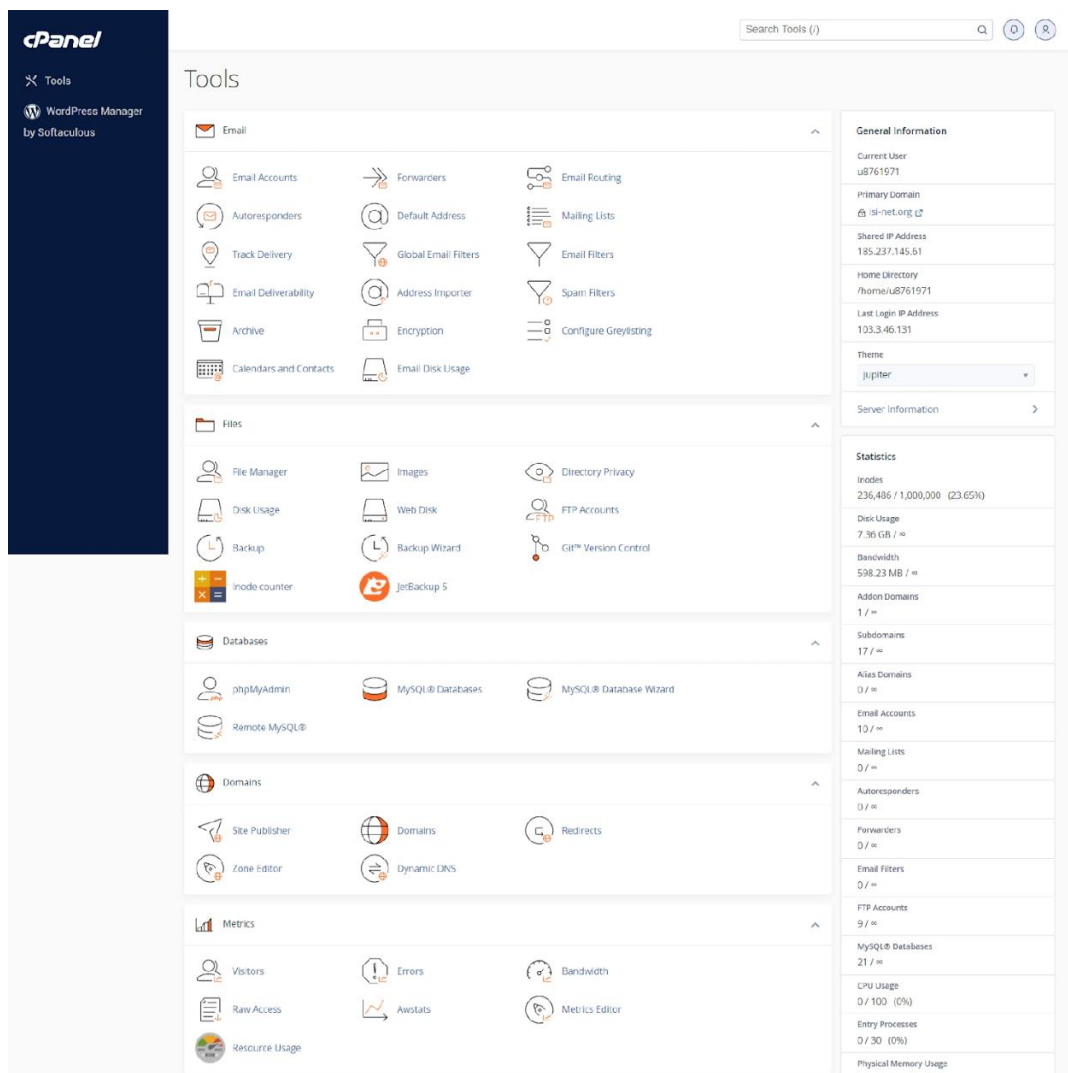
Gambar 2.4 XAMPP Control Panel

2.8 CPanel Hosting

Pada pembuatan aplikasi dilakukan perlisian aplikasi website melalui CPanel *Hosting* dengan alamat situs yaitu <https://telsa.isi-net.org/>. CPanel adalah platform manajemen *hosting* yang memiliki banyak fitur serta tampilan dasbor yang intuitif seperti pada Gambar 2.5. CPanel berfungsi sebagai panel kontrol yang memfasilitasi pengelolaan layanan *hosting* baik *server* maupun situs web. Saat ini CPanel adalah salah satu panel kontrol *hosting* yang paling banyak digunakan.

CPanel adalah panel kontrol *hosting* yang menyederhanakan pengelolaan *hosting* seperti *server*, email, dan situs web.

CPanel memiliki tampilan *dashboard* yang intuitif dan ramah. Selain itu, CPanel juga memiliki beberapa fitur yang memudahkan pengelolaan website lain seperti domain, email, instalasi CMS dan lainnya. CPanel adalah platform hosting berbasis Linux, sehingga *control* panel ini hanya dapat digunakan dengan sistem operasi berbasis Linux seperti Alma Linux, CentOS, CloudLinux dan Ubuntu. Beberapa pengguna terutama menggunakan Windows dan tentu saja belum bisa menggunakan CPanel.



Gambar 2.5 Dashboard CPanel

2.9 Software Development Life Cycle

Dalam sebuah *project* pembuatan aplikasi berbasis website maupun mobile, penulis memerlukan sebuah manajemen proses dalam *project* tersebut agar hasil yang diharapkan sesuai dengan tujuan yang ingin dicapai. Manajemen proses yang digunakan yaitu SDLC atau *Software Development Life Cycle* adalah serangkaian proses sistematis pembuatan *software* untuk mengembangkan aplikasi dan website dengan kualitas terbaik, biaya yang efisien, dan waktu yang efektif [11]. SDLC terdiri dari rencana terperinci yang menjelaskan bagaimana merencanakan, membangun, dan memelihara software.

SDLC memiliki peranan penting dalam proses *development* produk, yaitu

1. Memetakan perencanaan, penjadwalan, dan estimasi proyek secara rinci
2. Menyediakan *framework* untuk serangkaian aktivitas dan deliverables
3. Dapat digunakan sebagai mekanisme *men-tracking* dan mengontrol proyek
4. Meningkatkan visibilitas perencanaan proyek kepada semua *stakeholder* yang terlibat dalam proses *developing*
5. Meningkatkan kecepatan proses *developing*
6. Membantu untuk mengurangi resiko proyek dan *overhead* rencana manajemen proyek

SDLC memiliki beberapa metode dalam *software engineering*, diantaranya:

1. *Waterfall*

Pendekatan ini dilakukan secara *linear-sekuensial*. Dalam metode SDLC *Waterfall*, hasil dari satu fase menjadi input untuk fase berikutnya. Metode *waterfall* juga bersifat *documentation-intensive*. Karena, pada fase awal banyak mendokumentasikan apa saja yang perlu dilakukan pada fase berikutnya.

2. *Agile*

Prinsip yang *iteratif* atau berulang. Metode *Agile* membagi seluruh proyek menjadi siklus atau *build incremental* yang lebih kecil dan disediakan dalam iterasi dari satu sampai tiga minggu. Metode *Agile* menggunakan praktek yang

mendorong interaksi berkelanjutan dari pengembangan dan pengujian selama proses SDLC, sehingga model ini bisa *deliver* produk dengan sangat cepat

3. *Incremental*

Pada dasarnya metode *Incremental* adalah rangkaian dari model *Waterfall*. Pada awal rangkaian ini, *requirement* akan dibagi menjadi beberapa kelompok untuk mengembangkan *software*. Proses *Incremental* akan berulang, sehingga akan menghasilkan lebih banyak fungsionalitas hingga semua *requirement* terpenuhi dari seluruh bagian.

4. Metode V

Pengujian dan pengembangan metode V direncanakan secara parallel. Metode V memberikan fase verifikasi dan fase validasi dan model ini akan bergabung dengan fase *coding*.

5. *Spiral*

Metode *Spiral* adalah kombinasi dari *Rapid Prototyping and Concurrency* dalam desain dan pengembangan yang bersifat *risk-driven*. Metode ini mengadopsi elemen dari satu atau lebih metode SDLC lainnya seperti *Waterfall, Incremental, Agile*, dan lainnya yang merupakan fitur terbaik.

6. *Big Bang*

Metode ini berfokus pada semua *resource* yang ada dengan *requirement* yang minim atau bahkan tanpa perencanaan. Metode ini sangat ideal untuk proyek-proyek kecil dengan tim *developer* dengan ukuran lebih kecil. *Big Bang* adalah metode yang juga ideal. Ketika *requirement* tidak diketahui atau tanggal rilis final tidak diberikan.

2.10 Scrum

Manajemen proses pembuatan aplikasi selain menggunakan metode SDLC tetapi juga menggunakan metode *scrum*. *Scrum* merupakan metode untuk mengatur manajemen dan pelaksanaan proyek. *Scrum* dapat digunakan untuk pengembangan produk apapun yang bersifat kompleks dan senantiasa berubah [12]. *Scrum* membantu koordinasi tim agar lebih terstruktur dan menguatkan komunikasi antar anggota tim. *Scrum* berguna untuk mempercepat rilis produk kepada *user* dengan

produktivitas dan kualitas yang tinggi. Selain itu, *Scrum* juga dapat membantu pengerjaan proyek dengan perhitungan *cost* yang lebih rendah disesuaikan dengan pertimbangan kepuasan *user* sebagai fokus utama [11].

Hal utama dari *Scrum* adalah *Sprint*, dimana dalam *Sprint* sejumlah pekerjaan yang sudah direncanakan harus diselesaikan oleh tim dan dipersiapkan untuk dilakukan proses *review*. *Sprint* adalah aktivitas pekerjaan yang harus diselesaikan oleh tim *Scrum* dalam periode waktu yang telah ditentukan dengan durasi selama satu bulan. Tujuan dari *Sprint* adalah untuk memecah proyek menjadi potongan-potongan berukuran kecil. Hal ini memungkinkan tim untuk merencanakan satu *Sprint* pada satu waktu dan menyesuaikan *Sprint* di masa mendatang berdasarkan hasil *Sprint* yang sudah selesai [11].

Sprint juga dapat disebut *work cycle* dalam pengerjaan *Scrum*.

Sprint terdiri dari:

1. *Sprint Planning Meeting*

Proses durasi pengerjaan perencanaan *Sprint* harus sebanding dengan proses lamanya *Sprint*. *Meeting* ini yang menentukan tujuan dari *Sprint*

2. *Daily Scrum*

Daily Scrum dilakukan untuk mengulas *Backlog* yang sudah selesai dan membahas masalah atau kendala di hari sebelumnya

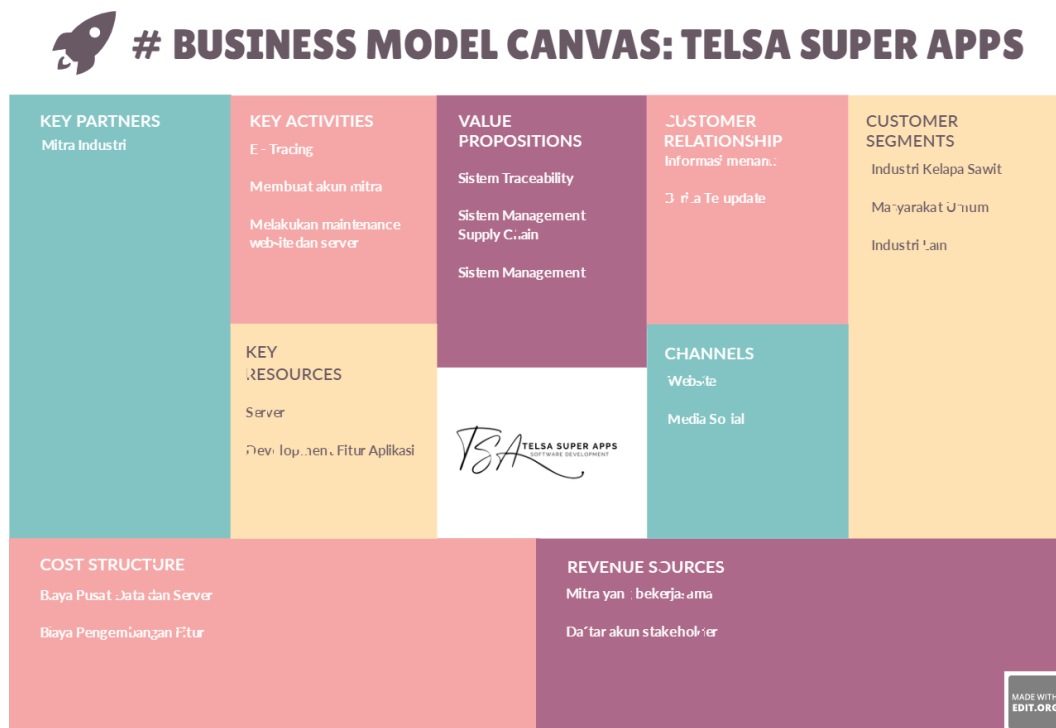
3. *Sprint Review*

Sprint Review dilakukan untuk memeriksa hasil *Sprint* dan menentukan penyesuaian di masa depan. Hasil dari *Sprint Review* adalah revisi yang berisikan item *Product Backlog* untuk *Sprint* berikutnya. *Product Backlog* juga dapat disesuaikan secara keseluruhan untuk menemukan peluang baru.

4. *Sprint Retrospective*

Dalam *Sprint Retrospective* akan dilakukan perencanaan untuk meningkatkan kualitas dan efektivitas. Proses akhir dari *Sprint Retrospective*, tim *Scrum* sudah mengidentifikasi peningkatan yang bisa diimplementasikan pada *Sprint* berikutnya.

2.11 Business Model Canvas



Gambar 2.6 Business Model Canvas

Business Model Canvas yang dibuat untuk rancangan aplikasi, kemudian di implementasi pada manajemen rantai pasok perusahaan kelapa sawit seperti pada Gambar 2.6 Aplikasi ini memiliki segmentasi pasar yaitu industri kelapa sawit, masyarakat umum, dan industri lain. Cara aplikasi membangun hubungan dengan pelanggan yaitu memberikan informasi yang menarik seputar sawit dan berita terupdate tentang pergerakan sistem distribusi tandan buah segar dari hulu hingga ke hilir. Keunggulan aplikasi ini yaitu menghadirkan sistem *traceability*, sistem manajemen *supply chain*, dan sistem manajemen umum. Aktivitas yang menjadi kunci dari aplikasi ini yaitu *E-tracking*, membuat akun mitra, dan melakukan *maintenance* website dan server. Pemasukan yang didapat dari aplikasi berasal dari mitra yang bekerjasama dengan menggunakan aplikasi dan biaya pendaftaran akun stakeholder, sedangkan biaya yang dikeluarkan untuk keperluan biaya pusat data dan server serta biaya pengembangan fitur aplikasi.

2.12 Stakeholder

Pada penelitian ini, membangun sebuah sistem dengan proses komunikasi antar *stakeholder*. *Stakeholder* terkait yaitu *stakeholder* petani, *stakeholder* kebun, *stakeholder* logistik, dan *stakeholder* pabrik. Pada tiap-tiap *stakeholder* memiliki master data, dan *session* (*objective* dan *features*) dalam sebuah sistem aplikasi seperti pada analisis kebutuhan atau *requirements project* berikut.

a. *Requirements/Analisis Kebutuhan*

1. *Portal Super Admin*

Portal super admin merupakan kumpulan fitur yang mencakup fungsi dari proses sistem yang dilakukan oleh super admin.

a. *Otentikasi Pengguna*

Otentikasi merupakan spesifikasi yang mencakup proses *login*, perubahan *profile*, *password* dan *logout* yang dilakukan oleh pengguna (*super admin*) melalui *portal admin* untuk menambah akun pengguna.

Tabel 2.1 *Otentikasi Pengguna*

| | |
|------------------|---|
| <i>Objective</i> | <ol style="list-style-type: none"> 1. <i>Super admin</i> dapat melakukan proses <i>login</i>, <i>logout</i>, perubahan data <i>profile</i> dan perubahan <i>password</i> 2. <i>Super admin</i> dapat menambah akun pengguna |
| <i>Features</i> | <ol style="list-style-type: none"> 1. <i>Autentifikasi login</i> menggunakan <i>username</i> dan <i>password</i> 2. Menampilkan data <i>profile</i> pengguna 3. Perubahan data <i>profile</i> pengguna 4. Perubahan data <i>password</i> pengguna 5. Menambah data <i>login</i> pengguna 6. <i>Approval data</i> pengajuan order 7. Menampilkan data pengajuan order |

Tabel 2.2 *Fields Item Otentikasi Pengguna*

| <i>Fields</i> | <i>Type</i> | <i>Restrictions</i> | <i>Default Value</i> | <i>Remarks</i> |
|----------------------|---------------------|---------------------|--|----------------|
| <i>username</i> | <i>varchar (25)</i> | <i>required</i> | | |
| <i>nama</i> | <i>varchar (25)</i> | <i>required</i> | | |
| <i>telepon</i> | <i>varchar (25)</i> | <i>required</i> | | |
| <i>hak akses</i> | <i>varchar (25)</i> | <i>required</i> | | |
| <i>alamat</i> | <i>varchar (25)</i> | <i>required</i> | | |
| <i>login sebagai</i> | <i>enum</i> | <i>required</i> | Petani, Kebun, Logistik, Pabrik | |
| <i>Foto</i> | <i>text</i> | <i>required</i> | | |

* Data yang ditampilkan pada tabel merupakan data yang dapat *diedit* oleh *super admin*.

2. Master Data

Master data merupakan modul data utama yang digunakan sebagai acuan untuk mendukung proses pengelolaan data pada sistem.

1. Master Data Petani

Master data kebun merupakan sub modul yang mengelola data kebun meliputi *fields* yang tersedia.

Tabel 2.3 *Master Data Petani*

| | |
|------------------|---|
| <i>Objective</i> | Akun petani dapat mengajukan order |
| <i>Features</i> | <ol style="list-style-type: none"> 1. Pengajuan order 2. Proses approval dari <i>stakeholder</i> kebun, <i>stakeholder</i> rlogistik, <i>stakeholder</i> pabrik 3. Keterangan rejected order dari <i>stakeholder</i> kebun, <i>stakeholder</i> logistik, <i>stakeholder</i> pabrik |

Tabel 2.4 *Fields Item Master Data Transaksi Petani*

| <i>Fields</i> | <i>Type</i> | <i>Restrictions</i> | <i>Default Value</i> | <i>Remarks</i> |
|--------------------------|---------------------|---------------------|----------------------|----------------|
| id_petani | <i>varchar (25)</i> | <i>primary key</i> | | |
| tanggal pengajuan | <i>date</i> | <i>required</i> | | |
| tanggal panen | <i>date</i> | <i>required</i> | | |
| Titik koordinat area | <i>varchar (25)</i> | <i>required</i> | | |
| area | <i>varchar (25)</i> | <i>required</i> | | |
| <i>Stakeholder</i> kebun | <i>enum</i> | <i>required</i> | akun kebun | |

2. Master Data Kebun

Master data kebun merupakan sub modul yang mengelola data kebun meliputi fields yang tersedia.

Tabel 2.5 *Master Data Kebun*

| | |
|------------------|---|
| <i>Objective</i> | Akun kebun dapat menyetujui order dari akun petani dan melanjutkan order tersebut |
| <i>Features</i> | <ol style="list-style-type: none"> 1. <i>Approval/Rejected</i> order 2. Penambahan data order 3. Proses approval dari <i>stakeholder</i> logistik, <i>stakeholder</i> pabrik 4. Keterangan rejected order dari <i>stakeholder</i> logistik, <i>stakeholder</i> pabrik 5. Semua data yang menampilkan seluruh order yang <i>record</i> oleh sistem portal 6. Beranda yang menampilkan informasi menunggu <i>approved</i> dan informasi order yang di <i>rejected</i> |

Tabel 2.6 *Fields Item Master Data Transaksi Kebun*

| <i>Fields</i> | <i>Type</i> | <i>Restrictions</i> | <i>Default Value</i> | <i>Remarks</i> |
|------------------------|---------------------|-----------------------|----------------------|----------------|
| id_kebun | <i>varchar (25)</i> | <i>primary key</i> | | |
| titik koordinat kebun | <i>text</i> | <i>required</i> | | |
| legalitas lahan | <i>enum</i> | <i>required</i> | SKGR, SKT, SHM | |
| status sertifikasi | <i>enum</i> | <i>required</i> | ISPO, RSPO | |
| stakeholder logistik | <i>enum</i> | <i>required</i> | Akun logistik | |
| tanggal approval kebun | <i>date</i> | <i>auto increment</i> | | Waktu approval |
| aksi | <i>enum</i> | <i>required</i> | Approved, Rejected | |
| keterangan reject | <i>text</i> | <i>required</i> | | |

3. Master Data Logistik

Master data logistik merupakan sub modul yang mengelola data logistik meliputi fields yang tersedia.

Tabel 2.7 *Master Data Logistik*

| | |
|------------------|--|
| <i>Objective</i> | Akun logistik dapat menyetujui order dari akun kebun dan melanjutkan order tersebut |
| <i>Features</i> | <ol style="list-style-type: none"> 1. <i>Approval/Rejected</i> order 2. Penambahan data order 3. Proses approval dari <i>stakeholder</i> pabrik 4. Semua data yang menampilkan seluruh order yang <i>ter-record</i> oleh sistem portal 5. Beranda yang menampilkan informasi menunggu <i>approved</i> dan informasi order yang di <i>rejected</i> |

Tabel 2.8 *Fields Item Master Data Transaksi Logistik*

| <i>Fields</i> | <i>Type</i> | <i>Restrictions</i> | <i>Default Value</i> | <i>Remarks</i> |
|---------------------------|--------------------|---------------------|----------------------|----------------|
| jumlah truk | <i>integer (8)</i> | <i>required</i> | | |
| truk tersedia | <i>integer (8)</i> | <i>required</i> | | |
| <i>stakeholder</i> pabrik | <i>enum</i> | <i>required</i> | Akun pabrik | |
| tanggal approval logistik | <i>date</i> | | | Waktu approval |
| aksi | <i>enum</i> | <i>required</i> | Approved, Rejected | |
| keterangan reject | <i>text</i> | <i>required</i> | | |

4. Master Data Pabrik

Master data pabrik merupakan sub modul yang mengelola data pabrik meliputi fields yang tersedia.

Tabel 2.9 *Master Data Pabrik*

| | |
|------------------|--|
| <i>Objective</i> | Akun pabrik dapat menyetujui order dari akun logistik dan melengkapi data <i>traceability</i> CPO |
| <i>Features</i> | <ol style="list-style-type: none"> 1. <i>Approval/Rejected</i> order 2. <i>Approval/Rejected</i> panen 3. Penambahan data order 4. Semua data yang menampilkan seluruh order yang <i>ter-record</i> oleh sistem portal 5. Beranda yang menampilkan informasi menunggu <i>approved</i> dan informasi order yang di <i>rejected</i> |

Tabel 2.10 *Fields Item Master Data Transaksi Pabrik*

| <i>Fields</i> | <i>Type</i> | <i>Restrictions</i> | <i>Default Value</i> | <i>Remarks</i> |
|---------------------------|--------------------|---------------------|----------------------|----------------|
| tanggal approval logistik | <i>date</i> | | | Waktu approval |
| total TBS ditimbang | <i>integer (8)</i> | <i>required</i> | | |
| aksi | <i>enum</i> | <i>required</i> | Approved, Rejected | |
| keterangan reject | <i>text</i> | <i>required</i> | | |

5. Master Data Pesanan

Master data pesanan merupakan sub modul yang mengelola data pesanan meliputi fields yang tersedia.

Tabel 2.11 *Fields Item Master Data Dasar Pesanan*

| <i>Fields</i> | <i>Type</i> | <i>Restrictions</i> | <i>Default Value</i> | <i>Remarks</i> |
|-------------------|--------------------|---------------------|----------------------|----------------|
| Kode | <i>integer (8)</i> | <i>required</i> | Auto Increment | |
| Rencana panen tbs | <i>integer (8)</i> | <i>required</i> | | |
| Tanggal panen | <i>date</i> | <i>required</i> | | Waktu panen |
| Tanggal pengajuan | <i>date</i> | <i>required</i> | | Waktu approval |
| Approval kebun | <i>date</i> | <i>required</i> | | Waktu approval |
| Approval logistik | <i>date</i> | | | Waktu approval |
| Approval pabrik | <i>integer (8)</i> | <i>required</i> | | |

Tabel 2.11 *Fields Item Master Data Dasar Pesanan (Lanjutan)*

| <i>Fields</i> | <i>Type</i> | <i>Restrictions</i> | <i>Default Value</i> | <i>Remarks</i> |
|--------------------------|--------------------|---------------------|----------------------|----------------|
| Truk yang dikirim | <i>enum</i> | <i>required</i> | Approved, Rejected | |
| Total TBS yang ditimbang | <i>integer (8)</i> | <i>required</i> | Auto Increment | |
| Status pesanan | <i>integer (8)</i> | <i>required</i> | | |
| Keterangan reject | <i>integer (8)</i> | <i>required</i> | | |
| Approval super admin | <i>integer (8)</i> | <i>required</i> | | |

3. Roles dan Permissions

Roles dan Permissions merupakan fitur untuk *super admin* dalam mengatur dari masing – masing *admin stakeholder*.

Tabel 2.12 *Roles dan Permissions*

| <i>Roles</i> | <i>Permissions</i> |
|----------------------|---|
| <i>Super Admin</i> | Memiliki semua hak akses pada seluruh fitur yang tersedia |
| <i>Akun Petani</i> | 1. Pengajuan order |
| <i>Akun Kebun</i> | 1. Approval kebun 2. Memilih logistik yang digunakan |
| <i>Akun Logistik</i> | 1. Approval logistik 2. Menambah data truk yang dikirim |
| <i>Akun Pabrik</i> | 1. Approval pabrik 2. Menambah data tandan buah segar yang ditimbang |

2.13 Indikator Keberhasilan

Platform aplikasi yang dibangun atau dibuat harus melalui tahap analisis melalui berbagai cara yang dilakukan untuk mengukur kinerja yang diperoleh dari model aplikasi selama pengembangan. Implementasi pengujian sistem *traceability* untuk industri kelapa sawit dilakukan dengan menggunakan metode *blackbox testing*. Metode *blackbox testing* merupakan salah satu tahapan pengujian perangkat lunak dengan memprioritaskan pengujian persyaratan fungsional program.

Pengujian dengan *blackbox testing* ini dirancang untuk menemukan bug atau masalah dalam program. Teknik *black box testing* yang akan diterapkan oleh peneliti yaitu *all-pair testing* atau *pairwise testing* dengan menguji semua kemungkinan atau *probabilitas* kombinasi dari seluruh hubungan antar bagian aplikasi berdasarkan *input* parameternya. Jika hasil pengujian parameter lebih didominasi nilai yang berhasil, maka nilai pengujian *black box testing* nya baik.

Pengujian selanjutnya dilakukan dengan *assessment* proses pengembangan dan *assessment* kepuasan *user*. *Assessment* ini bertujuan sebagai bahan evaluasi *developer* dalam *maintenance* aplikasi, sehingga aplikasi ini dapat berjalan dengan *sustain* atau keberlanjutan dan fitur–fitur yang ada saat ini akan terus berkembang dan akan ada proses perbaikan atau penambahan fitur yang diperlukan di masa yang akan datang.

BAB 3

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian dimulai pada Agustus 2022 sampai dengan Januari 2023. Penelitian ini dilakukan di Laboratorium Terpadu Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

3.2 Perangkat Penelitian

Adapun perangkat pada penelitian tugas akhir ini adalah sebagai berikut:

3.2.1 Perangkat Keras (*Hardware*)

1. Satu unit sebagai media perancangan sistem aplikasi berbasis website yang dibangun dengan sistem operasi Windows 10 home 64-bit, dengan spesifikasi minimum:
 - a. Processor Intel Core i7 11th Gen
 - b. VGA Nvidia Geforce 930 MX,
 - c. RAM 16 GB
 - d. SSD NVME berkapasitas 512GB

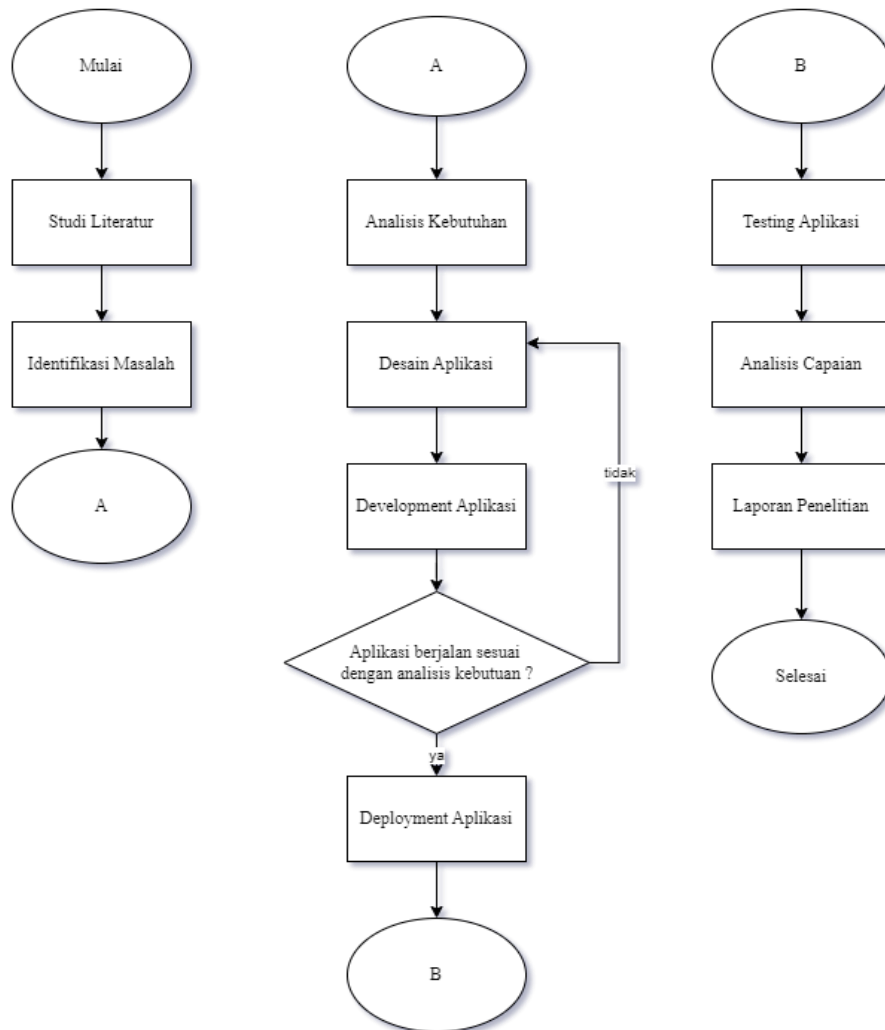
3.2.2 Perangkat Lunak (*Software*)

1. *Software Visual Studio Code* sebagai text editor dan *running script program*
2. *Xampp* sebagai *control panel localhost*
3. *Cpanel Hostinger* sebagai perilisasi Aplikasi

3.3 Metode Penelitian

Metode penelitian yang digunakan adalah metode experimental secara kualitatif dan kuantitatif. Alur penelitian ini terdiri dari studi literatur, identifikasi masalah, perancangan dan pengembangan aplikasi yang terdiri dari analisis kebutuhan, desain aplikasi, *development* aplikasi dan *deployment* aplikasi serta melakukan proses pengujian aplikasi dengan melakukan testing aplikasi dan analisis capaian. Kegiatan penelitian diakhiri dengan membuat laporan penelitian sebagai bahan rekomendasi bacaan dan menambah perspektif baru terhadap subjek tema

penelitian. Langkah-langkah penelitian yang dicakup dalam ruang lingkup penelitian ditunjukkan pada Gambar 3.1



Gambar 3.1 Diagram Alir Penelitian

1. Studi Literatur

Studi literatur digunakan penulis untuk mempelajari dan mengumpulkan data mengenai proses distribusi Tandan Buah Segar (TBS) hingga menjadi produk *Crude Palm Oil* (CPO). Selain itu, penulis juga mempelajari konsep *blockchain* yang akan diimplementasikan pada platform *traceability* dan dapat diterapkan pada aplikasi sebagai manajemen rantai pasok perusahaan kelapa sawit.

2. Identifikasi Masalah

Identifikasi masalah yang didapat penulis yaitu pada rantai pasok industri kelapa sawit dimulai dari TBS dipanen dari kebun petani kemudian dikumpulkan koperasi/kelompok petani. Setelah itu TBS dijemput oleh agen kecil, yang mendistribusikannya ke agen besar, lalu ke pemegang *delivery order* (DO). Setelah itu, TBS sampai ke pabrik kelapa sawit. Rantai pasok ini cenderung sulit untuk ditelusuri asal-usul nya sehingga menyebabkan adanya indikasi yang terjadi di industri kelapa sawit adalah kurangnya visibilitas dan akuntabilitas dalam transfer pasokan bahan baku dari petani ke pabrik, ke produsen, ke gerai ritel dan akhirnya ke konsumen. Solusi untuk meningkatkan visibilitas dan transparansi produk CPO adalah dengan menjadikan tiap – tiap entitas atau *stakeholder* menjadi satu blok dalam sistem portal *traceability* dan sistem *dashboard stakeholder* aplikasi.

3. Analisis Kebutuhan

Analisis kebutuhan aplikasi dibangun berdasarkan identifikasi masalah yang terjadi dengan metode pengembangan/pembuatan aplikasi sistem *traceability* untuk perusahaan kelapa sawit.

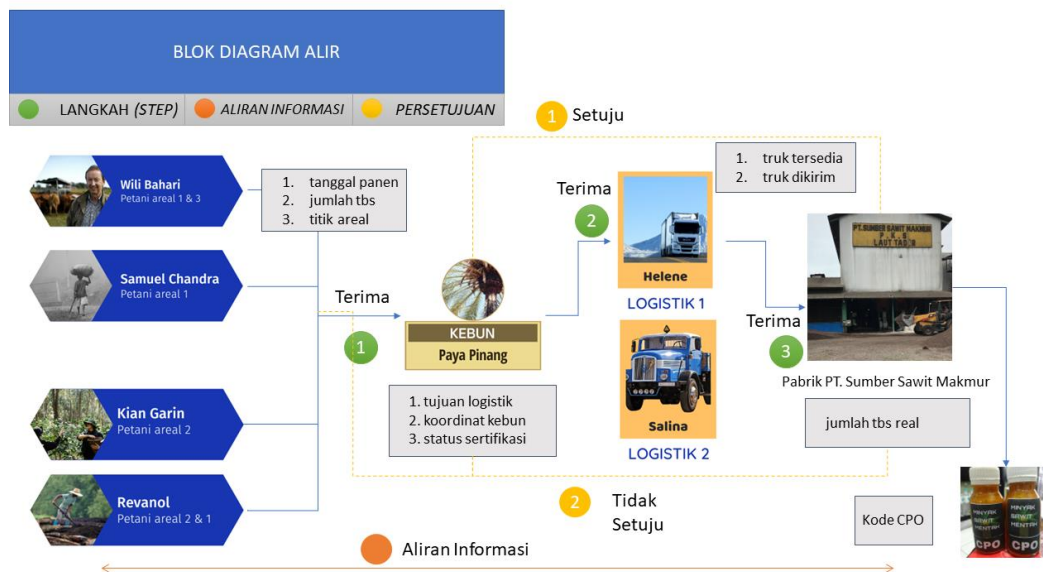
a. Alur Skenario Penelitian



Gambar 3.2 Titik Area *Loading Point*

Gambar 3.2 memperlihatkan skema sampel untuk area Kebun Bekri Lampung yang terdiri dari 8 lokasi *loading point* yang mewakili 8 *stakeholder* petani sebagai tempat pengumpulan tandan buah segar (TBS). TBS yang dipanen dari area panen dikumpulkan pada lokasi yang telah ditentukan (*loading point*). Selanjutnya TBS akan diangkut oleh truk logistik dari seluruh *loading point*, sesuai order yang telah dilakukan, menuju ke pabrik untuk diproses menjadi CPO.

b. Sistem *Dashboard Stakeholder*



Gambar 3.3 Sistem *Dashboard Stakeholder*

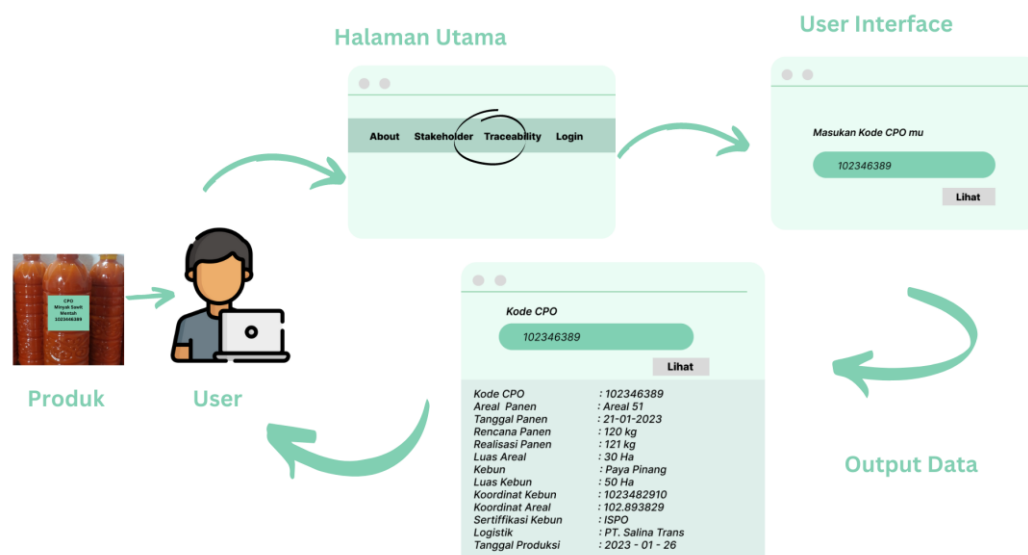
Sistem *dashboard stakeholder* seperti yang ditunjukkan pada Gambar 3.3 menjelaskan proses pengajuan order (proses pembuatan CPO dari TBS) yang dilakukan oleh seorang petani. Skema tersebut memudahkan perusahaan kelapa sawit untuk mengumpulkan seluruh petani menjadi satu kedalam bagian *stakeholder* kebun. Sebelum *stakeholder* melakukan pengajuan order, *stakeholder* terlebih dahulu mendaftarkan akun kepada super admin. Kemudian, jika sudah didaftarkan, maka *stakeholder* dapat mengajukan order. Data yang diajukan petani ke *stakeholder* kebun berupa tanggal panen, jumlah perkiraan TBS yang dipanen dan tempat pengumpulan TBS. Proses pengajuan panen ini sekaligus memberikan kode untuk produk CPO sebagai parameter untuk menelusuri asal-usul dari produk

CPO. Kode yang dihasilkan berasal dari proses fungsi sistem secara otomatis yang membangkitkan kode tersebut secara *random* karena kode tersebut menjadi identitas unik untuk masing-masing transaksi data. Data pengajuan panen dari petani terekam di *database* dan ter-*display* ke dalam sistem portal milik *stakeholder* kebun dan *stakeholder* rpabrik. Hal tersebut bertujuan untuk memberikan validasi dan persetujuan bahwa petani diperbolehkan melakukan aktivitas panen dan hasil panen dapat diterima dan diproses di pabrik kelapa sawit.

Setelah petani melakukan aktivitas panen TBS, selanjutnya *stakeholder* kebun memilih untuk menyetujui atau tidak menyetujui order yang dilakukan oleh petani. Jika order disetujui, maka *stakeholder* kebun selanjutnya memilih *stakeholder* logistik yang sudah teregistrasi dalam sistem. Data yang dikirimkan oleh *stakeholder* kebun, direkam ke dalam *database* dan ter-*display* pada sistem *dashboard stakeholder* logistik. Kemudian, *stakeholder* logistik memasukan data truk yang digunakan untuk mengantarkan TBS dari kebun menuju pabrik. Setelah TBS berada di pabrik, dilakukan kembali proses penimbangan TBS untuk memvalidasi apakah perkiraan berat TBS yang diajukan oleh petani sesuai dengan data yang ditimbang di pabrik kelapa sawit. Data hasil penimbangan TBS di pabrik menjadi data final dari TBS yang dipanen. *Stakeholder* pabrik melengkapi data dalam sistem dengan menginput data TBS yang sudah ditimbang. Kode CPO yang dihasilkan oleh sistem menjadi label dari produk CPO maupun produk turunan nya.

Seluruh data yang diinput kedalam sistem portal *traceability* aplikasi sudah terekam (*record*) secara menyeluruh dan bisa tercatat secara terverifikasi dan tidak dapat diubah, sehingga membantu dalam mengurangi resiko penipuan atau kecurangan yang terjadi.

c. Sistem Portal Traceability



Gambar 3.4 Rancangan Sistem Portal Traceability

Sistem portal *traceability* seperti yang ditunjukkan pada Gambar 3.4 menjelaskan alur atau langkah dari tahapan untuk *user* dapat melakukan *tracking* produk CPO. *User* mengakses situs website <https://telsa.isi-net.org/> menuju fitur *traceability* yang terdapat pada *navigation bar* website. *User* melihat tampilan atau *user interface* berupa text yaitu “Kode CPO” sebagai informasi atau judul atau *heading* pada dokumen *Hypertext Markup Language* (HTML). *User interface* memiliki kolom input dengan *type* text dengan *class* yaitu *form-control* dan *placeholder* “masukan kode CPO mu”. *User* menginputkan data kode CPO yang tersedia pada *database* dan kode tersebut berasal dari produk CPO maupun produk turunannya. Data kode CPO yang tersedia berasal dari proses komunikasi atau order yang dilakukan dari masing-masing *stakeholder* atau entitas yang tergabung menjadi satu blok dalam sistem aplikasi.

Kemudian sistem melakukan proses *query*. *Query* adalah sekumpulan baris perintah yang diproses untuk mendapatkan informasi yang berasal dari *database*. *Query database* ini digunakan untuk memudahkan pengelolaan data yang tersedia pada

database. Pada sistem ketertelusuran CPO ini dilakukan dengan menggabungkan seluruh tabel yang ada pada *database* untuk mendapatkan seluruh informasi berdasarkan parameter kode CPO. Setelah sistem melakukan proses *query*, *developer* melakukan tugasnya dengan meng-*coding* script PHP untuk *get* data yang tersedia pada *database* dengan perintah berupa `<?php echo $kode;?>`.

Output data yang dihasilkan ketika *user* menginputkan data kode CPO yang tersedia yaitu berupa data kode CPO yang berasal dari master data masing-masing *stakeholder* yaitu area panen, tanggal panen, perkiraan panen TBS, total tandan buah segar, titik koordinat area panen, luas area, titik koordinat kebun, luas kebun, sertifikasi kebun, nama kebun, titik koordinat pabrik, dan logistik yang digunakan. Jika *user* menginputkan data kode CPO yang tidak tersedia, maka proses *query* dan *get* data tidak menampilkan data pada *database*. Hal ini terjadi karena *query* dan *get* data hanya menampilkan data yang tersedia pada *database*.

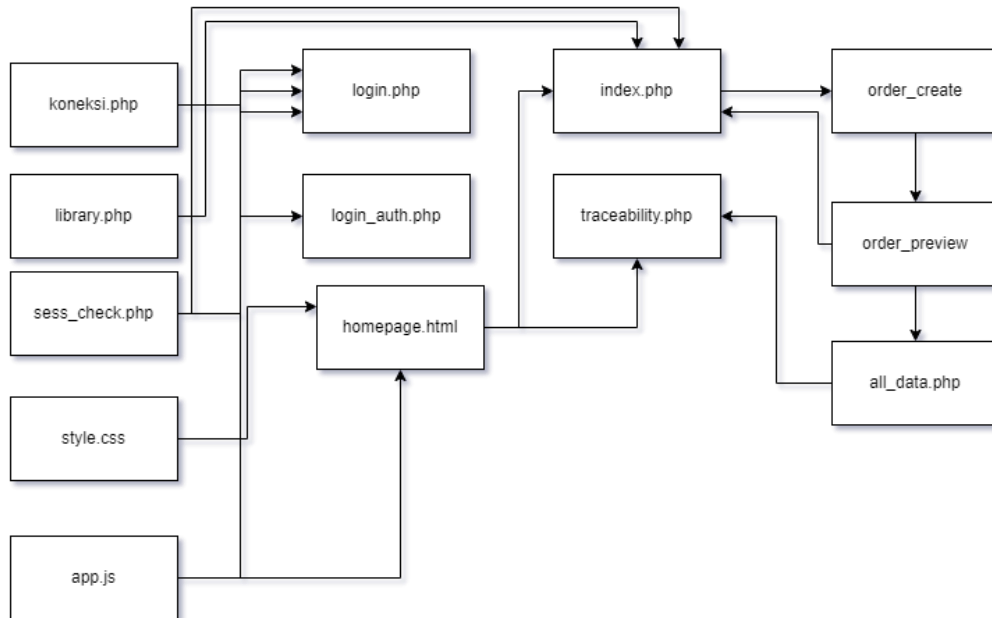
4. Desain Aplikasi

Desain aplikasi digunakan untuk memvisualisasikan gambaran awal dari aplikasi. Penulis membagi desain aplikasi menjadi dua yaitu *Low fidelity* dan *High fidelity*. *Low fidelity* berfokus pada elemen-elemen dasar dari sebuah antarmuka aplikasi, elemen tersebut berupa seperti *button*, *text*, navigasi, dan lain-lain. Sedangkan, *High fidelity* merupakan desain final aplikasi yang telah dirancang dengan memvisualisasikan dengan warna dan menampilkan gambar-gambar yang dibutuhkan seperti terlampir pada bagian desain aplikasi.

5. *Development* Aplikasi

Development aplikasi dirancang dengan membagi keseluruhan sistem pada aplikasi. *Development* aplikasi dilakukan dengan pembuatan struktur folder secara sistematis agar *project* yang dihasilkan bersifat *friendly code*. *Project* yang bersifat *friendly code* memudahkan *server* dalam mengirimkan permintaan kepada web

browser dan menampilkan *user interface* tanpa proses pengulangan perintah secara berkali-kali seperti pada Gambar 3.5



Gambar 3.5 Development Aplikasi

a. *Session Akun Role Permissions*

Developer membuat fungsi algoritma untuk memberikan *session* atau hak akses pada masing – masing akun. Pada penelitian ini, *developer* membagi menjadi 6 *session*, yaitu *session* petani, *session* kebun, *session* logistik, *session* pabrik, *session* super admin, dan *session* user. *Session* ini dibutuhkan untuk membagi *features* dan fungsionalitas/*objective* yang berbeda-beda pada tiap-tiap *stakeholder*.

b. *Authentication User*

Developer membuat *login auth* untuk *user* pada saat *user* ingin menuju portal *dashboard* masing – masing *session*. Hal ini mempermudah *user* dalam menggunakan aplikasi, karena seluruh *dashboard* yang dibangun tidak dijadikan satu dalam satu *session*. Namun, *dashboard* dibangun dengan membagi hak akses dari masing – masing *session*.

c. Membuat *Database*

Developer membuat *database* sesuai dengan master data yang dibutuhkan. Pada penelitian ini, master data dibagi menjadi 8 bagian yaitu master data admin, master data user, master data petani, master data kebun, master data area, master data logistik, master data pabrik, dan master data pesanan.

d. *Get Data Database*

Developer membuat fungsionalitas dalam proses pengembangan aplikasi Telsa Super Apps dengan memanggil atau men-*get* data dari *database* yang berfungsi untuk menampilkan data *backend* pada tampilan *frontend* aplikasi. Sehingga *user* dapat men-*tracking* produk CPO.

6. *Deployment* Aplikasi

Deployment aplikasi dilakukan dengan menggunakan dua *control panel* yaitu *localhost* dan *cloud internet hosting*. Aplikasi ini dilakukan pengujian dengan menggunakan *Xampp Control Panel Localhost* untuk menguji koneksi antar *database (backend)* dan *user interface (frontend)*. Kemudian, folder *project* aplikasi ini dilakukan perilisian melalui *Control Panel Hosting*. Perilisian aplikasi menghasilkan situs website berupa <https://telsa.isi-net.org/> dan dapat diakses oleh *user*.

7. *Testing*

Setelah rancangan dan *deployment* aplikasi didapatkan, *testing* dilakukan pada aplikasi dengan metode *blackbox testing*. Dengan menguji aplikasi dari segala

aspek, mulai dari *user interface*, *user experience*, fungsionalitas aplikasi, *get data database*, *authentication user* hingga fitur *traceability* produk CPO.

8. Analisis Capaian

Pengujian aplikasi yang sudah dilakukan dengan menggunakan metode *blackbox testing*, selanjutnya dilakukan analisis capaian melalui *assessment* proses pengembangan dan *assessment* kepuasan user.

9. Laporan Penelitian

Penulis menyajikan hasil penelitian dalam bentuk laporan akhir. Hasil penelitian ini dapat memberikan rekomendasi untuk pengimplementasian penggunaan aplikasi platform *traceability* pada manajemen rantai pasok perusahaan kelapa sawit dan dapat menambah perspektif baru mengenai subjek tema yang dibahas terhadap pembaca. Laporan ini digunakan sebagai bentuk tanggung jawab penulis terhadap tugas akhir yang telah dilakukan dan digunakan untuk melakukan seminar ujian komprehensi

BAB 4

KESIMPULAN

4.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Model manajemen rantai pasok perusahaan kelapa sawit dapat ditransformasikan dari metode tradisional/konvensional menjadi model *blockchain* melalui pembuatan *platform* aplikasi *traceability* menggunakan PHP Laravel. Aplikasi tersebut di *branding* dengan nama Telsa Super Apps.
2. Selain sebagai miniatur *Enterprise Resource Planning* (ERP), aplikasi Telsa Super Apps juga berfungsi sebagai media untuk menjamin ketelusuran (*traceability*) produk *Crude Palm Oil* (CPO) yang dihasilkan oleh perusahaan minyak kelapa sawit.
3. Dengan konsep *blockchain*, dimana seluruh data dasar dan data transaksi terekam/tercatat di aplikasi Telsa Super Apps dan dapat diakses oleh setiap stakeholder dalam jaringan *blockchain*, maka *visibility*, transparansi, dan akuntabilitas proses bisnis perusahaan dapat terjamin.
4. Data yang tersimpan dan informasi yang tersedia melalui aplikasi dapat dijadikan sebagai *decision support system* bagi manajemen perusahaan.
5. Ketelusuran produk CPO melalui aplikasi Telsa Super Apps dapat menjamin kepercayaan (*trust*) dan keberlanjutan (*sustainable*) produk sawit dan meminimalisasi dampak kampanye hitam terhadap produk sawit Indonesia.

4.2 Saran

Aplikasi yang dibangun melalui penelitian ini masih dapat terus dikembangkan dan ditingkatkan fungsi-fungsinya menjadi sebuah aplikasi ERP *dedicated*. Pengembangan platform ini dapat dilakukan dengan menambah *stakeholder* terkait dari hulu sampai ke hilir (konsumen) diluar 4 *stakeholder* utama yang telah diakomodir dalam versi saat ini. Selain itu juga dapat ditambahkan fungsi transaksi yang terkait dengan keuangan (pembelian/pembayaran) sehingga aplikasi dapat menjadi alternatif SAP sebagai platform dasar ERP dalam industri kelapa sawit.

DAFTAR PUSTAKA

- [1] Saefullah, 2020. "Rantai Pasok Sawit". <https://ditjenbun.pertanian.go.id/> , diakses pada 18 Januari 2022.
- [2] Sokoastri V., Setiadi D., Hakim A., Mawardhi A.D. 2019. Smallholders oil palm: = Problems and solutions. *Sodality Jurnal Sosiologi Pedesaan* 7 (3), 182-19 DOI:10.22500/sodality.v7i3.27221
- [3] Menteri Pertanian Republik Indonesia. 2020. Peraturan Menteri Pertanian Republik Indonesia Nomor 38 Tahun 2020 Tentang Penyelenggaraan Sertifikasi Perkebunan Kelapa Sawit Berkelanjutan Indonesia. Berita Negara Republik Indonesia. Jakarta.
- [4] N.K. Tsolakis, C.A. Keramydas, A.K. Toka, D.A. Aidonis, E.T. Iakovou, Agrifood supply chain manajemen: a comprehensive hierarchical decision-making framework and a critical taxonomy, *Biosyst. Eng.* 120 (2014) 47–64.
- [5] L.P. Tan, C.O. Fong, Determination of the crop mix of a rubber and oil palm plantation – a programming approach, *Eur. J. Oper. Res.* 34 (1998) 362–371.
- [6] R.G. García-Cáceres, A.L. Núñez, T.A. Ramírez, S.A. Jaimes, Characterization of the upstream phase of the Colombian oil palm agribusiness value and supply chain, *DYNA* 80 (179) (2013) 79–89.
- [7] Choi, T.-M., Wallace, S. W., & Wang, Y. (2017). Big data analytics in operations manajemen. *Production and Operations Manajemen*, 27(10), 1868-1883. <http://dx.doi.org/10.1111/poms.12838>
- [8] Kaur, H., & Singh, S. P. (2019). Heuristic 20odelling for sustainable procurement and logistics in a supply chain using big data. *Computers & Operations Research*, 98, 301-321. <http://dx.doi.org/10.1016/j.cor.2017.05.008>.
- [9] The Forest Trust, 2018. "Buku Panduan Pelaksanaan Traceability to Plantation (TiP) (kemamputelusuran produk hingga kebun)". <https://toolsfortransformation.net/indonesia/wp->

content/uploads/2017/05/Buku-Panduan-Pelaksanaan-TTP-Generik-T4T.pdf diakses pada 28 November 2022

- [10] Branch, M., Stentoft, J., Jensen, J. K., & Rajkumar, C. (2018). Android Practitioners understanding of big data and its applications in supply chain manajemen. *International Journal of Logistics Manajemen*, 29(2), 555-574. <http://dx.doi.org/10.1108/IJLM-05-2017-0115>.
- [11] El-Kassar, A.-N., & Singh, S. K. (2019). Green innovation and organizational performance: the influence of big data and the moderating role of manajemen commitment and HR practices. *Technological Forecasting and Social Change*, 144, 483-498 <http://dx.doi.org/10.1016/j.techfore.2017.12.016>.
- [12] Jelsma I., Slingerland M., Giller K.E., Bijman J. 2017. Collective action in a smallholder oil palm production system in Indonesia: The key to sustainable and inclusive smallholder palm oil. *J. of rural studies* 54: 198-210