

Pengembangan Sistem Pelaporan Pajak untuk Koperasi Sawit Berbasis Website

David John

Sistem Informasi, Fakultas Ilmu Komputer dan Desain, Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav.22, Jakarta 13210
Email: 2018103932@student.kalbis.ac.id

Abstrak: Penelitian ini bertujuan untuk membuat aplikasi sistem pelaporan pajak untuk koperasi sawit berbasis web. Aplikasi ini hanya dapat digunakan untuk koperasi sawit saja karena alur bisnis setiap jenis koperasi berbeda. Metode yang digunakan adalah Software Development Life Cycle (SDLC) dan software pengembangan yang digunakan adalah Visual Studio Code dengan Bahasa pemrograman JavaScript. Hasil yang didapatkan pada penelitian ini adalah aplikasi ini dapat memenuhi seluruh alur kerja dalam koperasi sawit, mempunyai jurnal transaksi yang lengkap dan dapat memudahkan atau mempercepat pengurus koperasi dalam pelaporan pajak

Keyword: Web, SDLC, JavaScript

Abstract: This study aims to create a web-based tax reporting system application for palm oil cooperatives. This application can only be used for oil palm cooperatives because the business flow of each type of cooperative is different. The method used is Software Development Life Cycle (SDLC) and the development software used is Visual Studio Code with JavaScript programming language. The results obtained in this study are that this application can fulfill all workflows in palm oil cooperatives, has a complete transaction journal and can facilitate or speed up cooperative management in tax reporting

Keyword: Web, SDLC, JavaScript

I. PENDAHULUAN

Koperasi merupakan suatu badan yang dibentuk atas asas kekeluargaan yang bertujuan untuk mensejahterakan para anggotanya. Koperasi diwajibkan untuk mencatat hasil transaksi ataupun stok barang ke dalam buku koperasi yang nantinya akan digunakan sebagai acuan dalam pembuatan laporan keuangan. Berdasarkan laporan Kementerian Koperasi dan UMKM (Kemenkop UKM) jumlah UMKM yang terhubung dengan platform digital sebanyak 16,4 juta pelaku UMKM [1] atau sekitar 25% dari total UMKM, 75% lainnya masih menggunakan cara manual seperti mencatat transaksi pembelian dan masuknya barang dari anggota. Dari wawancara penelitian dengan narasumber dari Koperasi Sawit, bahwa umumnya transaksi barang, penitipan barang dari anggota untuk dijual, ataupun keluar masuknya uang kas koperasi

dicatat dalam buku catatan koperasi dan dengan memberitahu anggota atau koperasi yang ada di tempat. Mengenai sistem kasir dan stok yang digunakan Koperasi sawit saat ini masih manual, yaitu pengurus koperasi mencatat transaksi pembelian ke dalam buku koperasi serta mengecek apakah stok persediaan barang sesuai dengan catatan koperasi, setelah itu memberitahukan kepada rekan pengurus lainnya yang berada di tempat. Namun dengan hal tersebut maka memerlukan proses dan waktu yang lama.

Permasalahan yang masih terjadi di Koperasi Sawit yaitu, meningkatnya jumlah transaksi dan anggota dan tidak semua pengurus atau anggotanya mempunyai hubungan yang baik sehingga banyak terjadinya perselisihan dan ketidakpercayaan antar anggotanya. Jika setiap anggota tidak saling percaya satu sama lain maka hasil, dan fungsi laporan keuangan yang disampaikan

setiap tahunnya kepada anggota ataupun saat melaporkan ke Kantor Pelayanan Pajak (KPP) dianggap tidak akurat oleh anggota dan pengurus koperasi tersebut dan dapat menyebabkan koperasi tidak menjalankan fungsinya

Salah satu solusi yang dapat digunakan untuk Koperasi Sawit adalah mengembangkan aplikasi secara daring yang dapat diakses oleh anggota ataupun pengurus, kapan saja, dan dimana saja yaitu *Point of Sales* (POS) dan fitur tambahan untuk memudahkan pengurus koperasi dalam membuat laporan keuangan tersebut. POS adalah suatu sistem yang digunakan dalam kebutuhan berbagai macam usaha bisnis ritel untuk dapat mempermudah proses transaksi jual beli secara cepat, aman dan sistematis. Keunggulan dari POS yaitu, memudahkan proses bisnis antara penjual dan pembeli, meningkatkan keamanan setiap transaksi, dan Proses transaksi menjadi lebih terstruktur.

Dari penelitian dahulu yang dilakukan oleh Sapto Catur Cahyodi, dkk 2017 berjudul "Sistem Informasi Point of Sales Berbasis Web Pada Colony Amaranta Bekasi" [2] bahwa penelitian tersebut membuat Sistem Informasi Point of Sales dibuat untuk meningkatkan kinerja dan pelayanan dalam melakukan proses registrasi customer, proses input data dan proses penjualan, sehingga mempermudah dalam pembuatan laporan-laporan yang dibutuhkan dan sekarang telah berbasis intranet sehingga mempermudah pemilik dalam mengontrol jalannya proses bisnis yang berjalan. Penelitian lainnya yang dilakukan oleh Silvester Dian Handy Permana, dkk 2015 berjudul "Analisa dan Perancangan Aplikasi Point Of Sale (POS) Untuk Mendukung Manajemen Hubungan Pelanggan"[3] bahwa Rancangan aplikasi *point of sales* (POS) ini mampu mengatasi permasalahan dan dapat menyajikan informasi secara lebih baik dan terkomputerisasi. Dengan diterapkannya aplikasi POS ini dapat

membantu tugas-tugas pihak-pihak terkait atau seluruh stakeholder yang berhubungan langsung dengan aplikasi POS ini

Penelitian ini bertujuan untuk menghasilkan aplikasi yang dapat memudahkan Koperasi Sawit dalam proses laporan keuangan dengan menyimpan, mencari dan menjalankan alur kerja koperasi. Aplikasi ini berbasis web, karena berdasarkan data statistik dari *InternetWorldStats* menyebutkan bahwa pengguna Internet di Indonesia memiliki persentase sebesar 76,8% dari total penduduk di Indonesia memiliki akses Internet, sehingga akan memudahkan bagi seluruh anggota atau pengurus dari Koperasi Sawit untuk mengakses aplikasi ini. Aplikasi Ini diharapkan dapat meningkatkan kesehatan koperasi dan memotivasi koperasi lainnya untuk menggunakan sistem informasi untuk memudahkan dan melancarkan dalam proses pembuatan laporan keuangan.

A. Rumusan Masalah

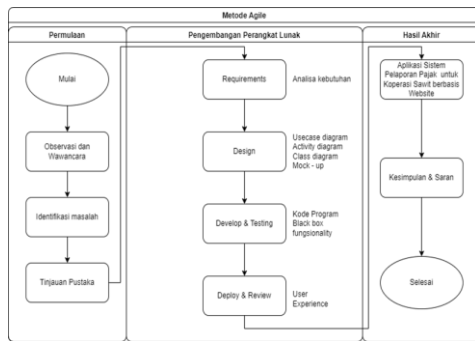
Berdasarkan latar belakang, rumusan masalah dari penelitian ini adalah bagaimana membangun aplikasi koperasi sawit yang dapat memudahkan proses pencatatan transaksi, dan inventaris

B. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk dapat membangun suatu aplikasi yang dapat memudahkan proses pengurus koperasi dalam pembuatan laporan keuangan

II. METODE PENELITIAN

A. Tahapan Penelitian



Gambar 1 Tahap Penelitian

Pada Gambar 1 adalah tahapan penelitian yang dilakukan sebagai gambaran dari proses pembuatan aplikasi Sistem Pelaporan Pajak untuk Koperasi Sawit berbasis Website. Langkah pertama yang dilakukan peneliti adalah Observasi dan Wawancara terhadap Pengurus Koperasi Sawit. Berdasarkan informasi yang diberikan oleh pengurus koperasi mengenai latar belakang permasalahan yang terjadi di dalam koperasi tersebut, peneliti dapat menarik kesimpulan atau mengidentifikasi masalah dan menyediakan solusi untuk permasalahan tersebut. Tahap selanjutnya menentukan metodologi yang digunakan pada penelitian ini serta untuk mendukung teori - teori dengan menggunakan beberapa referensi dari buku - buku, jurnal, website dan lainnya untuk menyelesaikan masalah yang ada pada penelitian ini.

Pada tahap pengembangan perangkat lunak penelitian ini menggunakan metode Agile dengan pemodelan UML (*Unified Modelling Language*). Metodologi Agile memiliki empat buah tahapan yaitu *Requirement*, *Design*, *Develop and Testing*, *Deploy and Review*. Dalam empat tahapan tersebut setiap fasenya terdapat alur sebagai panduan peneliti selama proses pengembangan perangkat lunak.

Tahap pertama dalam metodologi Agile yaitu tahap *Requirements*, untuk mendefinisikan dan mengumpulkan kebutuhan dalam pengembangan aplikasi berdasarkan hasil dari wawancara sebelum pengembangan aplikasi dilakukan. Tahap kedua adalah *Design*, tahap ini diperuntukan sebagai acuan desain dari aplikasi yang akan dikembangkan dan desain arsitektur sistem yang akan digunakan dalam pengembangan akan menggunakan *Class Diagram*. Tahap Ketiga adalah *Develop and Testing*, di tahap ini peneliti melakukan pemrograman berdasarkan kebutuhan sistem dari tahap sebelumnya dengan bahasa pemrograman yang dipilih oleh peneliti untuk memudahkan pengembangan aplikasi dari penelitian ini, setelah peneliti merasa aplikasi ini sudah memenuhi semua kriteria yang ditentukan di tahap - tahap sebelumnya. Peneliti melakukan *testing* dengan data *dummy* untuk mencoba fungsionalitas dan *User Experience* dalam menggunakan aplikasi ini, serta melakukan percobaan apakah sistem berjalan sesuai dengan kriteria dan logika yang sudah diberikan. Tahap keempat merupakan tahap terakhir yaitu *Deploy dan Review*, dimana pada tahap ini peneliti melakukan pengujian dengan sejumlah partisipan menggunakan metode *black box*, setelah itu partisipan mengisi sebuah kuesioner untuk menentukan apakah aplikasi sudah berjalan sesuai ekspektasi. Setelah selesai semua, maka aplikasi akan dapat digunakan melalui *website*.

B. Framework

Framework adalah library yang menawarkan pendapat tentang bagaimana perangkat lunak dibangun. Pendapat ini memungkinkan prediktabilitas dan homogenitas dalam aplikasi. Prediktabilitas memungkinkan perangkat lunak untuk berkembang ke skala ukuran yang sangat besar dan masih

dapat dirawat. Prediktabilitas dan perawatan merupakan hal yang penting untuk kelangsungan perangkat lunak tersebut [4].

Setiap JavaScript framework menyediakan cara membuat UI (User Interface) lebih deklaratif. Framework memungkinkan anda untuk menulis kode UI sesuai yang anda inginkan, dan framework mewujudkannya di DOM (Document Object Model) di belakang layar. Cara vanilla JavaScript untuk membuat DOM elements dalam repetisi sangat susah untuk dimengerti dalam sekilas.

1. Express.js

Express adalah Node web *framework* paling populer, dan merupakan bagian dasar web *framework* populer lainnya. Meskipun Express cukup minimalis, *developer* telah membuat paket *middleware* yang compatible untuk mengatasi hampir semua masalah pengembangan web [5]. Web *framework* sering menganggap diri mereka sebagai “*unopinionated*” atau “*unopinionated*”.

2. Vue

Vue adalah *framework* progresif *open-source* yang dirancang untuk dapat diintegrasikan secara bertahap, karena *library* ini difokuskan untuk *layer* tampilan saja. Meskipun demikian, *library* ini lebih dari cukup untuk membuat *Single Page Application*(SPA) dengan bantuan *tools*, dan mudah untuk diintegrasikan dengan *library* lain atau proyek yang sudah ada. Vue sering digunakan untuk membangun UI web dan SPA. selain itu Vue juga dapat digunakan dalam pembuatan aplikasi *Desktop* maupun *Mobile*, dengan bantuan dari ekstensi HTML dan JavaScript yang bekerjasama dengan Electron *framework* menjadikannya *Frontend tool* yang sangat diminati. Vue menggunakan *Model View Controller*

tradisional untuk menampilkan UI dari sebuah aplikasi atau web dengan menggunakan inti *library*-nya sebagai *default layer* [6]. Namun selain itu Vue juga dapat beroperasi menggunakan *Component Based Architecture* (CBA) seperti yang digunakan oleh React.

C. Database

Database adalah kumpulan terorganisir dari informasi terstruktur, atau data, biasanya disimpan secara elektronik dalam sistem komputer. Sebuah database biasanya dikendalikan oleh *database management system* (DBMS). Bersama-sama, data dan DBMS serta dengan aplikasi yang terkait dengannya, disebut sebagai *database system*, sering disingkat menjadi *database*. Data dalam tipe *database* yang paling umum beroperasi saat ini dimodelkan dalam baris dan kolom serangkaian tabel untuk membuat pemrosesan *query* data menjadi efisien. Data kemudian dapat dengan mudah diakses, dikelola, dimodifikasi, diperbarui, dikendalikan dan diatur. Sebagian besar *database* menggunakan bahasa *structured query language* (SQL) untuk menulis dan meminta data.

1. MongoDB

MongoDB adalah program *database management system* NoSQL *open-sourced*. NoSQL digunakan sebagai alternatif untuk *database relational* tradisional. *Database* NoSQL cukup berguna untuk bekerja dengan kumpulan besar data terdistribusi. MongoDB adalah alay yang dapat mengelola informasi berorientasi dokumen, menyimpan atau mengambil informasi [7].

MongoDB memanfaatkan catatan yang terdiri dari dokumen yang berisi struktur data yang terdiri dari gabungan kolom dan nilai. Dokumen adalah unit dasar data di MongoDB. Bentuk

dokumen-nya mirip dengan JavaScript Object Notation, tetapi menggunakan varian yang disebut *Binary JSON* (BSON). Manfaat menggunakan BSON adalah akomodasi tipe data yang lebih banyak. Bidang dalam dokumen ini mirip dengan kolom dalam *relational database*. Nilai yang terkandung dapat berupa berbagai tipe data, termasuk dokumen lain, *array*, menurut *user manual* dari MongoDB. Dokumen juga akan memasukan kunci utama sebagai pengidentifikasi unik.

D. GraphQL

GraphQL adalah *query language* dan *server-side runtime* untuk *Application Programming Interface* (API) yang memberikan prioritas untuk memberikan sesuai dengan *request* mereka. GraphQL didesain untuk membuat API dengan cepat, fleksibel, dan *developer-friendly* [8]. GraphQL bahkan dapat digunakan di dalam *Integrated Development Environment* (IDE) yang disebut dengan *GraphiQL*. Sebagai alternatif dari *Representative State Transfer* (REST), GraphQL memungkinkan *developer* membuat *requests* yang menarik data dari banyak sumber data dalam satu *API call*.

E. Dataloader

Dataloader adalah sebuah *tool* yang meningkatkan performa *query* GraphQL. Dataloader mendukung kemampuan fungsional seperti *batching* dan *caching*. Saat membuat Dataloader, integrasi *server* menghasilkan *service loader* dan tipe dokumen sebagai kunci. Setelah itu *service loader* memuat data untuk daftar kunci dan mengembalikan *list value*-nya [9].

F. Unified Modeling Language (UML)

Unified Modeling Language merupakan bahasa pemodelan standar

yang terdiri dari serangkaian diagram terintegrasi, yang dikembangkan untuk membantu *developer* sistem dan *software* untuk menentukan, memvisualisasikan, membangun dan mendokumentasikan arsitektur *software*, dan pemodelan bisnis. UML merepresentasikan kumpulan praktik rekayasa terbaik yang telah terbukti berhasil dalam pemodelan sistem yang besar dan kompleks. UML adalah bagian yang sangat penting dalam pengembangan *object oriented software* dan proses *development* dari *software*. UML biasanya digunakan untuk notasi grafik untuk memberi gambaran dari segi desain *software* tersebut. UML dapat membantu tim dalam berkomunikasi, mencoba potensi desain lainnya, dan memvalidasi desain arsitektur *software* tersebut [10]. Pada penelitian ini terdapat 3 jenis UML yang digunakan yaitu, *Use Case Diagram*, *Class Diagram*, dan *Activity Diagram*.

G. Black Box Testing

Black Box Testing adalah metode pengujian *software* di mana fungsionalitas aplikasi *software* diuji tanpa memiliki pengetahuan tentang struktur kode internal, detail implementasi, dan jalur internal. Pengujian *Black Box* terutama berfokus pada input dan output aplikasi *software* dan sepenuhnya didasarkan pada persyaratan dan spesifikasi *software*. Ini juga dikenal sebagai Pengujian Perilaku.



Gambar 2 Black box testing

Black Box di atas dapat berupa *software* atau apa pun yang ingin Anda uji. Contohnya, sistem operasi seperti Windows, situs web seperti Google, *database* seperti Oracle atau bahkan aplikasi anda sendiri. Setelah *Black Box*,

Anda dapat menguji aplikasi ini hanya dengan berfokus pada input dan output tanpa mengetahui implementasi kode internalnya [11].

III. HASIL DAN PEMBAHASAN

A. Requirements

Untuk mengumpulkan data peneliti melakukan wawancara terhadap pengurus dari Koperasi Sawit, Wawancara dilakukan secara virtual dikarenakan kondisi pandemi yang sedang berlangsung. Setelah dilakukan wawancara dengan pengurus Koperasi, peneliti mendapatkan kebutuhan fungsional dan non fungsional untuk aplikasi ini, serta pengurus memberikan data-data yang diperlukan untuk mendukung penelitian ini. Data yang diberikan merupakan Laporan neraca dari Koperasi Sawit, Laporan Laba Rugi dan laporan atau dokumen lainnya yang mendukung dalam proses penelitian ini

1. Non Fungsional

Pada Tabel 3.1 kebutuhan non fungsional yang digunakan peneliti saat melakukan penelitian ini:

Tabel 1 Requirement Non Fungsional

No.	Tahap Perancangan	Tahap Implementasi
1	Manusia: Penulis melakukan wawancara, analisa dan juga mengambarkan aplikasi penelitian	Manusia: Wawancara dengan Pengurus Koperasi Sawit
2	Software: Visual Studio Code Express GraphQL Dataloader MongoDB Atlas Vue Vuex Vue-Router	Software; Express GraphQL Dataloader MongoDB Atlas Vue Vuex Vue-Router
3	Hardware: Hewlett-Packard Laptop	Hardware: Cloud Server

2. Fungsional

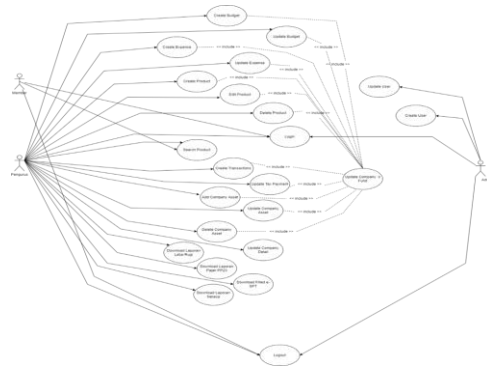
Berikut merupakan kebutuhan dari pengurus serta fitur lainnya yang mendukung kelancaran koperasi ini

1. Admin dapat masuk ke dalam sistem
2. Admin dapat membuat akun Pengurus
3. Admin dapat memperbarui detail akun Pengurus
4. Admin dapat membuat akun Member
5. Admin dapat memperbarui detail akun Member
6. Pengurus dapat masuk ke dalam sistem
7. Pengurus dapat menambahkan modal koperasi
8. Pengurus dapat menambahkan aset - aset koperasi serta mengubah aset - aset koperasi
9. Pengurus dapat membuat, menambahkan, mengubah, serta menghapus produk di koperasi
10. Pengurus dapat mencari produk yang ada di koperasi
11. Pengurus dapat melakukan penjualan melalui sistem
12. Pengurus dapat melihat transaksi penjualan terdahulu melalui sistem
13. Pengurus dapat membuat, serta merubah data pengeluaran koperasi melalui sistem
14. Pengurus dapat melihat total pajak pendapatan yang harus dibayarkan bulan ini melalui sistem
15. Pengurus dapat melihat Laporan Laba Rugi
16. Pengurus dapat *mendownload* Laporan Laba Rugi
17. Pengurus dapat melihat Laporan Pembayaran Pajak yang sudah maupun yang belum dibayarkan
18. Pengurus dapat *mendownload* Laporan Pembayaran Pajak
19. Pengurus dapat melihat Laporan Neraca
20. Pengurus dapat *mendownload* Laporan Neraca

21. Pengurus dapat *mendownload* Laporan SPT 1771 berdasarkan data - data yang sudah dimasukan ke dalam sistem
22. Pengurus dapat *mendownload* Laporan Khusus SPT 1771 berdasarkan data yang dimasukan ke dalam sistem
23. Pengurus dapat melihat perubahan apa saja yang dilakukan pengurus lainnya
24. Member dapat masuk ke dalam sistem
25. Member dapat melihat *dashboard* yang berisi transaksi hari ini, produk sedang dan hampir habis, serta perubahan apa saja yang dilakukan oleh pengurus hari ini
26. Member dapat melihat halaman Transaksi apa saja yang telah terjadi
27. Member dapat melihat riwayat perubahan yang terjadi pada sistem

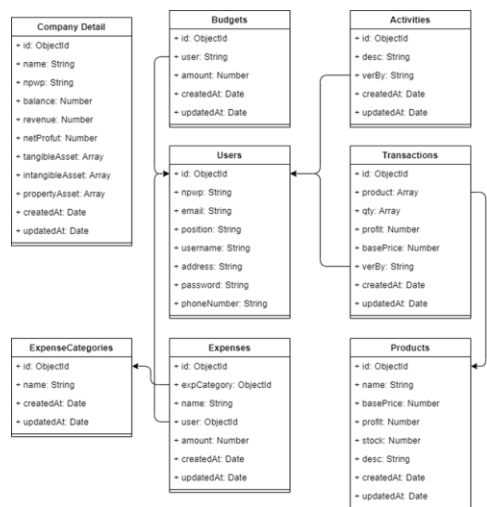
B. Desain

Pada tahap *design* yang dilakukan adalah membuat kerangka desain yang akan menghasilkan tampilan ataupun arsitektur dari aplikasi yang akan dibuat. Untuk menentukan ruang lingkup pada penelitian dan perencanaan yang dijelaskan dengan menggunakan *use case diagram*, *activity diagram*, dan *class diagram*. Pada gambar 3.3 adalah *use case diagram* pada aplikasi Sistem Pelaporan Pajak untuk Koperasi Sawit. Dalam aplikasi tersebut terdapat dua aktor yaitu Admin dan Pengurus. Masing - masing aktor terdapat beberapa *use case* atau fungsi yang diperlukan, sebagai berikut:



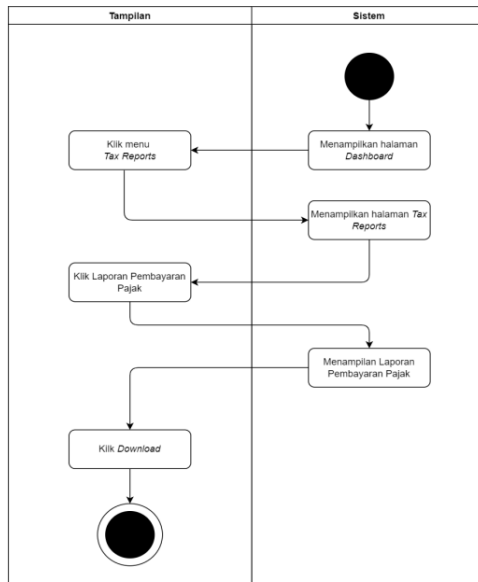
Gambar 3 Use Case Diagram

Berikut merupakan class diagram dari Sistem Pelaporan Pajak untuk Koperasi Sawit. *Database* dalam sistem ini menggunakan MongoDB Atlas yang bersifat *Cloud Data Platform*, bertipe *NoSql*. *Database* ini memiliki beberapa *collection* yang tidak secara langsung terhubung antara satu sama lain melainkan menyimpan *id* atau referensi dari *id* dokumen lainnya yang nantinya akan dijadikan satu dengan menggunakan *dataloader* untuk menangani permasalahan $n + 1$, serta *Resolver* dari *server* sistem ini. Hal ini akan dibahas lebih lanjut di bab selanjutnya. Pada Gambar 3 menunjukkan hubungan yang terdapat pada *database* yang digunakan untuk penelitian ini.



Gambar 4 Class Diagram

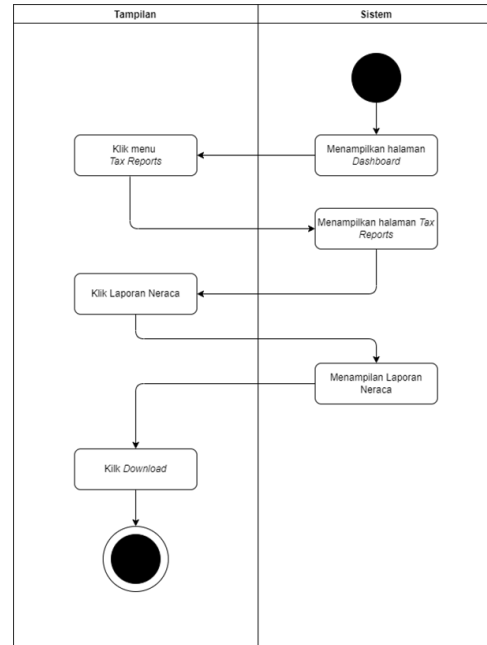
Pada Gambar 4 menjelaskan diagram aktivitas tentang proses *Download Laporan Laba Rugi* yang dilakukan Pengurus. Dimulai dari Pengurus masuk ke halaman *Dashboard* setelah *Login*, setelah itu Pengurus dapat menekan tombol *Tax Reports* pada *side navigation* menu. Sistem akan menampilkan halaman *Tax Reports*, setelah itu Pengurus dapat menekan tombol Laporan Laba Rugi. Selanjutnya Pengurus dapat menekan tombol *Download* yang ada pada halaman. Laporan Laba Rugi dihasilkan dari perhitungan dan data yang ada pada *database* sistem.



Gambar 5 Activity Diagram Pengurus: Download Laporan Pembayaran Pajak PP23

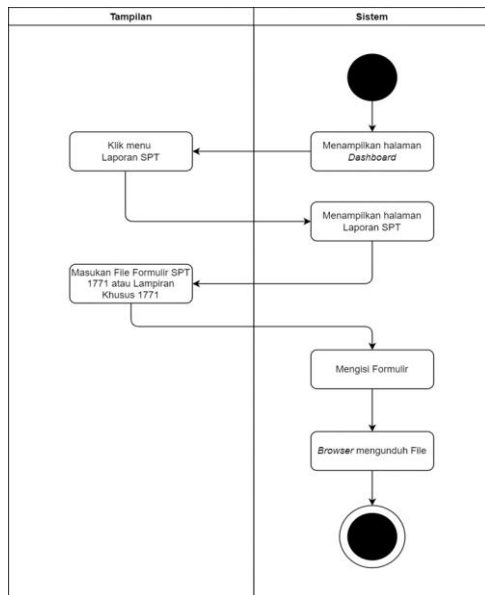
Pada Gambar 5 menjelaskan diagram aktivitas tentang proses *Download Laporan Pembayaran Pajak PP23* yang dilakukan Pengurus. Dimulai dari Pengurus masuk ke halaman *Dashboard* setelah *Login*, setelah itu Pengurus dapat menekan tombol *Tax Reports* pada *side navigation* menu. Sistem akan menampilkan halaman *Tax Reports*, setelah itu Pengurus dapat menekan tombol Laporan Pembayaran Pajak PP23. Selanjutnya Pengurus dapat menekan tombol *Download* yang ada

pada halaman. Laporan Pembayaran Pajak dihasilkan dari data yang ada pada *database* sistem.



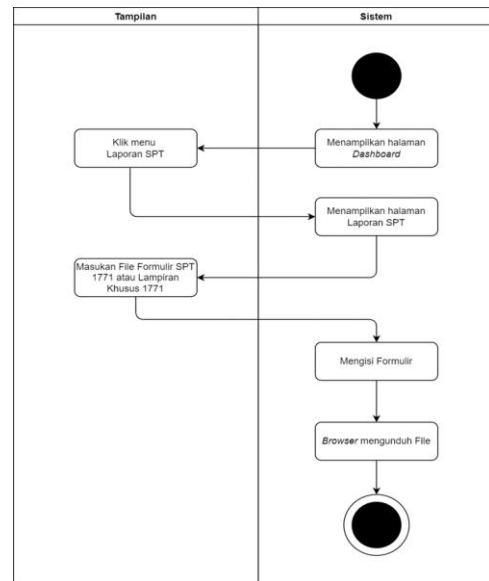
Gambar 6 Activity Diagram Pengurus: Download Laporan Neraca

Pada Gambar 6 menjelaskan diagram aktivitas tentang proses *Download Laporan Neraca* yang dilakukan Pengurus. Dimulai dari Pengurus masuk ke halaman *Dashboard* setelah *Login*, setelah itu Pengurus dapat menekan tombol *Tax Reports* pada *side navigation* menu. Sistem akan menampilkan halaman *Tax Reports*, setelah itu Pengurus dapat menekan tombol Laporan Neraca. Selanjutnya Pengurus dapat menekan tombol *Download* yang ada pada halaman. Laporan Neraca dihasilkan dari perhitungan dan data yang ada pada *database* sistem



Gambar 7 Activity Diagram Pengurus: Download Formulir SPT

Pada Gambar 7 menjelaskan diagram aktivitas tentang proses *Download Formulir SPT* yang dilakukan Pengurus. Dimulai dari Pengurus masuk ke halaman *Dashboard* setelah *Login*, setelah itu Pengurus dapat menekan tombol *Laporan SPT* pada *side navigation* menu. Sistem akan menampilkan halaman *Laporan SPT*, setelah itu Pengurus dapat meng-*upload file* formulir SPT 1771 pada kolom Formulir 1771. Selanjutnya *browser* Pengurus akan secara otomatis men-*download* Formulir SPT yang sudah diisi oleh sistem. Formulir SPT 1771 dihasilkan dari perhitungan dan data yang ada pada *database* sistem.



Gambar 8 Activity Diagram Pengurus: Download Lampiran Khusus 1771

Pada Gambar 8 menjelaskan diagram aktivitas tentang proses *Download Lampiran Khusus 1771* yang dilakukan Pengurus. Dimulai dari Pengurus masuk ke halaman *Dashboard* setelah *Login*, setelah itu Pengurus dapat menekan tombol *Laporan SPT* pada *side navigation* menu. Sistem akan menampilkan halaman *Laporan SPT*, setelah itu Pengurus dapat meng-*upload file* formulir Lampiran Khusus 1771 pada kolom Lampiran Khusus 1771. Selanjutnya *browser* Pengurus akan secara otomatis men-*download* Lampiran Khusus 1771 yang sudah diisi oleh sistem. Lampiran Khusus 1771 dihasilkan dari perhitungan dan data yang ada pada *database* sistem.

C. Develop and Testing

Pada tahap ini semua desain struktur dan tampilan yang telah dibuat akan diimplementasikan ke dalam bentuk kode program untuk membangun aplikasi Sistem Pelaporan Pajak untuk Koperasi Sawit berbasis Website. Penggunaan *website* ini menggunakan *software* Visual Studio Code untuk *text editor*, MongoDB Atlas sebagai *database* yang berorientasi NoSQL, GraphQL

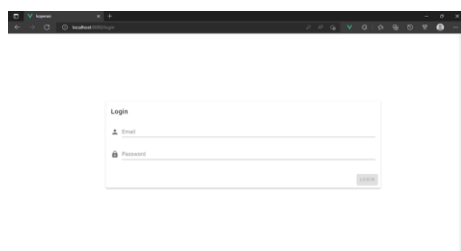
sebagai bahasa *Query* antara *front-end* dan *back-end*, *Vue* sebagai *framework front-end*, *Express Js* sebagai *framework backend* serta *Dataloader* untuk mengatasi permasalahan *N+1*. Berikut ini adalah penerapan kode yang dibuat untuk pembangunan aplikasi Sistem Pelaporan Pajak untuk Koperasi Sawit berbasis Website.

D. Deployment and Review

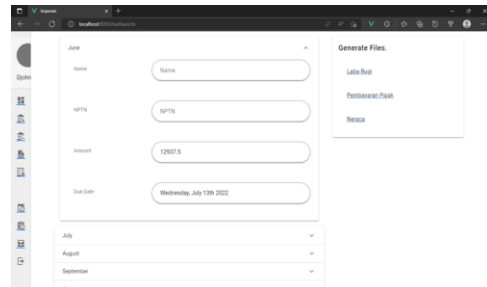
Pada tahap ini peneliti melakukan pengujian terhadap sistem dengan menggunakan metode *black box* dan *user experience*. Pengujian ini ditujukan untuk menguji fungsionalitas pada sistem *website*, apakah fungsi berjalan sesuai dengan desain yang ditentukan pada awal tahap desain atau belum. Peneliti akan membuat skenario pengujian dan membuat indikator kesuksesan untuk mengetahui apakah fungsi berjalan sesuai keinginan atau belum. Berikut adalah tabel pengujian *black box*.

E. Hasil Tampilan Aplikasi

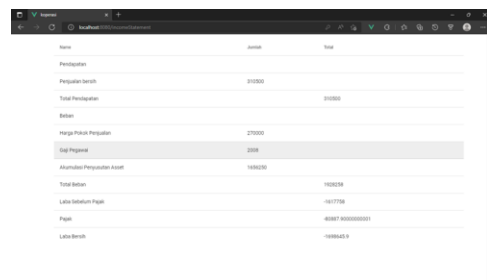
Pada bab ini peneliti menampilkan tampilan aplikasi yang sudah dibuat dengan menjelaskan beberapa fitur utama pada aplikasi dan menjelaskan kegunaan fitur tersebut. Tampilan aplikasi ini didesain sesuai dengan kebutuhan pengguna yang telah ditentukan pada tahap sebelumnya. Tampilan aplikasi Sistem Pelaporan Pajak untuk Koperasi Sawit berbasis Website terdiri dari beberapa tampilan yaitu tampilan Admin, Pengurus dan Member. Tampilan aplikasi akan dijelaskan sebagai berikut



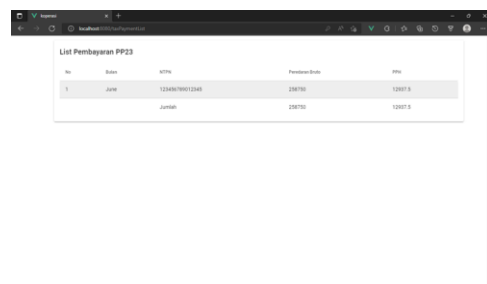
Gambar 9 Hasil Tampilan Halaman Login



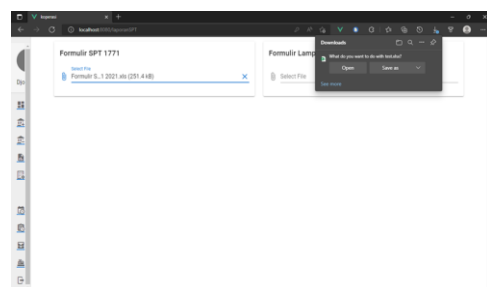
Gambar 10 Hasil Tampilan Halaman Tax Reports



Gambar 11 Hasil Tampilan Halaman Tax Reports: Laba Rugi



Gambar 12 Hasil Tampilan Halaman Tax Reports: Laporan Pembayaran Pajak



Gambar 13 Hasil Tampilan Halaman Laporan SPT

F. Hasil Black Box Testing

Pada tahap pengujian ini, peneliti akan melakukan pengujian fungsionalitas maupun non fungsionalitas pada sistem website. Apakah aplikasi ini sudah berfungsi sesuai dengan kebutuhan atau tidak, dengan fitur yang sudah ditentukan

pada bab sebelumnya. Pada penelitian ini peneliti sebelumnya sudah membuat skenario pengujian dan indikator kesuksesan dari fitur tersebut. Berikut ini adalah hasil pengujian *black box* yang dapat dilihat pada Tabel dibawah

12	Company Details : Create Tangible Asset	Mengisi kolom amount lebih banyak dari balance	Sistem membatasi amount dengan balance	Berhasil
		Menekan tombol save dengan kolom salah satunya kosong	Sistem tidak memperbolehkan user menekan tombol tersebut	Berhasil
		Mengosongkan salah satu kolom	Sistem menampilkan kolom tersebut "is required"	Berhasil
		Menekan tombol save	Sistem menampilkan data terbaru	Berhasil
14	Company Details : Update Tangible Asset	Mengisi kolom amount lebih banyak dari balance	Sistem membatasi amount dengan balance	Berhasil
		Menekan tombol save dengan kolom salah satunya kosong	Sistem tidak memperbolehkan user menekan tombol tersebut	Berhasil
		Mengosongkan salah satu kolom	Sistem menampilkan kolom tersebut "is required"	Berhasil
		Menekan tombol save	Sistem menampilkan data terbaru	Berhasil
14	Company Details : Delete Tangible Asset	Menekan tombol delete	Sistem menghapus aset tersebut	Berhasil
15	Company Details : Create Intangible Asset	Mengisi kolom amount lebih banyak dari balance	Sistem membatasi amount dengan balance	Berhasil
		Menekan tombol save dengan kolom salah satunya kosong	Sistem tidak memperbolehkan user menekan tombol tersebut	Berhasil
		Mengosongkan salah satu kolom	Sistem menampilkan kolom tersebut "is required"	Berhasil
		Menekan tombol save	Sistem menampilkan data terbaru	Berhasil
16	Company Details : Update Intangible Asset	Mengisi kolom amount lebih banyak dari balance	Sistem membatasi amount dengan balance	Berhasil
		Menekan tombol save dengan kolom salah satunya kosong	Sistem tidak memperbolehkan user menekan tombol tersebut	Berhasil
		Mengosongkan salah satu kolom	Sistem menampilkan kolom tersebut "is required"	Berhasil
		Menekan tombol save	Sistem menampilkan data terbaru	Berhasil
17	Company Details : Delete Intangible Asset	Menekan tombol delete	Sistem menghapus aset tersebut	Berhasil

Intangible Asset		salah satu kolom	tersebut "is required"	
18	Company Details : Create Property Asset	Mengisi kolom amount lebih banyak dari balance	Sistem membatasi amount dengan balance	Berhasil
		Menekan tombol save dengan kolom salah satunya kosong	Sistem tidak memperbolehkan user menekan tombol tersebut	Berhasil
		Mengosongkan salah satu kolom	Sistem menampilkan kolom tersebut "is required"	Berhasil
		Menekan tombol save	Sistem menampilkan data terbaru	Berhasil
19	Company Details : Update Property Asset	Mengisi kolom amount lebih banyak dari balance	Sistem membatasi amount dengan balance	Berhasil
		Menekan tombol save dengan kolom salah satunya kosong	Sistem tidak memperbolehkan user menekan tombol tersebut	Berhasil
		Mengosongkan salah satu kolom	Sistem menampilkan kolom tersebut "is required"	Berhasil
		Menekan tombol save	Sistem menampilkan data terbaru	Berhasil
20	Company Details : Delete Property Asset	Menekan tombol delete	Sistem menghapus aset tersebut	Berhasil
21	Tax Reports : Update Tax Report	Mengosongkan	Sistem menampilkan kolom	Berhasil
		Menekan tombol save	Sistem menampilkan data terbaru	Berhasil
22	Company Details : Update Company Detail	Mengisi kolom npwp kurang dari 15 karakter	Sistem menampilkan "Npwp must be 15 characters"	Berhasil
		Mengosongkan salah satu kolom	Sistem menampilkan kolom tersebut "is required"	Berhasil
		Menekan tombol save dengan kolom salah satunya kosong	Sistem tidak memperbolehkan user menekan tombol tersebut	Berhasil
		Menekan tombol save	Sistem menampilkan data terbaru	Berhasil
23	Laporan SPT:Download SPT1771	Memilih file yang salah	Sistem tidak menampilkan file tersebut	Berhasil
		Memasukkan file spt 1771	Sistem mengisi file tersebut	Berhasil
24	Laporan SPT:Download Lampiran Khusus 1771	Memilih file yang salah	Sistem tidak menampilkan file tersebut	Berhasil
		Memasukkan file spt 1771	Sistem mengisi file tersebut	Berhasil

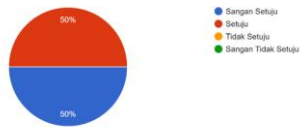
G. Hasil User Experience

Pada tahap ini peneliti akan menampilkan hasil dari pengujian aplikasi yang sudah dilakukan sebelumnya. Data tersebut dilakukan

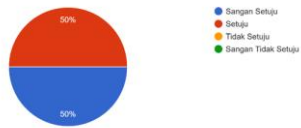
untuk mengetahui pengalaman Member, Pengurus dan Admin dalam menggunakan aplikasi Sistem Pelaporan Pajak untuk Koperasi Sawit berbasis Website. Hal ini juga dilakukan untuk mengetahui kekurangan website berdasarkan kebutuhan Member, Pengurus, dan Admin. Untuk hasil pengujian dapat dilihat pada gambar berikut:

Kuesioner Pengurus

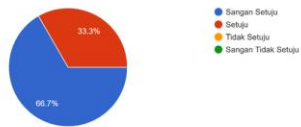
Apakah aplikasi mempermudah pengurus untuk mengelola koperasi ?
6 responses



Apakah aplikasi mempermudah pengurus untuk memeriksa laporan pembayaran pajak ?
6 responses



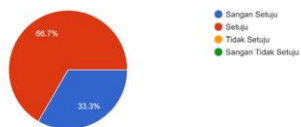
Apakah aplikasi mempermudah pengurus untuk memeriksa laporan laba rugi ?
6 responses



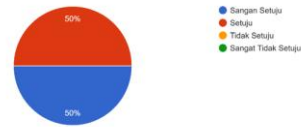
Apakah aplikasi mempermudah pengurus untuk membuat laporan SPT ?
6 responses



Apakah aplikasi mempermudah pengurus untuk membuat Lampiran Khusus ?
6 responses



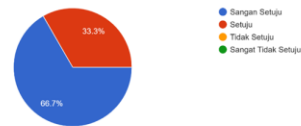
Apakah aplikasi memudahkan anggota untuk memeriksa kegiatan pengurus ?
6 responses



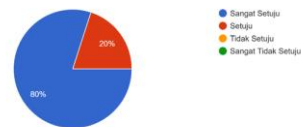
Apakah aplikasi memudahkan anggota untuk memeriksa riwayat transaksi ?
6 responses



Apakah aplikasi memudahkan anggota untuk melihat kegiatan koperasi secara keseluruhan ?
6 responses



Apakah aplikasi mempermudah admin untuk mengelola anggota ?
5 responses



IV. SIMPULAN

Pada bab ini akan dijelaskan secara keseluruhan tentang hasil akhir dari penelitian dan pembangunan aplikasi yang sudah dilakukan. Kesimpulan tersebut berisikan hal - hal yang mengacu pada hasil yang didapat. Berikut kesimpulan dari penelitian ini :

1. Berdasarkan pengujian *black box testing*, fitur sistem aplikasi terhadap pengguna member, pengurus, dan admin dapat digunakan dan berjalan dengan baik.
2. Berdasarkan pengujian *user experience* pada pengurus untuk pertanyaan “Apakah aplikasi mempermudah pengurus untuk membuat laporan SPT” mendapat hasil 100% dari 6 responden menjawab setuju.

3. Berdasarkan pengujian *user experience* pada member untuk pertanyaan “Apakah aplikasi memudahkan anggota untuk memeriksa kegiatan Pengurus” mendapat hasil 100% dari 6 responden menjawab setuju
4. Berdasarkan pengujian *user experience* pada admin untuk pertanyaan “Apakah aplikasi mempermudah admin untuk mengelola anggota ?” mendapat hasil 100% dari 5 responden menjawab setuju

Pada bab ini akan dijelaskan tentang saran yang diterima untuk aplikasi web agar dapat dikembangkan di kemudian hari. Dalam melakukan penelitian ini, sangat disadari sekali bahwa penelitian ini masih banyak kekurangan dalam beberapa hal. Berikut saran yang didapat pada penelitian ini:

1. Pada aplikasi bagian *Delete Asset* memerlukan fitur pada server yang dimana pengurus menghapus data tersebut sistem menanyakan alasan penghapusan, jika pengurus menghapus untuk alasan dijual atau semacamnya sistem akan meminta pengurus memasukan nominal penjualan agar sistem dapat menambahkan nominal tersebut ke dalam nominal uang kas
2. Pada aplikasi memerlukan fitur penyusutan nominal aset yang lebih mendalam agar dalam pembuatan laporan laba rugi, neraca sistem dapat memberikan estimasi yang lebih akurat
3. Pada aplikasi ini masih banyak menggunakan *sub-document* yang dimana menurut peneliti merupakan solusi yang tidak optimal oleh karena itu peneliti menyarankan untuk tidak menyimpan data dalam bentuk *sub-document*.

DAFTAR RUJUKAN

- [1] “Pembukuan Digital Mudahkan Laporan Keuangan UMKM.” <https://ekbis.sindonews.com/read/634079/34/pembukuan-digital-mudahkan-laporan-keuangan-umkm-1640002348> (accessed Aug. 04, 2022).
- [2] S. C. Cahyodi and R. W. Arifin, “Sistem Informasi Point Of Sales Berbasis Web Pada Colony Amaranta Bekasi,” *Information System For Educators And Professionals*, vol. 1, no. 2, pp. 189–204, 2017.
- [3] S. Dian Handy Permana and Faisal, “ANALISA DAN PERANCANGAN APLIKASI POINT OF SALE (POS) UNTUK Mendukung MANAJEMEN HUBUNGAN PELANGGAN,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 2, no. 1, pp. 20–28, 2015.
- [4] “Introduction to client-side frameworks - Learn web development | MDN,” Feb. 18, 2022. https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Introduction (accessed Feb. 22, 2022).
- [5] “Express/Node introduction - Learn web development | MDN,” Feb. 03, 2022. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction (accessed Feb. 23, 2022).
- [6] “7 Reasons Why VueJS Is So Popular | Kofi Group.” <https://www.kofi-group.com/7-reasons-why-vuejs-is-so-popular/> (accessed Feb. 23, 2022).
- [7] B. Botelho and J. Vaughan, “What is MongoDB? A definition from WhatIs.com,” Aug. 2020. <https://searchdatamanagement.techtarget.com/definition/MongoDB> (accessed Feb. 23, 2022).
- [8] “What is GraphQL?,” Jan. 08, 2019. <https://www.redhat.com/en/topics/api/what-is-graphql> (accessed Feb. 23, 2022).
- [9] “Working with a GraphQL Dataloader.” https://documentation.softwareag.com/webmethods/microservices_container/msc10-5/10-5_MSC_PIE_webhelp/index.html#page/integration-server-integrated-webhelp/to-graphql_11.html (accessed Feb. 23, 2022).
- [10] “What is Unified Modeling Language (UML)?” <https://www.visual-paradigm.com/guide/uml-unified->

modeling-language/what-is-uml/
(accessed Jun. 19, 2022).

- [11] T. Hamilton, "What is BLACK Box Testing? Techniques, Example & Types," Dec. 25, 2021. <https://www.guru99.com/black-box-testing.html> (accessed Feb. 23, 2022).