

PERANCANGAN SISTEM APLIKASI GUDANG TOKO ANGKASA JAYA MOTOR BERBASIS WEB MENGGUNAKAN NODE.JS

Ical F. Balino*¹⁾, Yos R. Beeh²⁾

1. Universitas Kristen Satya Wacana, Indonesia
2. Universitas Kristen Satya Wacana, Indonesia

Article Info

Kata Kunci: Aplikasi Gudang; Barang; Node.js; Pengelolaan; Web

Keywords: Goods; Management; Node.js; Warehouse Application; Web

Article history:

Received 29 September 2024

Revised 13 Oktober 2024

Accepted 4 November 2024

Available online 4 December 2024

DOI :

<https://doi.org/10.29100/jipi.v9i4.5587>

* Corresponding author.

Corresponding Author

E-mail address:

672018209@student.uksw.edu

ABSTRAK

Dalam menjalankan aktivitas bisnisnya, toko Angkasa Jaya Motor sering mengalami permasalahan dalam melakukan pengelolaan dan penyimpanan barang di gudang. Permasalahan terjadi dikarenakan toko masih menerapkan cara kerja manual dalam proses bisnisnya. Salah satunya adalah proses pencatatan barang yang masih menggunakan buku catatan. Data yang tercatat dalam buku catatan kemudian dipindahkan ke dalam spreadsheet tool. Pada proses ini, sinkronisasi data juga menjadi kendala. Data stok barang yang ada di gudang, di buku catatan, dan spreadsheet tool terdapat ketidaksesuaian data antara ketiga sumber tersebut sehingga mempengaruhi akurasi data barang. Penelitian bertujuan untuk membangun aplikasi gudang berbasis Web menggunakan Node.js untuk membantu dalam menghadapi permasalahan tersebut. Metode yang digunakan dalam penelitian ini adalah metode Research and Development (R&D). Penelitian ini menghasilkan sistem aplikasi gudang berbasis Web yang mampu membantu karyawan melakukan pengelolaan gudang dengan lebih praktis serta membuat pengelolaan gudang menjadi lebih optimal dan integrasi data membantu pemilik toko dalam mengakses informasi barang di gudang secara efisien dan real-time, sehingga keputusan bisnis dapat diambil berdasarkan data yang akurat.

ABSTRACT

In carrying out its business activities, Angkasa Jaya Motor stores often experience problems in managing and storing goods in the warehouse. The problem occurs because the store still employs manual work methods in its business processes. One of them is that the process of recording goods still utilizes a notebook. The data recorded in the notebook is then transferred to the spreadsheet tool. In this process, data synchronization is also a constraint. There are data discrepancies between the stock data in the inventory, notebooks, and spreadsheet tools, thereby affecting the accuracy of the goods data. The research aims to develop a Web-based warehouse application using Node.js to help address these issues. The method used in this research is the Research and Development (R&D) method. This research yields a Web-based warehouse application system that enables employees to conduct inventory management more practically and optimize warehouse operations and data integration helps store owners efficiently access real-time information goods on the inventory, so that enable business decisions to be made based on accurate data.

I. PENDAHULUAN

TEKNOLOGI Informasi (TI), atau yang dalam bahasa Inggris dikenal dengan sebutan *Information Technology* (IT) merupakan istilah yang secara umum digunakan untuk menyebutkan teknologi apapun yang dapat membantu manusia dalam mengkomunikasikan, membuat, menyimpan, mengubah, dan/atau menyebarkan informasi. Teknologi informasi memungkinkan komputasi dan komunikasi yang berkecepatan tinggi untuk video, suara, dan data. Salah satu penerapannya adalah dengan teknologi komputer. Orang-orang dalam suatu jaringan komputer dapat saling berhubungan, berkomunikasi dan menyebarkan informasi. Melalui teknologi informasi, sistem yang sudah terkomputerisasi akan lebih efektif dan efisien dibandingkan dengan sistem yang lama atau sistem manual [18]. Teknologi informasi sendiri memiliki peranan yang sangat penting bagi pelaku bisnis, instansi perusahaan swasta maupun pemerintah dalam kehidupan sehari-hari. Salah satu bagian dari teknologi informasi yang paling cepat berkembang yaitu teknologi *Web* atau *Aplikasi Web*. *Aplikasi Web* adalah suatu aplikasi yang berbentuk *client/server* yang dapat membentuk halaman-halaman *Web* berdasarkan permintaan pengguna. *Client* disini merupakan pengguna yang meminta halaman *Web* atau layanan, sedangkan *server*

merupakan penyedia layanan yang melayani permintaan dari pengguna. *Client* dan *server* berhubungan dalam suatu jaringan internet atau intranet [19].

Toko Angkasa Jaya Motor merupakan toko yang terletak di Jl. Lintas Kalimantan Poros Tengah, Kec. Belimbing, Kab. Melawi, Kalimantan Barat. Toko Angkasa Jaya Motor adalah usaha bisnis servis kendaraan roda dua dan menjual produk onderdil kendaraan roda dua seperti aki motor, oli mesin, rantai motor, ban, kampas dan tali rem, kampas kopling, busi, aksesoris motor, serta lampu motor. Komponen-komponen tersebut merupakan komponen onderdil yang paling banyak terjual pada toko ini. Dalam operasional bisnisnya, Toko Angkasa Jaya Motor mencatat rata-rata penjualan produk hingga sekitar 70 komponen onderdil per hari. Toko Angkasa Jaya Motor sudah berdiri selama kurang lebih 9 tahun dengan jumlah karyawan yang bekerja di toko saat ini berjumlah 6 orang. Dengan jumlah karyawan tersebut pekerjaan dibagi menjadi bagian yang menangani proses transaksi (penjualan dan pembelian), bagian yang mengelola gudang, dan bagian layanan servis motor.

Dalam menjalankan aktivitas bisnisnya, toko sering mengalami permasalahan dalam melakukan pencatatan, karyawan kesulitan dalam pencarian barang, dan pengelolaan barang pada gudang. Hal ini terjadi dikarenakan pada toko ini masih menerapkan banyak beban kerja atau cara kerja manual dalam proses bisnisnya, salah satunya adalah proses pendataan barang yang masih belum optimal karena sebagian pencatatan barang masih menggunakan buku catatan. Karyawan yang memiliki tanggung jawab dalam mengelola gudang harus secara manual melakukan pemeriksaan persediaan barang dan mencatatnya kedalam buku catatan. Tindakan ini perlu dilakukan secara berkala untuk mencatat barang yang telah habis, barang yang berada dalam kondisi hampir habis, dan barang yang mengalami kerusakan. Hal ini dapat menghabiskan banyak waktu dan tenaga dalam proses pengelolaan gudang, khususnya proses pengelolaan untuk mendapatkan serta mencatat data terbaru. Data yang tercatat dalam buku catatan kemudian dipindahkan ke dalam *spreadsheet tool*. Tujuannya adalah agar pemilik toko dapat menggunakan data tersebut untuk melakukan evaluasi dan membuat keputusan bisnis yang lebih efektif terkait pembelian barang, penanganan barang yang rusak, serta analisis penjualan. Pada proses ini, sinkronisasi data (integrasi data *real-time*) juga menjadi kendala [1][2]. Informasi mengenai stok barang yang ada di gudang, yang tercatat dalam buku catatan, dan *spreadsheet tool* terdapat perbedaan atau ketidaksesuaian data antara ketiga sumber tersebut sehingga mempengaruhi akurasi data barang. Hal ini mengakibatkan pemilik toko kesulitan dalam menentukan keputusan bisnis yang lebih efisien berdasarkan data yang kurang akurat tersebut, sehingga berdampak pada penurunan kualitas layanan kepada konsumen dan dapat menyebabkan kerugian pada usaha bisnis yang sedang dijalankan [3].

Perancangan sistem aplikasi gudang toko Angkasa Jaya Motor berbasis *Web* bertujuan untuk membantu karyawan dan pemilik toko dalam menghadapi permasalahan tersebut. Aplikasi berbasis *Web* dimanfaatkan karena mudah dikembangkan sesuai kebutuhan bisnis toko, dapat dimodifikasi sesuai keinginan, memiliki keunggulan pada aksesibilitas, berjalan secara *real-time*, penyimpanan data terintegrasi sepenuhnya, lintas *platform*, pembaruan yang mudah, dapat diakses dimanapun dan kapanpun, serta tidak terikat pada *device* tertentu [4]. Untuk mendukung perancangan sistem aplikasi berbasis *Web* dibutuhkan teknologi yang dapat berjalan di lingkungan *server* yang mampu menangani permintaan (*request*) dari *client* dan dapat diskalakan, salah satu teknologi tersebut adalah Node.js. Node.js adalah sistem perangkat lunak yang dimaksudkan untuk membantu dalam pengembangan aplikasi *Web* dan dapat berjalan di *server* [5]. Node.js ditulis dalam campuran bahasa pemrograman C++ dan juga JavaScript, mempunyai model *event driven* (basis *event*) dan *asynchronous I/O*. Karena mendukung *V8 Engine* buatan Google dan memiliki modul bawaan yang terintegrasi seperti modul HTTP, modul *Filesystem*, modul *Security*, dan modul lainnya [20].

Berdasarkan latar belakang tersebut, dapat dibentuk rumusan masalah yaitu bagaimana cara membangun sistem aplikasi gudang berbasis *Web* menggunakan Node.js untuk membantu optimalisasi pengelolaan gudang secara praktis dan keputusan bisnis dapat diambil berdasarkan data yang akurat?

Terdapat beberapa penelitian terdahulu, yaitu penelitian yang dilakukan oleh Hendri Cahyo (2014) dengan judul “Perancangan Aplikasi Pengelolaan Data Barang dan Data Nota Pembelian menggunakan Mobile Android dan Desktop (studi kasus: Laksana Elektronik)”. Temuan yang dimanfaatkan yaitu implementasi *Web Services* sebagai penyedia layanan data [6]. Perbedaan penelitian adalah penelitian yang sedang dilakukan menggunakan Node.js. Penelitian yang dilakukan oleh Christine Dewi dan Anjar Widhyo Sasongko (2018) dengan judul “Sistem Pelaporan Infrastruktur Berbasis Web menggunakan Node.js (Studi Kasus: PSDA Kota Salatiga)”. Temuan yang dimanfaatkan yaitu fitur *export* laporan kedalam bentuk *file pdf* [7]. Perbedaan penelitian adalah penelitian yang sedang dilakukan berfokus pada pengelolaan gudang. Penelitian yang dilakukan oleh Anugerah Christian Rompis dan Rizal Fathoni Aji (2018) dengan judul “Perbandingan Performa Kinerja Node.js, PHP, dan Python dalam Aplikasi REST”. Temuan yang dimanfaatkan yaitu Node.js sebagai teknologi dengan kinerja yang paling cocok untuk membangun aplikasi dengan arsitektur REST [8]. Perbedaan penelitian adalah penelitian yang sedang dilakukan bertujuan untuk merancang dan membangun aplikasi berbasis *Web*. Penelitian yang dilakukan oleh Anis Rohmadi dan Verdi Yasin (2020) dengan judul “Desain dan Penerapan Website Tata Kelola Percetakan pada CV Apicdesign Kreasindo Jakarta dengan Metode Prototyping”. Temuan yang dimanfaatkan yaitu implementasi

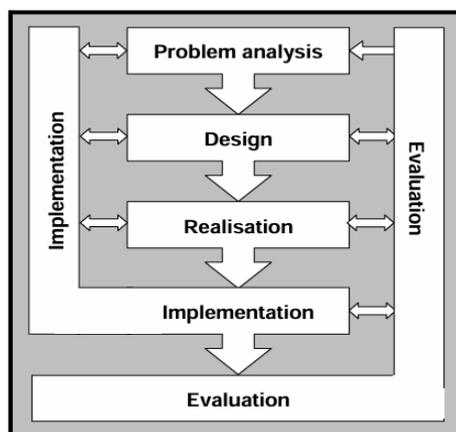
sistem untuk pengelolaan *vendor* dan produk [9]. Perbedaan penelitian adalah penelitian yang sedang dilakukan menggunakan teknologi Node.js dalam pengembangan *Web* yang lebih interaktif dan tidak statik. Penelitian yang dilakukan oleh S. L. Bangare, Dkk (2016) dengan judul “*Using Node.js to Build High Speed and Scalable Backend Database Server*”. Temuan yang dimanfaatkan yaitu implementasi Node.js dalam pengembangan *Database Server* untuk penyimpanan yang efisien dan transfer data [10]. Perbedaan penelitian adalah penelitian yang sedang dilakukan membahas mengenai pengembangan aplikasi secara keseluruhan untuk optimalisasi.

Dari penelitian-penelitian tersebut didapati bahwa aplikasi berbasis *Web* terbukti memberikan kemudahan bagi para pengguna dalam melakukan pengelolaan dan teruji memiliki keunggulan pada aksesibilitas, berjalan secara *real-time*, dan mudah dikembangkan sesuai kebutuhan bisnis dalam hal ini berarti bisnis toko Angkasa Jaya Motor. Aplikasi *Web* menggunakan Node.js sebagai *platform* dan teknologi dalam perancangannya. Node.js adalah pilihan yang baik untuk membuat aplikasi yang cepat, cepat dalam menangani respon, mudah diskalakan, berjalan secara *asynchronous*, dan cepat dalam operasi I/O. Selain itu, teknik *non-blockingnya* memungkinkan aplikasi dijalankan secara paralel oleh sistem tanpa harus menunggu operasi sebelumnya selesai. Berdasarkan fakta yang ada, untuk membantu optimalisasi pengelolaan gudang secara praktis dan keputusan bisnis dapat diambil berdasarkan data yang akurat maka sistem aplikasi gudang akan dibangun berbasis *Web* menggunakan teknologi Node.js.

Masing-masing penelitian tersebut memiliki studi kasus dan fokus penyelesaian masalah yang berbeda-beda dalam menggunakan teknologi informasi khususnya *Web*. Hal tersebut menjadikan penelitian yang dilakukan berbeda dengan penelitian yang telah disebutkan diatas mulai dari pembahasan, fokus penyelesaian masalah, dan studi kasus. Penelitian ini menitikberatkan pada perancangan aplikasi gudang toko Angkasa Jaya Motor, yang merupakan aspek penting dalam membantu optimalisasi pengelolaan gudang oleh karyawan, terutama dalam usaha bisnis yang dijalankan oleh toko. Berbeda dengan fokus penelitian-penelitian sebelumnya yang tidak membahas bagaimana pengembangan dan implementasi sistem aplikasi gudang untuk toko-toko seperti Angkasa Jaya Motor yang berbasis *Web* dengan menggunakan Node.js.

II. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah metode *Research and Development* (R&D), dipilih karena metode sesuai dengan sasaran penelitian. *Research and Development* adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut [11]. Tahap penelitian disesuaikan menggunakan metode *Research and Development* seperti yang ditunjukkan pada Gambar 1 [12].



Gambar 1. Metode *Research and Development* [12]

A. *Problem Analysis and Design*

Tahap pertama adalah *Problem Analysis*, sebelum aplikasi mulai dikembangkan dilakukan observasi dan wawancara dengan pihak toko Angkasa Jaya Motor serta menemukan kendala yang terjadi pada toko. Keterlibatan pihak atau orang-orang di toko Angkasa Jaya Motor sangat penting agar dapat mendata masalah apa saja yang sedang dialami oleh toko. Hasil dari proses ini adalah identifikasi model kerja dan indikator keberhasilan penelitian. Tahap kedua adalah *Design*, menganalisis setiap persyaratan secara menyeluruh dari setiap *requirement* yang diterima sehingga menemukan solusi untuk masalah yang ada. Setelah itu, desain model kerja dan batasan-batasan teknis dapat didefinisikan. Pada tahap ini, dilakukan pembuatan UML yang meliputi *Use Case* diagram, *Class* diagram dan *Activity* diagram.

B. Realization/Construction and Implementation

Tahap ini dimulai dari melakukan pembangunan purwarupa (*prototype*) dari solusi-solusi yang telah dirumuskan pada tahap *design*. *Prototyping* terdiri dari beberapa tahapan yang dimulai dengan membuat *wireframe*. *Wireframe* sendiri dapat dikatakan sebagai cetak biru dalam sebuah arsitektur. Tujuannya dari *wireframe* bukanlah desain visual, melainkan untuk menyampaikan susunan struktur aplikasi yang meliputi tata letak komponen dan navigasi. Setelah *wireframe* dibuat, langkah selanjutnya adalah perancangan *mockup*. Pada tahap ini barulah desain diberikan efek visual sehingga dapat memberikan gambaran nyata dari konsep yang akan dibangun. Tahap selanjutnya adalah *Implementation*, setelah *prototype* dirasa telah memecahkan permasalahan maka tahap berikutnya adalah melakukan pengembangan aplikasi berdasarkan *prototype* yang telah dibuat.

C. Testing, Evaluation, and Revision

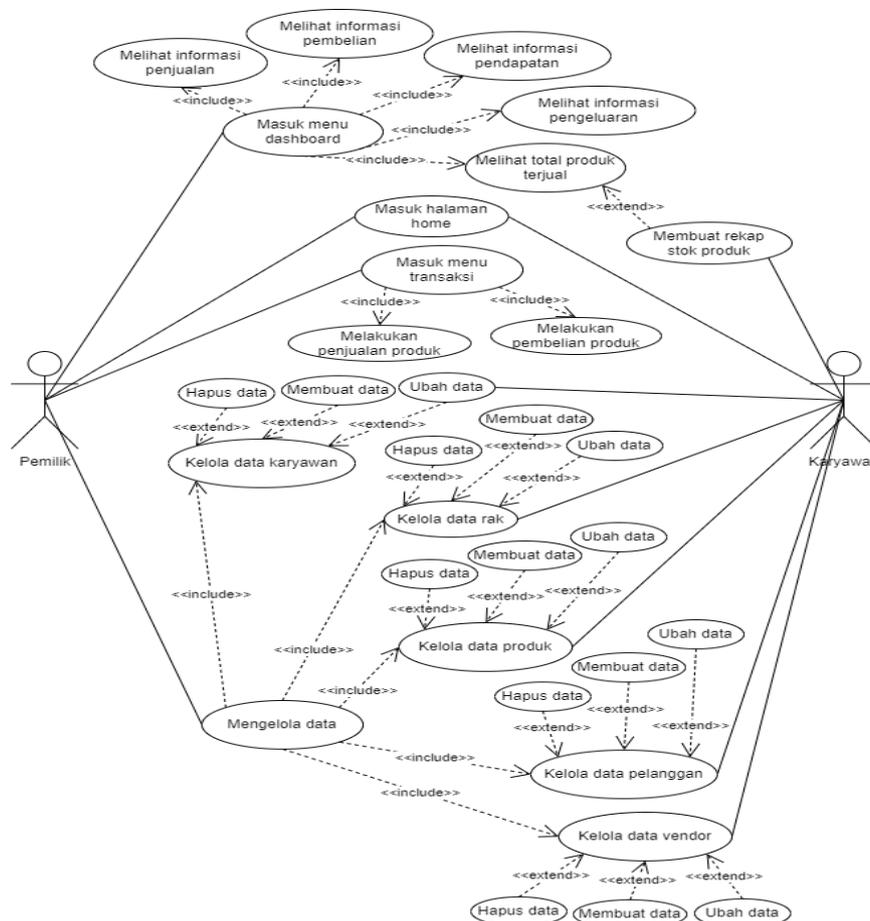
Pada tahap ini dilakukan pengujian *prototype* yang dihasilkan pada tahap sebelumnya. *Prototyping* yang telah dibangun diuji-cobakan ke pengguna untuk memeriksa apakah semua fungsi sistem berfungsi dengan baik sesuai dengan *requirement*. Pengujian dilakukan menggunakan *User Acceptance Test*. Pengguna perlu memberikan umpan balik (*feedback*), sehingga dapat dilakukan peninjauan dan perbaikan jika ada kesalahan pada aplikasi atau proses dari *prototype* yang harus dibenahi. Aplikasi yang telah selesai dibuat dan diuji kemudian diserahkan kepada pemilik toko Angkasa Jaya Motor untuk digunakan dalam membantu pengelolaan gudang.

III. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

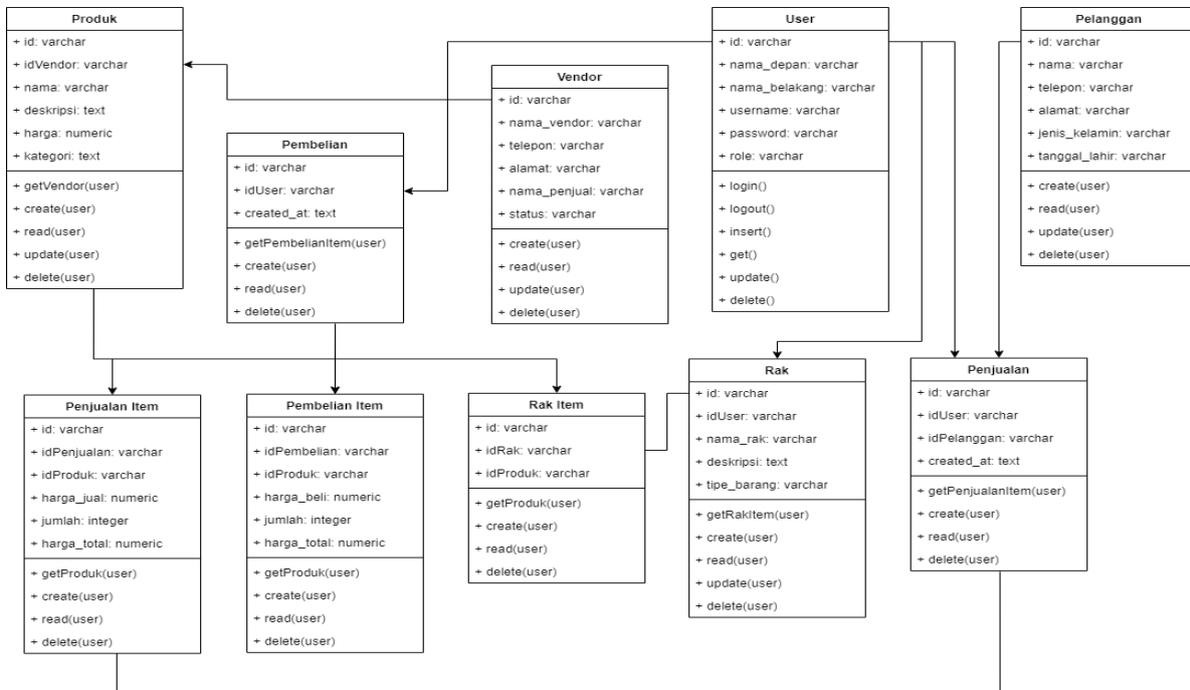
Hasil penelitian ini merupakan aplikasi berbasis *Web* menggunakan *Node.js* untuk pengelolaan gudang pada toko Angkasa Jaya Motor. Tahap awal yang dilakukan pada penelitian ini dimulai dengan merancang diagram.

Gambar 2 merupakan diagram *Use Case* dari aplikasi *Web* untuk menggambarkan interaksi pengguna dan sistem. *Use Case* pada gambar terdapat dua *actor* yaitu pemilik toko dan karyawan. Pemilik toko memiliki hak akses utama beserta semua fungsi atau operasi yang dapat dilakukan pada aplikasi. Sedangkan karyawan memiliki hak akses terbatas pada aplikasi, sehingga operasi yang digambarkan berdasarkan tugasnya yaitu kelola data. *Use Case* diagram ditunjukkan pada Gambar 2 berikut.



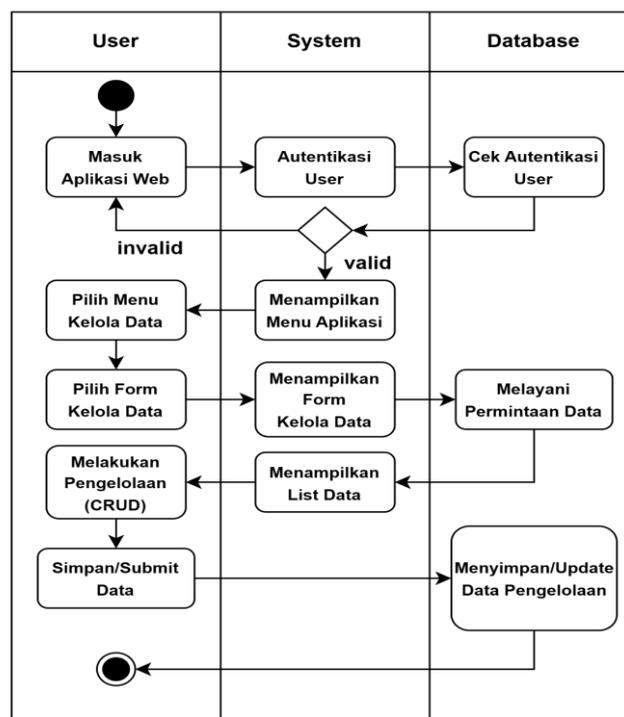
Gambar 2. Use Case Diagram

Gambar 3 merupakan *Class diagram* yang menampilkan 10 *Class* yang memiliki hubungan dan fungsinya masing-masing sehingga memberikan gambaran untuk membantu proses pengembangan aplikasi. Fungsi yang harus ada dari setiap *Class* digunakan untuk mengelola data seperti *insert*, *read*, dan *delete* data. *Class diagram* dimodelkan sehingga komponen antara *Class* berjalan sesuai dengan alur yang direncanakan. *Class diagram* ditunjukkan pada Gambar 3 berikut.



Gambar 3. *Class Diagram*

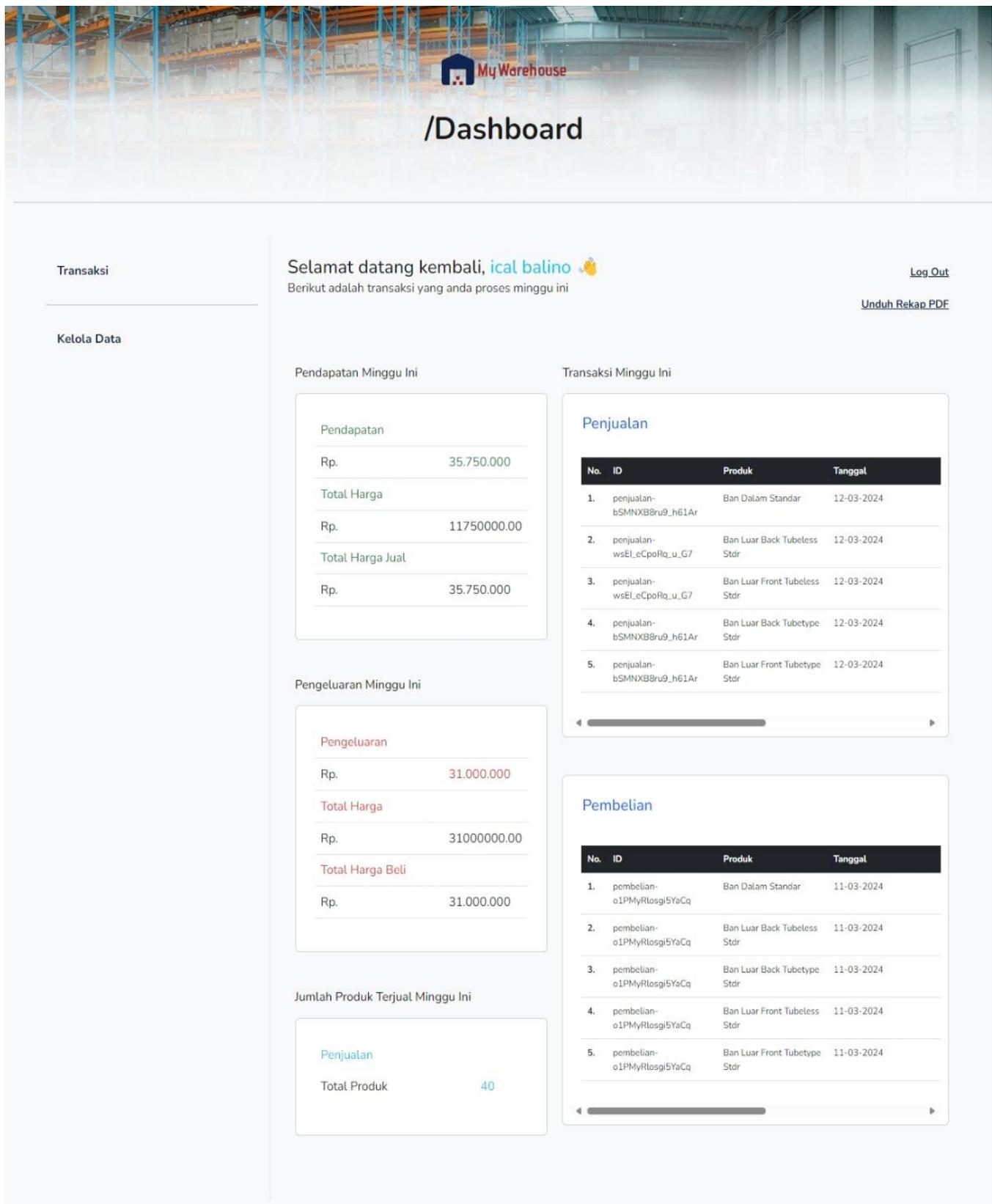
Gambar 4 merupakan *Activity diagram* yang menampilkan bagaimana proses aktivitas kelola data oleh *user*. Prosesnya dimulai dengan *user* membuka *Web* aplikasi gudang, kemudian pilih menu kelola data, pilih *form* data yang ingin dikelola, lakukan pengelolaan, setelah melakukan pengelolaan simpan perubahan atau tekan tombol *submit*. *User* juga dapat mengunduh laporan rekap mingguan dalam bentuk *pdf* pada menu/halaman *dashboard* di aplikasi. *Activity diagram* kelola data ditunjukkan pada Gambar 4 berikut.



Gambar 4. *Activity Diagram* Kelola Data

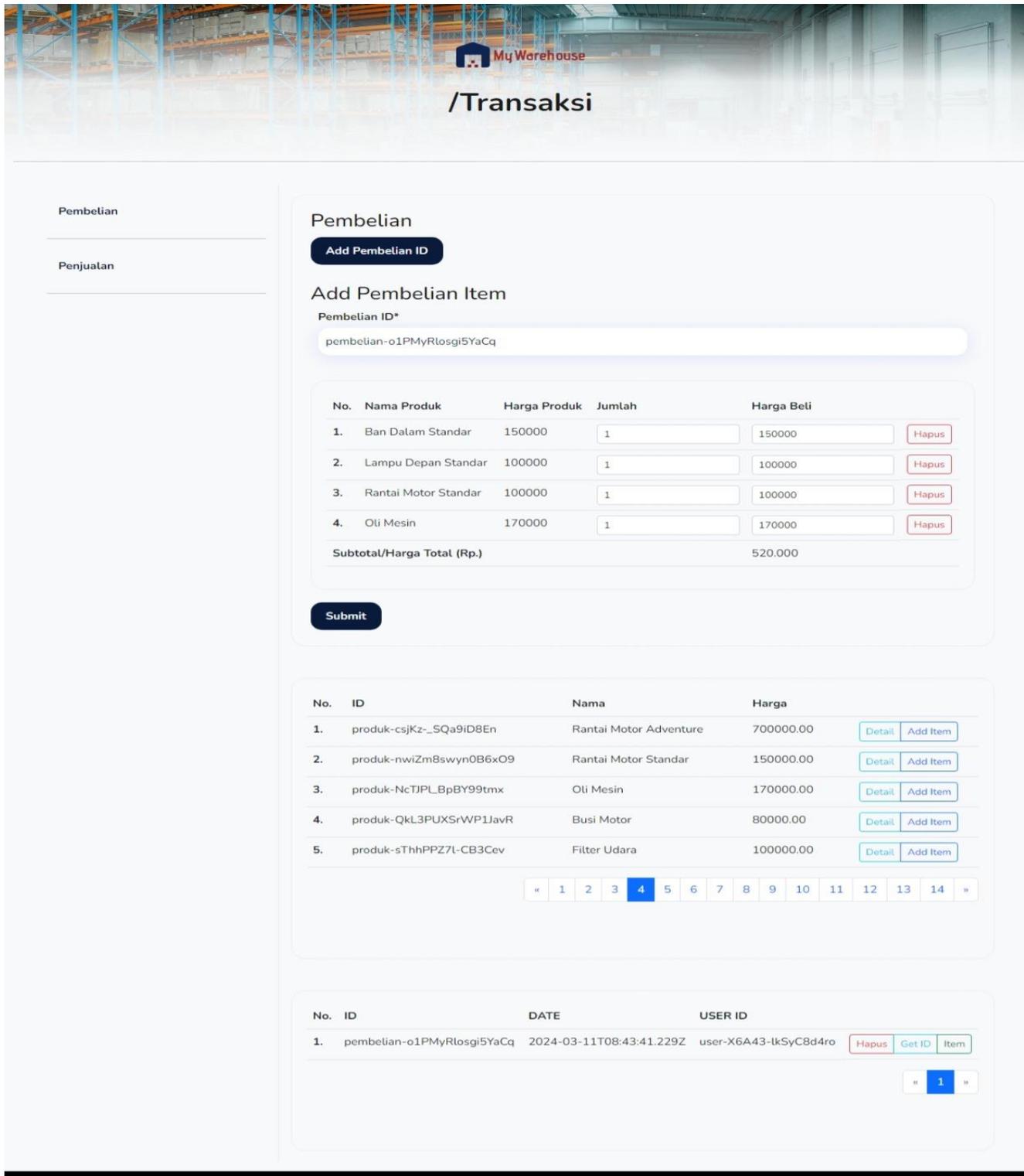
B. Realisasi dan Implementasi

Bagian pertama adalah *client* (admin) dengan perangkat untuk membuka aplikasi gudang toko Angkasa Jaya Motor berbasis *Web* menggunakan *browser*. Pada saat aplikasi *Web* diakses oleh *client* maka *server* akan memberikan respon berupa halaman *layout* tampilan dari aplikasi *Web*. *Frontend* dari aplikasi gudang toko Angkasa Jaya Motor dibangun menggunakan HTML, CSS, dan JavaScript *native* tanpa menggunakan *framework*. Berikut merupakan hasil tampilan aplikasi yang telah dibangun.



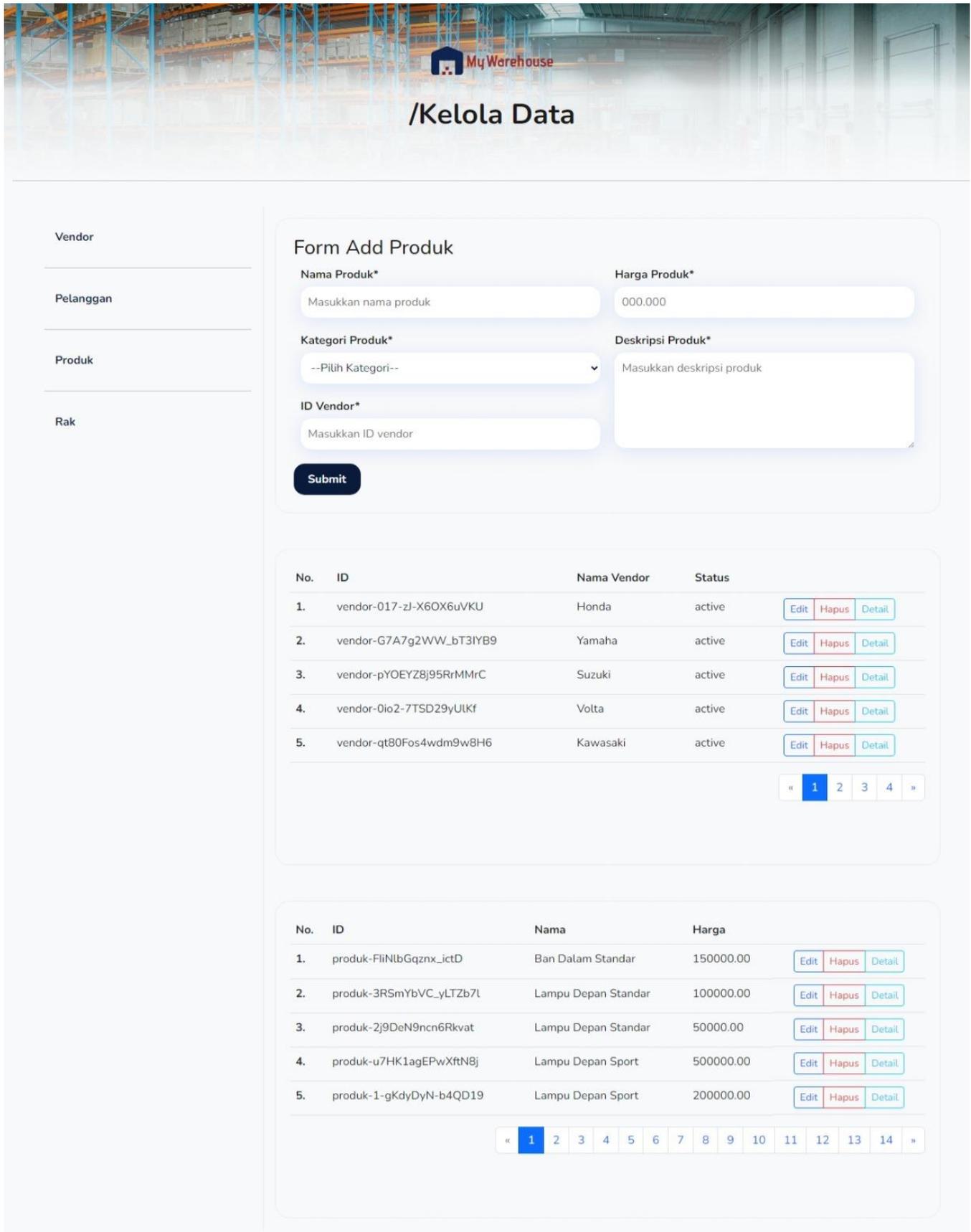
Gambar 5. Halaman Dashboard

Gambar 5 merupakan bagian halaman *Dashboard*, menggambarkan halaman informasi rekap proses transaksi yang dilakukan oleh *client* (admin). Detail halaman ini memberikan informasi mengenai *pendapatan*, *pengeluaran*, *jumlah produk terjual*, *transaksi*, *download rekap* dan menu *logout* serta menu navigasi ke halaman *Transaksi* dan halaman *Pengelolaan Data*.



Gambar 6. Halaman Transaksi

Pada Gambar 6 di atas merupakan bagian halaman *Transaksi*, yang terdapat menu *pembelian* dan *penjualan*. Menu pembelian digunakan untuk melakukan pembelian produk pada *vendor* yang kemudian akan menambahkan ketersediaan barang di gudang. Menu penjualan digunakan untuk melakukan transaksi penjualan produk dengan pelanggan. Data pembelian dan penjualan tersebut akan langsung disimpan kedalam *database* serta data *item* penjualan atau pembelian akan ditampilkan dalam bentuk *pagination view*.



Form Add Produk

Nama Produk*
Masukkan nama produk

Harga Produk*
000.000

Kategori Produk*
--Pilih Kategori--

Deskripsi Produk*
Masukkan deskripsi produk

ID Vendor*
Masukkan ID vendor

Submit

No.	ID	Nama Vendor	Status	
1.	vendor-017-zj-X6OX6uVKU	Honda	active	Edit Hapus Detail
2.	vendor-G7A7g2WW_bT3IYB9	Yamaha	active	Edit Hapus Detail
3.	vendor-pYOEYZ8j95RMMrC	Suzuki	active	Edit Hapus Detail
4.	vendor-0io2-7TSD29yULkf	Volta	active	Edit Hapus Detail
5.	vendor-qt80Fos4wdm9w8H6	Kawasaki	active	Edit Hapus Detail

« 1 2 3 4 »

No.	ID	Nama	Harga	
1.	produk-FiiNlbGqznx_ictD	Ban Dalam Standar	150000.00	Edit Hapus Detail
2.	produk-3RSmYbVC_yLTzb7L	Lampu Depan Standar	100000.00	Edit Hapus Detail
3.	produk-2j9DeN9ncn6Rkvat	Lampu Depan Standar	50000.00	Edit Hapus Detail
4.	produk-u7HK1agEPwXftN8j	Lampu Depan Sport	500000.00	Edit Hapus Detail
5.	produk-1-gKdyDyN-b4QD19	Lampu Depan Sport	200000.00	Edit Hapus Detail

« 1 2 3 4 5 6 7 8 9 10 11 12 13 14 »

Gambar 7. Halaman Pengelolaan Data

Pada Gambar 7 di atas merupakan bagian halaman *Kelola Data*, halaman kelola data merupakan halaman atau fitur yang digunakan pengguna untuk melakukan proses pengelolaan (*insert, view, update, delete*) data. Halaman ini meliputi pencatatan atau pengelolaan data *Vendor, Pelanggan, Produk*, dan *Rak* yang kemudian akan langsung disimpan kedalam *database* sehingga data terintegrasi secara *real-time*.

Backend aplikasi dibangun menggunakan Node.js. Dimulai dari *import module*, membuat HTTP server melalui *method Hapi.server()*, dan memasang *plugin* menggunakan *method await server.register()*. *Plugin* didaftarkan agar fungsi kelola data dan *plugin* lain berjalan sesuai alur logika bisnis yang telah dirancang. Selanjutnya menjalankan server secara *asynchronous* menggunakan *method await server.start()*. Backend aplikasi menggunakan PostgreSQL sebagai tempat penyimpanan data. PostgreSQL adalah *object-relational database management system* (ORDBMS) yang *open-source* [14].

TABEL I
 KODE PROGRAM FUNGSI FORM CRUD KELOLA DATA

```
const conForm = document.querySelector('.con-form')
if (conForm.innerHTML === "") {
    make_form_add_vendor()
    const formVendor = document.querySelector('#form-add-vendor')
    formVendor.addEventListener('submit', async (event) => { // ... })
    const tombolProduk = document.querySelector('#tombol-produk')
    tombolProduk.addEventListener('click', async () => {
        const formProduk = document.querySelector('#form-add-produk')
        formProduk.addEventListener('submit', async (event) => { // ... })
        produkList.addEventListener('click', async (event) => {
            if(event.target.classList.contains('hapus-produk')) { // ... }
            else if (event.target.classList.contains('edit-produk')) { // ... }
            else if (event.target.classList.contains('detail-produk')) { // ... }
        })
    })
    const tombolEtalase = document.querySelector('#tombol-etalase')
    tombolEtalase.addEventListener('click', async () => {
        if (data[0] !== undefined && data[0] !== null && data[0] !== "") {
            const formEtalase = document.querySelector('#form-add-etalase')
            formEtalase.addEventListener('submit', async (event) => { // ... })
            const hapusRak = document.querySelectorAll('.hapus-rak')
            hapusRak.forEach((hapus) => { // ... })
            const editRak = document.querySelectorAll('.edit-rak')
            editRak.forEach((edit) => { edit.addEventListener('click', async (event) => { // ... }) })
            const detailRak = document.querySelectorAll('.detail-rak')
            detailRak.forEach((detail) => { detail.addEventListener('click', async (event) => { // ... }) })
        }
    })
}
```

Pada Tabel Kode Program I di atas merupakan fungsi dalam mendukung operasional seperti *input, view, update, delete* pada menu kelola data (*vendor, produk, pelanggan, rak*) dengan *form*. *Form* digunakan untuk melakukan pengelolaan termasuk pencatatan data dan kode program juga berfungsi untuk menampilkan data barang dengan akurat serta menjadikan proses pengelolaan lebih praktis yang membuat pengelolaan gudang menjadi lebih optimal.

TABEL II
 KODE PROGRAM FUNGSI HANDLER PRODUK (BARANG)

```
class ProdukHandler {
    constructor (vendorService, produkService, validator) { // ... }
    async postProdukHandler (request, h) {
        try {
            this._validator.validateProdukPayload(request.payload)
            const { vendorId } = request.params
            const { nama, deskripsi, harga, kategori } = request.payload
            await this._vendorService.verifyVendorExist(vendorId)
            const productId = await this._produkService.addProduk({ nama, deskripsi, harga, kategori }, vendorId)
            const response = h.response({
                status: 'success',
                message: 'Produk berhasil ditambahkan',
                data: { productId } })
            response.code(201)
            return response
        } catch (error) { console.error(error) }
    }
}
module.exports = ProdukHandler
```

Tabel Kode Program II merupakan salah satu fungsi *handler* khususnya untuk membuat *handler produk/barang*. *Handler* berfungsi untuk menangani permintaan dari layanan yang berbeda-beda terhadap sumber daya dan berfungsi sebagai *controller* untuk melakukan komunikasi data pada fungsi *database service*. Kontrol data yang terpusat membantu pemilik toko dalam mengakses informasi barang serta persediaan barang di gudang secara efisien dan *real-time*.

TABEL III
 KODE PROGRAM UNTUK EXPORT DATA KE DALAM FILE PDF

```

async writePdfHandler (dataPenjualan, user, transaksi, infoTransaksi) {
    const doc = new PDFDocument()
    const filename = 'src/api/file/output/laporan_mingguan_fix.pdf'
    doc.pipe(createWriteStream(filename))
    doc.fontSize(16).font('Helvetica-Bold').text('Laporan Rekap Mingguan Toko Angkasa Jaya Motor', { align: 'center'
    }).moveDown(0.3) // Judul
    const keuntungan = pendapatanObj.Pendapatan - pengeluaranObj.Pengeluaran
    const profit = keuntungan.toLocaleString('id-ID')
    doc.fontSize(10).font('Helvetica-Oblique').text(`Keuntungan (Profit) : Rp. ${profit}`, doc.x).moveDown(0.5)
    doc.font('Helvetica').fontSize(12).text('Pendapatan (Income) Minggu Ini', { underline: true }).moveDown(0.5)
    generateTableInfo(doc, ['Keterangan', 'Rp.'], Object.entries(pendapatanObj))
    doc.font('Helvetica-Bold').fontSize(10).text('Penjualan', doc.x).moveDown(0.5)
    generateTableInfoTransaksi(doc, ['No.', 'Tanggal', 'Produk', 'Jumlah', 'Harga Jual', 'Admin'], penjualanItemAll)
    function generateTableInfo (doc, headers, data) {
        const startX = doc.x
        const startY = doc.y
        if (data.length === 3) {
            doc.rect(startX, startY, (headers.length * 150), 25).fill('lightgray')
            let cols_width = startX + cellPadding
            for (const header of headers) { // ... }
                for (let row = 0; row < data.length; row++) { for (let col = 0; col < rowData.length; col++) { // ... } }
            }
        }
        function generateTableInfoTransaksi (doc, headers, data) { // ... }
    }
    doc.end()
    console.log(`Laporan telah dibuat: ${filename}`)
}
    
```

Tabel Kode Program III adalah fungsi program untuk *export* data ke dalam *file pdf*. Tujuannya agar pemilik toko dapat menggunakan informasi tersebut untuk melakukan evaluasi serta membuat keputusan bisnis yang lebih efektif terkait pembelian, barang yang rusak, barang hampir habis, dan analisis penjualan.

C. Pengujian

Acceptance Test merupakan jenis pengujian yang dilakukan untuk menyesuaikan perangkat lunak (*services*) yang dibangun sesuai dengan fungsionalitas dan *requirement* yang telah disepakati. Pengujian akan menentukan diterima atau tidaknya perangkat lunak yang sedang atau telah dibangun, dengan menggunakan *Acceptance Test Driven Development* [15]. *User Acceptance Test* yang telah diterapkan dan diuji dapat dilihat pada Tabel IV berikut.

TABEL IV
 PENGUJIAN MENGGUNAKAN ACCEPTANCE TEST DRIVEN DEVELOPMENT

Skenario Use Case	Endpoint Test	Hasil yang diharapkan	Hasil	Kesimpulan
Melakukan login dengan username dan password	/authentications	Dapat melakukan autentikasi dengan cara login melalui form	Berhasil login dan masuk ke halaman selanjutnya	Sukses
Melakukan proses pengelolaan data (vendor, produk, pelanggan, rak)	/vendor /produk/{vendorId} /pelanggan /rak	Dapat menampilkan form yang sesuai untuk insert data (vendor, produk, pelanggan, rak) dan menambahkan data	Berhasil menampilkan form yang dipilih dan insert data kedalam database	Sukses
	/vendor /produk /pelanggan /rak	Dapat menampilkan list data (vendor, produk, pelanggan, rak) pada halaman pengelolaan data	Berhasil menampilkan list data (vendor, produk, pelanggan, rak) dalam bentuk pagination	Sukses
	/vendor/{vendorId} /produk/{produkId} /pelanggan/{pelangganId} /rak/{rakId}	Dapat melihat detail data, menghapus, dan memperbarui data yang dipilih pada halaman pengelolaan	Berhasil menampilkan informasi detail data yang dipilih, menghapus data, dan memperbarui data	Sukses
Melakukan transaksi	/pembelian/{pembelianId}/pembelianItem /penjualan/{penjualanId}/produkId/penjualanItem	Dapat membuat transaksi penjualan atau pembelian serta dapat menambahkan record item pembelian dan item penjualan	Berhasil membuat transaksi serta menambahkan record item transaksi ke dalam database	Sukses
	/pembelian/{pembelianId}/pembelianItem /penjualan/{penjualanId}/penjualanItem	Dapat menampilkan list of view pagination data pembelian item atau data penjualan item pada halaman transaksi	Berhasil menampilkan list of view data pembelian item atau data penjualan item dalam pagination	Sukses
Masuk ke halaman dashboard	/produk /pendapatan /transaksiInfo	Dapat menampilkan informasi mingguan mengenai jumlah record penjualan, pembelian, pendapatan, pengeluaran, dan jumlah total produk terjual serta stok produk di gudang	Berhasil menampilkan informasi rekap data mingguan dalam jumlah dan pagination	Sukses

Pengujian dilakukan dengan membagi *services* menjadi bagian-bagian (*unit*) kecil yang terpisah. Proses transaksi data atau interaksi antara aplikasi *frontend* dan *backend* ini dilakukan melalui layanan *Web Service* ini. Program dirancang untuk memisahkan komponen aplikasi yang dibuat khususnya komponen yang digunakan untuk kelola data dengan komponen lainnya agar menjadi komponen terpisah dan terisolasi. Komponen tersebut berupa logika bisnis, *database service*, utilitas, atau *validator*. Setelah melakukan pengujian, *services* dari sistem aplikasi yang dikembangkan sudah berjalan sesuai dengan fungsionalitasnya. Sistem aplikasi gudang berbasis *Web* yang dikembangkan menggunakan Node.js ini telah berjalan sesuai dengan alur logika bisnis yang diharapkan yaitu memberikan layanan untuk mendukung dalam melakukan optimalisasi pengelolaan pada gudang.

Salah satu keunggulan dari aplikasi berbasis *Web* yang telah dibangun menggunakan Node.js adalah dapat melakukan sinkronisasi data yang terintegrasi secara *real-time*. Kontrol data yang terpusat membantu pemilik toko dalam mengakses informasi barang serta stok barang di gudang secara praktis dan optimal, sehingga keputusan bisnis dapat diambil berdasarkan data yang akurat.

Telah disebutkan sebelumnya bahwa proses transaksi atau interaksi antara aplikasi *frontend* dan *backend* ini dilakukan melalui *Web Service*, tetapi dalam pengembangan penelitian ini belum menggunakan HTTPS sebagai protokol dalam mendistribusikan atau transaksi data antara *client* dan *server*. Sehingga menjadikan ini sebagai keterbatasan dari penelitian yang sedang dilakukan sekaligus batasan dari penelitian yaitu tidak membahas sisi keamanan aplikasi.

Pengujian kedua adalah dengan membagikan kuesioner dan menggunakan skala *Likert* untuk penghitungannya. Skala *Likert* diterapkan pada hasil kuesioner yang dibagikan kepada pemilik dan karyawan toko dengan jumlah total 6 orang serta total pernyataan sebanyak 8. Skala *Likert* adalah skala yang dapat digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang suatu fenomena [16]. Berikut pernyataan dari kuesioner yang dibagikan:

- 1) Menu-menu pada aplikasi berbasis *Web* ini mudah untuk dioperasikan.
- 2) Aplikasi mampu melakukan operasional (*Input, View, Update, Delete*) data pada menu *Kelola Data*.
- 3) Aplikasi mampu mencatat, menyimpan dan melayani permintaan data secara akurat.
- 4) Aplikasi mampu menampilkan detail data yang akurat sesuai dengan permintaan pengguna.
- 5) Aplikasi mampu menampilkan informasi persediaan barang pada gudang dengan akurat.
- 6) Aplikasi mampu menampilkan list informasi transaksi mingguan pada menu *Dashboard*.
- 7) Aplikasi mampu melakukan/melayani pembelian dan penjualan pada menu *Transaksi*.
- 8) Aplikasi dapat melakukan *export data PDF* (memberikan/menampilkan hasil unduh laporan rekap mingguan dalam bentuk pdf).

Berikut skor dari setiap nilai jawaban pada kuesioner dengan skala 1-5 seperti pada Tabel V di bawah ini.

TABEL V
 SKALA PENILAIAN

Skor	Nilai
5	Sangat Setuju
4	Setuju
3	Ragu-ragu
2	Tidak Setuju
1	Sangat Tidak Setuju

Rumus persamaan untuk menghitung skala *Likert* adalah dengan mencari jumlah *Total Skor* (skala 1-5) terlebih dahulu dari masing-masing pernyataan. Kemudian, hitung *Skor Maksimum* dengan cara $5 \times Total Responden$. Berikut rumus persamaan skala *Likert*.

$$Hasil Likert (\%) = (Total Skor / Skor Maksimum) \times 100\%$$

TABEL VI
 HASIL PERHITUNGAN KUESIONER PENGUJIAN

Pernyataan	Skala Penilaian					Total Skor	Hasil (%)
	SS (5)	S (4)	RG (3)	TS (2)	STS (1)		
1	11	9	1	-	-	94	89,52
2	12	9	-	-	-	96	91,43
3	6	15	-	-	-	90	85,71
4	13	6	2	-	-	95	90,48
5	12	8	1	-	-	95	90,48
6	10	10	1	-	-	93	88,57
7	5	15	1	-	-	88	83,81
8	14	7	-	-	-	98	93,33

Pada Tabel VI di atas merupakan hasil pengujian dengan membagikan kuesioner kepada responden. Berdasarkan hasil pada tabel, sebanyak 91% dari responden menyatakan aplikasi mampu melakukan operasional (*input, view, update, delete*) data pada menu kelola data, 90% responden menyatakan aplikasi mampu menampilkan detail data yang akurat sesuai dengan permintaan pengguna dan aplikasi mampu menampilkan informasi persediaan barang pada gudang dengan akurat, serta 93% responden menyatakan aplikasi dapat melakukan *export* data PDF (memberikan/menampilkan hasil unduh laporan rekap mingguan dalam bentuk *pdf*).

Dari hasil pengujian yang telah dilakukan, ruang lingkup penerapan pengujian berfokus dan khusus digunakan pada toko Angkasa Jaya Motor. Pengujian dilakukan dengan membagikan kuesioner kepada responden yang berfokus pada orang-orang di toko Angkasa Jaya Motor (karyawan dan pemilik toko). Keterlibatan pihak atau orang-orang di toko Angkasa Jaya Motor sangat penting sehingga hasil pengujian yang diperoleh tersebut didasarkan dari data responden pihak toko terkait tingkat pemahaman saat menggunakan fitur atau layanan yang ada di aplikasi. Berdasarkan hasil pengujian melalui kuesioner menunjukkan aplikasi belum sangat baik pada bagian *user interface* khususnya pendekatan menggunakan *Mobile First Approach* dari *layout* menu-menu aplikasi (*frontend*) yang telah dikembangkan. Hal ini merupakan *feedback* yang bertujuan agar penelitian selanjutnya dapat menjadikan *feedback* yang diberikan oleh responden ini sebagai dasar dalam pengembangan aplikasi lebih lanjut.

Perbandingan hasil dari pendekatan dengan menggunakan *platform Web* dan teknologi Node.js pada penelitian ini dengan penelitian lain adalah sebagai berikut:

- Penelitian dengan judul “*Sistem Pelaporan Infrastruktur Berbasis Web Menggunakan Node.js (Studi Kasus: Dinas Bina Marga dan PSDA Kota Salatiga)*”. Memaparkan hasil penelitian bahwa Dinas Bina Marga dan PSDA telah menyediakan berbagai fasilitas bagi masyarakat untuk memberikan informasi mengenai infrastruktur. Informasi dapat diberikan melalui pemberitahuan langsung ke kantor dinas dan pemberitahuan melalui sms. Penyampaian informasi yang masih manual maupun berupa teks tersebut lalu diubah menjadi sistem yang dapat dilakukan secara online yang berbasis *Web*. Hasil dari penelitian menunjukkan bahwa dengan adanya sistem pelaporan dapat membantu pihak dinas Bina Marga dan PSDA kota Salatiga mengetahui detail informasi laporan infrastruktur dan melacak lokasi yang terdapat permasalahan infrastruktur. Aplikasi ini dapat membantu pihak dinas Bina Marga dan PSDA kota Salatiga dalam menyajikan laporan masalah pada infrastruktur. Penggunaan Node.js pada *server-side* memudahkan dalam pembangunan aplikasi *Web* karena mendukung komunikasi *asynchronous* yang dapat membuat kinerja sistem menjadi lebih ringan [7].
- Penelitian dengan judul “*Using Node.js to Build High Speed and Scalable Backend Database Server*”. Menjelaskan bahwa ada berbagai *3rd party cloud database providers* (Google, AmazonAWS, Mongolab) bagi *developers* untuk memulai membangun aplikasi tanpa mengkhawatirkan tentang *backend services*. Penelitian ini menjelaskan bahwa metode ini sangat tidak efisien, selalu ada kekhawatiran (*concern*) terhadap privasi data karena data tidak disimpan di *local machine* (mesin pribadi). Oleh karena itu dibangun kerangka kerja menggunakan Node.js untuk membangun *server database backend* yang sangat *scalable* dan berkecepatan tinggi untuk pengembangan berbasis *Web*. Hasil penelitian ini menyatakan Node.js adalah sebuah *server-side platform* yang digunakan dalam membangun *real-time application* karena Node.js memiliki *event-driven architecture* dan *non-blocking I/O*. Hal tersebut membuat Node.js ditemukan 10 kali lebih cepat dalam operasi I/O, membuat *developer* bebas dari *server-side coding* yang membosankan serta dapat membangun aplikasi bersama-sama dengan *hosting local machine* pribadi sebagai *database server* [10].

IV. KESIMPULAN

Dari hasil penelitian dan pengujian, disimpulkan bahwa pembangunan sistem aplikasi gudang berbasis *Web* menggunakan Node.js dimulai dengan pembuatan tampilan *layout (frontend)* aplikasi, pengembangan *backend* aplikasi (termasuk *server*, fungsi-fungsi *handler*, dan *database services*), implementasi *web services*, dan menentukan rute-rute API. Integrasi layanan dan fungsi-fungsi *handler* yang telah diterapkan membuat aplikasi *Web* dapat digunakan untuk melakukan pengelolaan gudang dengan lebih praktis serta membuat pengelolaan gudang menjadi lebih optimal. Sinkronisasi data yang terpusat membantu pemilik toko dalam mengakses informasi barang di gudang secara efisien dan *real-time*, sehingga keputusan bisnis dapat diambil berdasarkan data yang akurat.

DAFTAR PUSTAKA

- [1] M. Shodiq, R. Wongso, R. S. Pratama, E. Rhenardo, and Kevin, "Implementation of Data Synchronization with Data Marker using Web Service Data," *Procedia Computer Science*, vol. 59, pp. 366–372, 2015, doi: 10.1016/j.procs.2015.07.538.
- [2] U. Dayal, M. Castellanos, A. Simitsis, and K. Wilkinson, "Data Integration Flows for Business Intelligence," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, Mar. 2019. Accessed: Feb. 02, 2024. [Online]. Available: <http://dx.doi.org/10.1145/1516360.1516362>.
- [3] Institut Teknologi Telkom Surabaya, "Sains Data: Meningkatkan Keputusan Bisnis yang Lebih Baik," *Pendaftaran IT Telkom Surabaya - Admisi S1 IT Telkom Surabaya*, May 16, 2023. <https://smb.itelkom-sby.ac.id/sains-data-meningkatkan-keputusan-bisnis-yang-lebih-baik/> (accessed Feb. 02, 2024).
- [4] F. Noviyanto, "Implementation of Ria Technology to Develop Real Time Access Web Application (Case Study: Online Cellular Phone Order)," *Perpustakaan Universitas Gadjah Mada*, Jan. 01, 2017. https://etd.repository.ugm.ac.id/home/detail_pencarian/148389 (accessed Feb. 21, 2024).
- [5] R. Fajrin, "Pengembangan Sistem Informasi Geografis Berbasis Node.js untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia," *Jurnal Komputer Terapan*, vol. 3, no. 1, pp. 33–40, 2017.
- [6] H. C. Afrilianto, "Perancangan Aplikasi Pengelolaan Data Barang dan Data Nota Pembelian menggunakan Mobile Android dan Desktop: studi kasus Laksana Elektronik," *Universitas Kristen Satya Wacana Institutional Repository*, Jan. 01, 2014. <http://repository.uksw.edu/handle/123456789/8767> (accessed Feb. 16, 2024).
- [7] C. Dewi and A. W. Sasongko, "Sistem Pelaporan Infrastruktur Dinas Bina Marga dan PSDA Kota Salatiga menggunakan Node.js Berbasis Web," *Indonesian Journal of Computing and Modeling*, vol. 1, no. 1, pp. 10–17, 2018.
- [8] A. C. Rompis and R. F. Aji, "Perbandingan Performa Kinerja Node.js, PHP, dan Python dalam Aplikasi REST," *CogITo Smart Journal*, vol. 4, no. 1, pp. 171–187, 2018.
- [9] A. Rohmadi and V. Yasin, "Desain dan Penerapan Website Tata Kelola Percetakan pada CV Apicdesign Kreasindo Jakarta dengan Metode Prototyping," *JISICOM (Journal of Information System, Informatics and Computing)*, vol. 4, no. 1, pp. 70–85, 2020.
- [10] S. L. Bangare, S. Gupta, M. Dalal, and A. Inamdar, "Using Node.js to Build High Speed and Scalable Backend Database Server," *International Journal of Research in Advent Technology*, vol. 4, p. 19, 2016.
- [11] Okpatrioka, "Research and Development (R&D) Penelitian yang Inovatif dalam Pendidikan," *Dharma Acarya Nusantara: Jurnal Pendidikan, Bahasa dan Budaya*, Mar. 2023. <https://e-journal.nalanda.ac.id/index.php/jdan/article/view/154/150> (accessed Mar. 01, 2024).
- [12] S. Gustiani, "Research and Development (R&D) Method as a Model Design in Educational Research and Its Alternatives," *Holistics Journal Hospitality and Linguistics*, vol. 11, no. 2, Dec. 2019.
- [13] G. Pomberger and R. Weinreich, "The Role of Prototyping in Software Development," in *TOOLS 1994: 13th International Conference on Technology of Object-Oriented Languages and Systems*, B. Magnusson, Ed. New York [u.a.]: Prentice Hall, 1994. Accessed: Mar. 30, 2024. [Online]. Available: https://www.researchgate.net/publication/220877649_The_Role_of_Prototyping_in_Software_Development.
- [14] M. A. Rosid, "Implementasi JSON untuk Minimalisasi Penggunaan Jumlah Kolom suatu Tabel pada Database PostgreSQL," *JOINCS (Journal of Informatics, Network, and Computer Science)*, vol. 1, no. 1, p. 33, Jul. 2017, doi: 10.21070/joincs.v1i1.802.
- [15] Sugiarto, B. Nugroho, and A. D. Putri, "Penggunaan Metode Acceptance Test Driven Development pada Proses Acceptance Test menggunakan Co-deception," *Scan: Jurnal Teknologi Informasi dan Komunikasi*, vol. 14, no. 1, pp. 45–50, Jan. 2019, doi: 10.33005/scan.v14i1.1460.
- [16] Rohmad and Supriyanto, "Pengantar Statistika: Panduan Praktis Bagi Pengajar dan Mahasiswa," *Repository UIN Profesor Kyai Haji Saifuddin Zuhri*, 2016. <https://eprints.uinsaizu.ac.id/10985/> (accessed Mar. 31, 2024).
- [17] Hapi, "The Simple, Secure Framework Developers Trust," *hapi.dev*, 2012. Accessed: Nov. 15, 2023. [Online]. Available: <https://hapi.dev/>.
- [18] S. Bakhri, F. Hanif, and A. Haidir, "Rancang Bangun Aplikasi Kasir Penjualan Susu Berbasis Web pada Alomgada Kids Jakarta," *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 5, no. 1, pp. 47–54, 2020, doi: 10.31294/ijcit.v5i1.6397, 2020.
- [19] A. P. Mangunsong, "Pembangunan Situs Web Jual Beli Baju Batak," *e-journal.uajy.ac.id*, May 25, 2016. Accessed: Nov. 23, 2023. [Online]. Available: <http://e-journal.uajy.ac.id/id/eprint/10934>.
- [20] R. Fajrin, "Pengembangan Sistem Informasi Geografis Berbasis Node.js untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia," *Jurnal Komputer Terapan*, vol. 3, no. 1, pp. 33–40, 2017.