

Progressive Web Application Persewaan

Sahirul Alim Tri Bawono^{1*}, Rudi Hartono¹, Agus Purbayu¹, Ramadhan Wahyu Indra Pradana¹.

¹Prodi Diploma III Teknik Informatika, Universitas Sebelas Maret

*Email: sahirul@staff.uns.ac.id

Info Artikel

Kata Kunci :

persewaan barang, progressive web application, nuxt.

Keywords :

rent, progressive web application, nuxt.

Tanggal Artikel

Dikirim : 12 April 2024

Direvisi : 11 November 2024

Diterima : 21 November 2024

Abstrak

Persewaan adalah tempat untuk meminjam barang dengan imbalan uang tertentu tergantung pada nilai dan lama meminjam barang. Barang yang banyak memerlukan pencatatan yang baik bagi pemilik barang selain itu tempat untuk menawarkan barang yang akan disewakan dibutuhkan untuk dapat mempertemukan kebutuhan dari penyewa. Nuxt adalah salah satu *framework* untuk membuat PWA (*Progressive Web Application*) yang digunakan untuk membuat aplikasi dan diagram yang digunakan untuk analisis adalah usecase serta teknik pengujian yang digunakan adalah *black-box*. Aplikasi yang dibuat sudah sesuai dengan kebutuhan (*requirement system*) dan tidak ditemukan error ketika menjalankan aplikasi

Abstract

Rental is a place to borrow goods in exchange for certain money depending on the value and duration of borrowing the goods. Renting is something that usually happens in life when the need for goods is only for a certain period of time and is not used again later. Nuxt is a framework for making PWA (Progressive Web Applications) which is used to create applications and the diagrams used for analysis are use cases and the testing technique used is black-box. The application made is in accordance with the requirements (requirement system) and no errors were found when running the application.

1. PENDAHULUAN

Sewa menyewa menjadi semakin hari semakin diminati. Setiap barang seperti mobil, tanah atau rumah memerlukan biaya untuk perawat dan beberapa terdapat pajak didalamnya. Dengan menyewa seseorang dapat menggunakan barang tersebut tanpa berfikir tentang pajak dan perbaikan hanya berfikir untuk membayar sejumlah uang sehingga dapat menggunakan barang tersebut dalam jangka waktu yang disepakati bersama dengan pemilik barang. Pemilik barang jika memiliki banyak barang yang disewakan maka diperlukan pencatatan yang baik. Selain itu juga terdapat barang baru yang ingin ditawarkan maka dibutuhkan media untuk menyebarkan informasi ke orang yang membutuhkan barang tersebut untuk disewa.

Beberapa penelitian yang pernah dilakukan tentang tema persewaan adalah oleh Molloy, R. yaitu pengaruh kendala pasokan perumahan pada keterjangkauan perumahan yang diukur secara langsung menggunakan persewaan yang disesuaikan dengan kualitas serta secara tidak langsung menggunakan ukuran struktur dan lokasi rumah [1]. Penelitian oleh Ni, X. mengukur volatilitas persewaan sumber daya alam: bukti dari EGARCH dan TGARCH untuk data global (alat investigasi secara empiris) [2]. Penelitian oleh Chen, J. meneliti biaya penggunaan rumah dan perbandingan harga sewa di Shanghai [3]. Breidenbach, P, meneliti tentang regulasi persewaan dinamis secara temporal studi kasus kontrol persewaan di pemerintah German [4].

Progressive Web Application adalah pengembangan *website* dengan tujuan banyak media atau dalam istilah multi-platform, yang berarti aplikasi yang dimaksudkan untuk fungsi yang kebanyakan *mobile* dan *desktop* (Android, iOS, Windows, dan lainnya) dan hal ini memungkinkan karena aplikasi diakses melalui *browser* dan *browser* memungkinkan untuk dijalankan dibanyak sistem [5]. Dengan teknologi ini memungkinkan untuk aplikasi yang berjalan ini dilakukan instalasi di *browser* ataupun di media Android dan iOS yang merupakan sistem operasi ponsel pintar. Kemampuan tersebut akan memberikan pengalaman dalam menggunakan aplikasi lebih terbiasa.

Nuxt adalah *framework* yang bersifat kode terbuka dikembangkan dari *framework* Vue.js yang berbasis JavaScript yang membuat sederhana dan *powerfull* [6]. JavaScript yang merupakan basis dari *framework* Nuxt merupakan Bahasa pemrograman yang berjalan diatas *browser*. Bahasa pemrograman ini membuat tampilan lebih interaktif karena memiliki kemampuan 1) dinamis; 2) memiliki fungsi *first-class*; 3) bebas dari *class*, programan berorientasi *object* yang menggunakan pewarisan *prototype*; 4) objek di JS mewarisi properti dari objek lain secara langsung dan semua properti yang diwarisi ini dapat diubah saat *runtime* [7].

Berdasarkan permasalahan yang diungkapkan sebelumnya. Dimana dalam persewaan terdapat masalah pencatatan dan *Progressive Web Application* adalah teknologi yang dinamis untuk digunakan maka tujuan dalam penelitian ini adalah membuat rancangan dan aplikasi persewaan dengan menerapkan teknologi *Progressive Web Application*.

2. METODE PENELITIAN

Metode penelitian diawali dari pengertian atau teknologi untuk merancang atau membuat sistem. Kemudian penjelasan tentang alat, bahan dan langkah-langkah yang dilakukan.

2.1 Kajian Pustaka

Dalam menyusun metodologi penelitian terdapat beberapa materi yang dibahas katitan dengan desain sistem yaitu *System requirement*, *usecase* dan *black box*.

2.1.1 System requirement

Bagian yang paling penting dari proses *engineering* untuk kasus pengembangan sistem yang kompleks adalah *software requirement*. *Stakeholder* proyek biasanya akan mengkomunikasikan kebutuhan formal yang diinginkan dalam bahasa yang baku yang dipilih [8]. Pembahasan oleh *Stakeholder* tersebut akan dibuat *point-point* serta diagram oleh analisis sehingga memudahkan pengembang untuk menyelesaikan pengembangan sistem. Proses analisis akan menghasilkan diagram analisis yang akan disampaikan ke UI/UX dan *programmer* untuk menghasilkan program yang sesuai dengan harapan.

2.1.2 Use case diagram

Pemodelan *use case* adalah Teknik terdepan untuk menspesifikasikan permintaan dalam sistem [9]. Permintaan-permintaan yang didapat dari proses komunikasi dengan *Stakeholder* dibuat dalam diagram yang menggambarkan sistem seharusnya berbuat apa. Konsep utama dari klausa ini adalah *actor*, *usecase* dan *subject*. Setiap *subject* merepresentasikan sistem dibawah pertimbangan *usecase*. *User* dan sistem lain yang berinteraksi dengan dengan *subject* disebut sebagai aktor [10]. Sehingga *Use-case* ini dapat menggambarkan interaksi antara *Actor* dengan sistem yang akan dibuat.

2.1.3 Blackbox

Menurut Gao, Jerry Zeyu dalam bukunya *Testing and Quality Assurance for Component-Based Software* menerangkan mengenai pengertian *black-box* adalah metode pengujian *software* yang memeriksa aplikasi secara fungsional tanpa melihat struktur internal berjalan [11]. Teknis dari metode ini adalah menyesuaikan aplikasi yang telah dibuat dengan permintaan sistem dan memastikan semua permintaan sudah tertangani dan tidak ditemukan kesalahan.

2.2 Metode Penelitian

Metode yang digunakan untuk menjelaskan penelitian ini adalah alat, bahan dan alur penelitian:

A. Alat

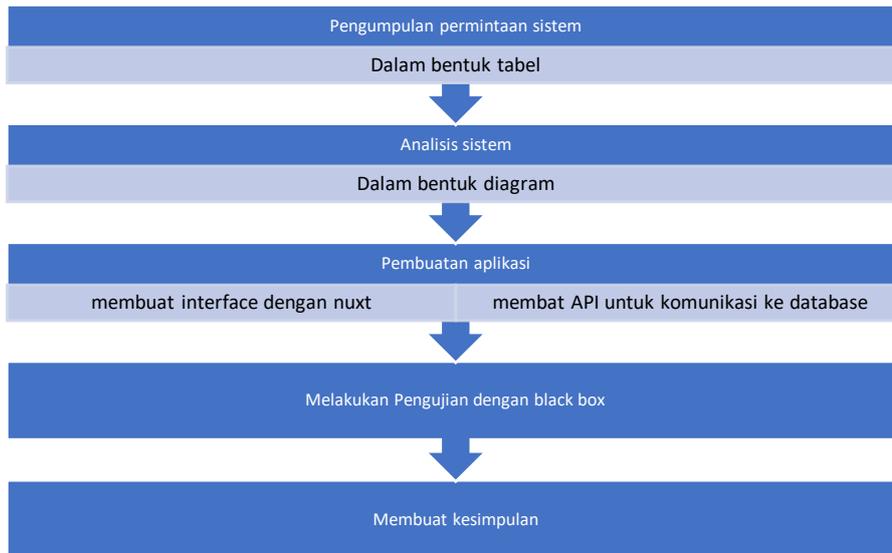
Dalam mengembangkan sistem ini menggunakan bahasa pemrograman JavaScript dengan menggunakan *framework* Vue.js dan Nuxt. *Database* yang digunakan melalui komunikasi API berstandar JSON.

B. Bahan

Bahan untuk mengembangkan sistem adalah data *requirement* dari beberapa pemilik rental dan sistem-sistem rental yang ada di Internet.

C. Alur penelitian

Alur penelitian dapat digambarkan dengan diagram Gambar 1 yang terdiri dari lima Langkah yaitu pengumpulan permintaan sistem, analisis permintaan sistem ke dalam bentuk aplikasi, pembuatan aplikasi, pengujian aplikasinya, menarik kesimpulan dari pengujian.



Gambar 1. Alur penelitian

3. HASIL DAN PEMBAHASAN

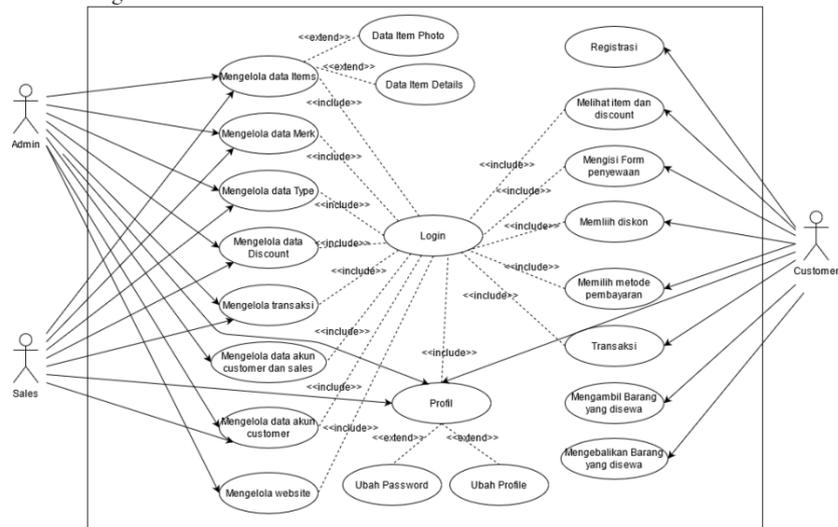
Proses Bisnis dari sistem dimulai dengan Pengguna yang ingin menyewa barang dari sistem harus mendaftarkan diri ke sistem kemudian dari sistem akan memberikan kode OTP ke e-mail yang telah dimasukkan untuk memastikan e-mail yang digunakan adalah valid. Setelah memiliki akun dapat melakukan *login* dengan menggunakan *username* dan *password* yang telah didaftarkan sebelumnya. Proses *login* selesai pengguna dapat memilih barang yang akan dipinjam dan mengkonfirmasi data pinjaman untuk kemudian mengetahui biaya sewa dan membayarnya. Jika pembayaran sudah dilakukan oleh pengguna maka petugas akan mengkonfirmasi pembayaran sehingga setelah dikonfirmasi pembayaran pengguna dapat mengambil barang yang akan dipinjam ke toko. Setelah peminjaman selesai pengguna bisa mengembalikan barang yang telah disewanya dapat dilihat pada Gambar 2.



Gambar 2. Proses bisnis peminjaman barang.

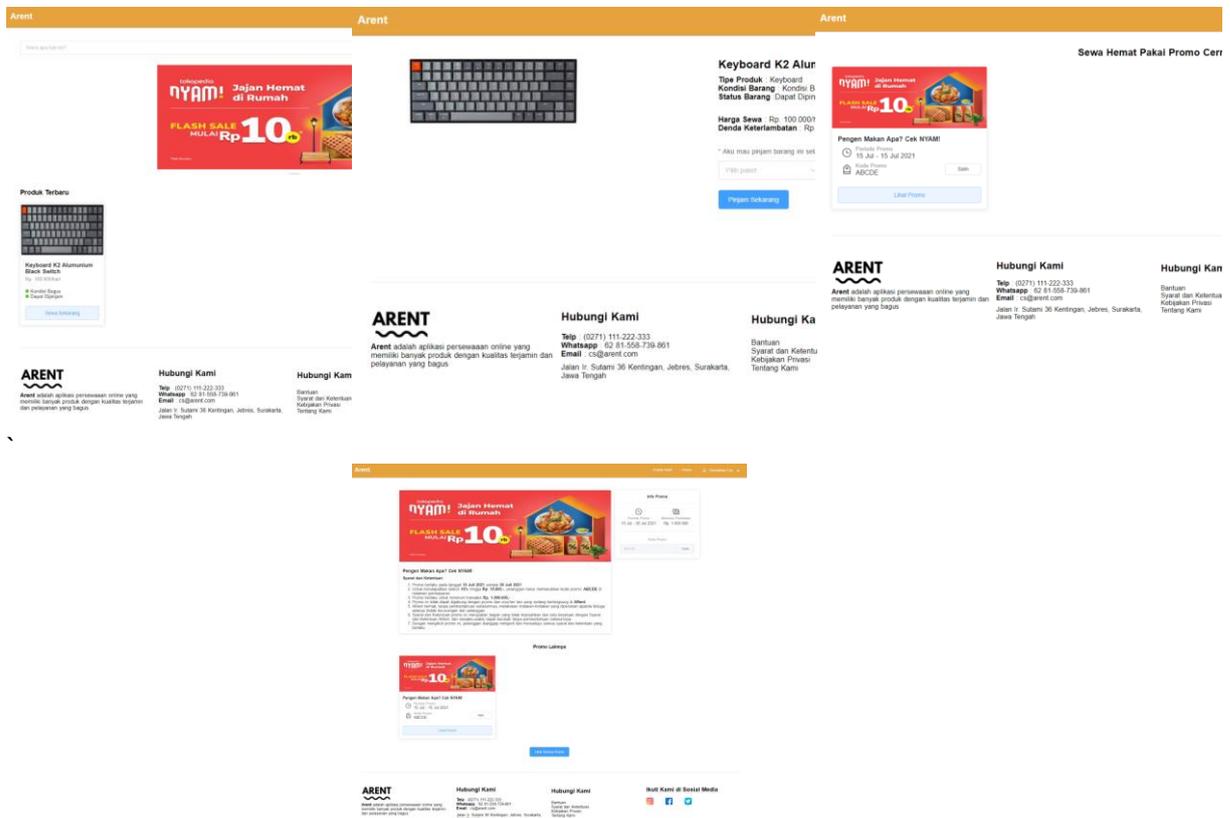
Proses Bisnis yang dibuat diimplementasikan secara detail dalam sistem yang melibatkan empat pengguna yaitu Admin, Sales, dan Customer. Admin dapat mengelola data items, mengelola data merk, mengelola type, mengelola data *discount*, mengelola transaksi, mengelola data akun *customer* dan *sales*, mengelola data akun

customer dan mengelola *website*. *Sales* memiliki hak akses yang sama dengan admin selain mengelola *website*. *Customer* memiliki hak akses registrasi, melihat item dan *discount*, mengisi form penyewaan, memilih *discount*, memilih metode pembayaran, transaksi, mengambil barang yang disewa, mengembalikan barang yang disewa dapat dilihat pada Gambar 3.

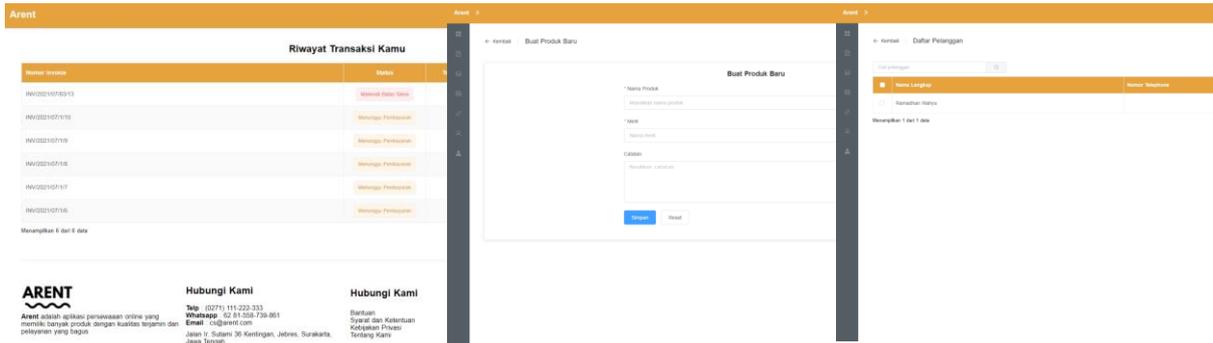


Gambar 3. Usecase Diagram Aplikasi Penyewaan Barang.

Implementasi dalam tampilan program dapat dilihat secara singkat pada gambar empat dan lima. Gambar empat menunjukkan fasilitas inti dari program yaitu dari melihat daftar produk yang disewakan dan promo. Gambar 5 memperlihatkan tampilan riwayat transaksi menampilkan semua riwayat transaksi yang pernah terjadi, input data produk yang akan dijual dan daftar *sales* atau pegawai untuk menentukan *role* tiap pegawai.



Gambar 4. Tampilan pada sisi customer.



Gambar 5. Tampilan pada sisi admin

Implementasi kode salah satunya dapat dilihat pada gambar 6. Gambar tersebut merupakan kode dari tampilan input data produk pada gambar 5. Pada gambar 6 terdapat 4 jenis input yaitu input teks baris tanggal, pilihan, teks dengan banyak baris dan tombol. Input teks baris tanggal yaitu pada kode *el-input*. Pilihan terlihat pada kode *SelectMerk*. Teks dengan banyak baris diwakilkan pada kode *el-input* dengan *type textarea*. Tombol-tombol pada kode *el-button*.

```

<template>
  <el-container>
    <el-header class="page-header">
      <NavigationPageHeader :content="navContent" />
    </el-header>
    <el-main v-if="isInit">
      <el-skeleton :rows="5" />
    </el-main>
    <el-main v-else v-loading="isLoading">
      <el-card>
        <el-container class="w-100">
          <el-row
            type="flex"
            align="middle"
            justify="center"
            style="width: 100%; flex-direction: column"
          >
            <h3 class="fw-bold mb-20">{{ navContent }}</h3>
            <el-col :sm="24" :md="24" :lg="12" :xs="6">
              <el-form
                ref="form"
                :model="form"
                :rules="rules"
                label-width="15%"
                :label-position="labelPosition"
                @submit.prevent.native="submitForm('form')"
              >
                <el-form-item label="Nama Produk" prop="name">
                  <el-input
                    maxlength="100"
                    placeholder="Masukkan nama produk"
                    show-word-limit
                    v-model="form.name"
                  ></el-input>
                </el-form-item>
                <el-form-item label="Merk" prop="merk_id">
                  <SelectMerk
                    v-model="form.merk_id"
                    @change="changeMerk"
                    style="min-width: 100%"
                  />
                </el-form-item>
                <el-form-item label="Catatan" prop="notes">
                  <el-input
                    maxlength="1000"
                    type="textarea"
                    placeholder="Masukkan catatan"
                    show-word-limit
                    rows="5"
                    v-model="form.notes"
                  ></el-input>
                </el-form-item>
                <el-form-item style="margin-top: 10px">
                  <el-button
                    :loading="isLoading"
                    type="primary"
                    native-type="submit"
                  >
                    {{ buttonContent }}
                  </el-button>
                  <el-button @click="resetForm('form')>Reset</el-button>
                </el-form-item>
              </el-form>
            </el-col>
          </el-row>
        </el-container>
      </el-card>
    </el-main>
  </el-container>
</template>

```

Gambar 6. Implementasi kode *form* tampilan input produk

Pengujian dilakukan agar aplikasi yang dibuat sudah sesuai dengan permintaan. Pengujian yang tepat untuk memastikan apakah aplikasi sudah sesuai dengan harapan adalah dengan menggunakan metode *blackbox* dimana pengujian perpermintaan dari sistem dapat dilihat pada tabel 1 terdiri dari kolom nomor, pengujian mendeskripsikan objek yang akan diuji, *scenario* untuk cara kerja pengujian, harapan adalah *output* yang diinginkan dan hasil adalah *output* yang keluar dari program sudah berhasil atau belum. Dari pengujian *blackbox* dari poin nomor satu sampai dengan 51 menunjukkan hasil uji sesuai dengan harapan.

Tabel 1. Pengujian *black box*

No	Pengujian	Skenario	Harapan	Hasil
1.	<i>Home</i>	Mengakses halaman <i>home</i>	Sistem menampilkan halaman <i>home</i>	Berhasil
2.	Menampilkan data daftar produk	Pengguna mengakses halaman daftar produk	Sistem menampilkan data daftar produk	Berhasil
3.	Menampilkan data detail produk	Pengguna mengakses halaman daftar produk → Pelanggan memilih produk	Sistem menampilkan informasi detail produk yang di pilih	Berhasil
4.	Menampilkan data daftar promo	Pengguna mengakses halaman daftar promo	Sistem menampilkan data daftar promo	Berhasil
5.	Menampilkan data detail promo	Pengguna mengakses halaman daftar promo → Pelanggan memilih promo	Sistem menampilkan informasi detail promo yang di pilih	Berhasil
6.	<i>Log In</i>	Mengakses halaman <i>login</i> dan memasukkan <i>username</i> dan <i>password</i>	Sistem masuk ke halaman <i>Dashboard</i> apabila terdaftar sebagai pegawai atau admin.	Berhasil
7.	Melakukan registrasi	Memilih link registrasi → mengisi form pendaftaran → klik tombol daftar	Redirect ke halaman <i>OTP (One Time Password)</i> untuk verifikasi akun	Berhasil
8.	Verifikasi akun pengguna	Admin mengakses halaman <i>OTP (One Time Password)</i> → memasukkan kode <i>OTP</i> → klik tombol verifikasi	Status akun pengguna menjadi aktif → <i>redirect</i> ke halaman <i>login</i>	Berhasil
9.	Mengubah profil	Pengguna mengakses halaman profil → mengisi biodata yang akan diubah → klik simpan	Menampilkan profil pengguna → menyimpan profil pengguna yang baru	Berhasil
10.	Mengubah <i>password</i>	Pengguna mengakses halaman ubah <i>password</i> → mengisikan <i>password</i> lama, <i>password</i> baru dan konfirmasi <i>password</i> baru → klik simpan	Menampilkan tampilan formulir ubah <i>password</i> → Menyimpan <i>password</i> baru	Berhasil
11.	Menampilkan daftar riwayat transaksi	Pengguna mengakses halaman daftar riwayat	Menampilkan daftar riwayat transaksi	Berhasil

12.	Menampilkan halaman <i>dashboard</i>	Admin dan petugas mengakses halaman <i>dashboard</i>	Menampilkan halaman <i>dashboard</i>	Berhasil
13.	Menampilkan halaman daftar transaksi	Admin dan petugas mengakses halaman daftar transaksi	Menampilkan halaman daftar transaksi	Berhasil
14.	Menerima peminjaman	Admin dan petugas mengakses halaman daftar transaksi → memilih transaksi yang akan diterima → klik tombol terima	Status transaksi berubah menjadi diterima.	Berhasil
15.	Menolak peminjaman	Admin dan petugas mengakses halaman daftar transaksi → memilih transaksi yang akan ditolak → klik tombol tolak	Status transaksi berubah menjadi ditolak.	Berhasil
16.	Memproses peminjaman	Admin dan petugas mengakses halaman daftar transaksi → memilih transaksi yang akan diproses → klik tombol proses	Status transaksi berubah menjadi proses.	Berhasil
17.	Menyelesaikan peminjaman	Admin dan petugas mengakses halaman daftar transaksi → memilih transaksi yang akan diselesaikan → klik tombol selesai	Status transaksi berubah menjadi selesai.	Berhasil
18.	Menampilkan halaman produk	Admin dan petugas mengakses halaman daftar produk	Menampilkan data produk	Berhasil
19.	Menambah produk	Admin dan petugas mengakses halaman daftar produk → klik tombol tambah → mengisi <i>form</i> tambah produk	Menampilkan halaman <i>form</i> tambah produk → menyimpan produk	Berhasil
20.	Mengubah produk	Admin dan petugas mengakses halaman daftar produk → memilih produk yang akan diubah → mengubah produk	Menampilkan halaman <i>form</i> ubah produk → menyimpan produk	Berhasil
21.	Menghapus produk	Admin dan petugas mengakses halaman daftar produk → memilih produk yang akan dihapus → klik tombol hapus	Menampilkan pesan konfirmasi penghapusan produk → menghapus produk	Berhasil
22.	Menampilkan halaman produk turunan	Admin dan petugas mengakses halaman daftar produk turunan	Menampilkan data produk turunan	Berhasil

23.	Menambah produk turunan	Admin dan petugas mengakses halaman daftar produk turunan → klik tombol tambah → mengisi form tambah produk turunan	Menampilkan halaman <i>form</i> tambah produk turunan → menyimpan produk turunan	Berhasil
24.	Mengubah produk turunan	Admin dan petugas mengakses halaman daftar produk turunan → memilih produk turunan yang akan diubah → mengubah produk turunan	Menampilkan halaman <i>form</i> ubah produk turunan → menyimpan produk turunan	Berhasil
25.	Menghapus produk turunan	Admin dan petugas mengakses halaman daftar produk turunan → memilih produk turunan yang akan dihapus → klik tombol hapus	Menampilkan pesan konfirmasi penghapusan produk turunan → menghapus produk turunan	Berhasil
26.	Menampilkan halaman promo	Admin dan petugas mengakses halaman daftar promo	Menampilkan data promo	Berhasil
27.	Menambah promo	Admin dan petugas mengakses halaman daftar promo → klik tombol tambah → mengisi form tambah promo	Menampilkan halaman <i>form</i> tambah promo → menyimpan promo	Berhasil
28.	Mengubah promo	Admin dan petugas mengakses halaman daftar promo → memilih promo yang akan diubah → mengubah promo	Menampilkan halaman <i>form</i> ubah promo → menyimpan promo	Berhasil
29.	Menghapus promo	Admin dan petugas mengakses halaman daftar promo → memilih promo yang akan dihapus → klik tombol hapus	Menampilkan pesan konfirmasi penghapusan promo → menghapus promo	Berhasil
30.	Menampilkan halaman referensi merk	Admin dan petugas mengakses halaman referensi merk	Menampilkan data referensi merk	Berhasil
31.	Menambah referensi merk	Admin dan petugas mengakses halaman daftar referensi merk → klik tombol tambah → mengisi form tambah referensi merk	Menampilkan halaman <i>form</i> tambah referensi merk → menyimpan referensi merk	Berhasil
32.	Mengubah referensi merk	Admin dan petugas mengakses halaman daftar referensi merk → memilih referensi merk yang akan diubah →	Menampilkan halaman <i>form</i> ubah referensi merk → menyimpan referensi merk	Berhasil

		mengubah referensi merk		
33.	Menghapus referensi merk	Admin dan petugas mengakses halaman daftar referensi merk → memilih referensi merk yang akan dihapus → klik tombol hapus	Menampilkan pesan konfirmasi penghapusan referensi merk → menghapus referensi merk	Berhasil
34.	Menampilkan halaman referensi tipe produk	Admin dan petugas mengakses halaman referensi tipe produk	Menampilkan data referensi tipe produk	Berhasil
35.	Menambah referensi tipe produk	Admin dan petugas mengakses halaman daftar referensi tipe produk → klik tombol tambah → mengisi <i>form</i> tambah referensi tipe produk	Menampilkan halaman <i>form</i> tambah referensi tipe produk → menyimpan referensi tipe produk	Berhasil
36.	Mengubah referensi tipe produk	Admin dan petugas mengakses halaman daftar referensi tipe produk → memilih referensi tipe produk yang akan diubah → mengubah referensi tipe produk	Menampilkan halaman <i>form</i> ubah referensi tipe produk → menyimpan referensi tipe produk	Berhasil
37.	Menghapus referensi tipe produk	Admin dan petugas mengakses halaman daftar referensi tipe produk → memilih referensi tipe produk yang akan dihapus → klik tombol hapus	Menampilkan pesan konfirmasi penghapusan referensi tipe produk → menghapus referensi tipe produk	Berhasil
38.	Menampilkan halaman daftar pelanggan	Admin dan petugas mengakses halaman daftar pelanggan	Menampilkan data daftar pelanggan	Berhasil
39.	Menampilkan halaman pegawai	Admin mengakses halaman pegawai	Menampilkan data pegawai	Berhasil
40.	Menambah pegawai	Admin mengakses halaman daftar pegawai → klik tombol tambah → mengisi <i>form</i> tambah pegawai	Menampilkan halaman <i>form</i> tambah pegawai → menyimpan pegawai	Berhasil
41.	Mengubah pegawai	Admin mengakses halaman daftar pegawai → memilih pegawai yang akan diubah → mengubah pegawai	Menampilkan halaman <i>form</i> ubah pegawai → menyimpan pegawai	Berhasil
42.	Mengaktifkan pegawai	Admin mengakses halaman daftar pegawai → memilih pegawai yang akan diaktifkan → klik tombol aktifkan	Menampilkan pesan konfirmasi pengaktifan pegawai → mengubah status pegawai menjadi aktif	Berhasil

43.	Menonaktifkan pegawai	Admin mengakses halaman daftar pegawai → memilih pegawai yang akan di non aktifkan → klik tombol non aktifkan	Menampilkan pesan konfirmasi penonaktifan pegawai → mengubah status pegawai menjadi non aktif	Berhasil
44.	Menampilkan halaman daftar menu	Admin mengakses halaman daftar menu	Menampilkan data menu	Berhasil
45.	Menambah menu	Admin mengakses halaman daftar menu → klik tombol tambah → mengisi <i>form</i> tambah menu	Menampilkan halaman <i>form</i> tambah menu → menyimpan menu	Berhasil
46.	Mengubah menu	Admin mengakses halaman daftar menu → memilih menu yang akan diubah → mengubah menu	Menampilkan halaman <i>form</i> ubah menu → menyimpan menu	Berhasil
47.	Menghapus menu	Admin mengakses halaman daftar menu → memilih menu yang akan dihapus → klik tombol hapus	Menampilkan pesan konfirmasi penghapusan menu → menghapus menu	Berhasil
48.	Menampilkan halaman daftar hak akses	Admin mengakses halaman daftar hak akses	Menampilkan data hak akses	Berhasil
49.	Menambah hak akses	Admin mengakses halaman daftar hak akses → klik tombol tambah → mengisi <i>form</i> tambah hak akses	Menampilkan halaman <i>form</i> tambah hak akses → menyimpan hak akses	Berhasil
50.	Mengubah hak akses	Admin mengakses halaman daftar hak akses → memilih hak akses yang akan diubah → mengubah hak akses	Menampilkan halaman <i>form</i> ubah hak akses → menyimpan hak akses	Berhasil
51.	Menghapus menu	Admin mengakses halaman daftar hak akses → memilih hak akses yang akan dihapus → klik tombol hapus	Menampilkan pesan konfirmasi penghapusan hak akses → menghapus hak akses	Berhasil

4. KESIMPULAN

Aplikasi persewaan dengan teknologi *Progressive Web Application* telah berhasil dibuat. Teknologi PWA ini memungkinkan untuk berjalan baik di komputer maupun di *handphone*. Selain itu dapat melakukan instalasi dalam komputer ataupun di *handphone* tapi harus tetap terkoneksi dengan internet untuk menjalankannya. Proses urutan yang berjalan sudah tergambar pada diagram Proses Bisnis sedangkan hubungan entitas yang terlibat dengan sistem dapat dilihat pada *use case*. Out program sudah sesuai dengan harapan serta semua fasilitas program sesuai dengan permintaan berdasarkan pengujian dengan menggunakan *Black-Box*.

DAFTAR PUSTAKA

- [1] R. Molloy, C. G. Nathanson, and A. Paciorek, "Housing Supply and Affordability: Evidence from Rents, Housing Consumption and Household Location," *Journal of Urban Economics*, p. 103427, Feb. 2022, doi:

- 10.1016/j.jue.2022.103427.
- [2] X. Ni, Z. Wang, A. Akbar, and S. Ali, "Measuring natural resources rents volatility: Evidence from EGARCH and TGARCH for global data," *Resources Policy*, vol. 76, p. 102553, Jun. 2022, doi: 10.1016/j.resourpol.2022.102553.
 - [3] J. Chen, Y. Chen, R. J. Hill, and P. Hu, "The user cost of housing and the price-rent ratio in Shanghai," *Regional Science and Urban Economics*, vol. 92, p. 103738, Jan. 2022, doi: 10.1016/j.regsciurbeco.2021.103738.
 - [4] P. Breidenbach, L. Eilers, and J. Fries, "Temporal dynamics of rent regulations – The case of the German rent control," *Regional Science and Urban Economics*, vol. 92, p. 103737, Jan. 2022, doi: 10.1016/j.regsciurbeco.2021.103737.
 - [5] A. Biørn-Hansen, T. A. Majchrzak, and T.-M. Grønli, "Progressive Web Apps: The Possible Web-native Unifier for Mobile Development," in *Proceedings of the 13th International Conference on Web Information Systems and Technologies*, 2017, pp. 344–351. doi: 10.5220/0006353703440351.
 - [6] nuxtjs, "The Intuitive Vue Framework." <https://nuxtjs.org/> (accessed Feb. 10, 2022).
 - [7] A. M. Fard and A. Mesbah, "JSNOSE: Detecting JavaScript Code Smells," in *2013 IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, Sep. 2013, pp. 116–125. doi: 10.1109/SCAM.2013.6648192.
 - [8] M. R. R. Ramesh and Ch. S. Reddy, "Metrics for software requirements specification quality quantification," *Computers & Electrical Engineering*, vol. 96, p. 107445, Dec. 2021, doi: 10.1016/j.compeleceng.2021.107445.
 - [9] M. El-Attar, "Evaluating and empirically improving the visual syntax of use case diagrams," *Journal of Systems and Software*, vol. 156, pp. 136–163, Oct. 2019, doi: 10.1016/j.jss.2019.06.096.
 - [10] Object Management Group, OMG® Unified Modeling Language® (OMG UML®). Milford: Object Management Group, 2017.
 - [11] J. Z. Gao, H.-S. J. Tsao, and Y. Wu, "Testing and Quality Assurance for Component-Based Software."