

BAB V

SIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka penulis menyimpulkan bahwa:

1. Sistem informasi ini dapat memudahkan pelanggan dalam mencari kost-an kapan pun dan dimana cukup lewat *website* ini saja.
2. Menghasilkan informasi yang lengkap dan akurat mengenai data kost-an yang akan dipilih.
3. Memberikan manfaat kepada pemilik rumah kost-an karena lebih dikenal di masyarakat luas melalui *internet*.

5.2 Saran

Dalam mengimplementasikan *website* ini masih banyak terdapat kekurangan. berikut hal – hal yang perlu dikembangkan kedepannya:

1. Meningkatkan tampilan yang *user friendly* dan kekinian.
2. Memperbanyak aneka pilihan pembayaran.
3. Perlu menambahkan fitur *maps* akurat dan terupdate untuk memudahkan pelanggan mengetahui lokasi kost yang akan dipilih.

DAFTAR PUSTAKA

- Abdul Kadir (2013). Pengertian *MySQL*. Tersedia dalam : Buku Pintar Programmer Pemula PHP. Yogyakarta. Mediakom.
- Ambarwati, Desti. 2020. "NASKAH_PUBLIKASI Bab1." 2013 1–12.
- Al-Bahra Bin Ladjamudin. 2013. Analisis dan Desain Sistem Informasi. Graha Ilmu.
- Andi, 2013. Adobe Dreamweaver CS6. Yogyakarta: Elcom.
- Bin Tahir, Tamus, Muh. Rais, and Moch. Apriyadi HS. 2019. "Aplikasi Point OF Sales Menggunakan Framework Laravel." *JIKO (Jurnal Informatika Dan Komputer)* 2(2):55–59. doi: 10.33387/jiko.v2i2.1313.
- Bambang, Riyanto. 2012. Dasar-dasar Pembelanjaan, Edisi 4, Yogyakarta: BPFE.
- Destiana Yusma, Nita Merlina, Nurajijah. 2021. "Sistem Informasi Pencarian Rumah Kost Berbasis WEB." *Jurnal Inti Nusa Mandiri* 15(2):9–16.
- Fitriani, Kiki, and Agung Purwanto. 2019. "Sistem Informasi Rumah Kost Di Baamang Berbasis Web." *Jurnal Penelitian Dosen Fikom (UNDA)* 10(1).
- Gelinas, J. U. , Dull, Richard B. , Wheeler, Patrick R. (2012). Accounting Information
- Hutahaean, J. (2014), Konsep Sistem Informasi, Yogyakarta: Deepublish. [5] Ladjamudin, A. B. Bi
- Kadir, Abdul. 2014, Pengenalan Sistem Informasi Edisi Revisi. Yogyakarta : Andi.
- Kustiyangsih, Yeni dan Devie Rosa Anamisa. 2011, Pemrograman Basis Data Berbasis Web Menggunakan PHP & *MySQL*. Jakarta : Graha Ilmu.
- Kustiyangsih, Yeni. 2011, Pemrograman Basis Data Berbasis Web Menggunakan PHP & *MySQL*. Jakarta : Graha Ilmu.
- Longkutoy, J. John (2012). Pengenalan Komputer. PT. Prehalindo, Jakarta.
- Luthfi, Farizan. 2017. "Penggunaan Framework Laravel Dalam Rancang Bangun Modul Back-End Artikel Website Bisnisbisnis.ID." *JISKA (Jurnal Informatika Sunan Kalijaga)* 2(1):34–41. doi: 10.14421/jiska.2017.21-05.
- Maulida Solihat, Nurul, Program Studi Teknik Informatika, Uts Sumbawa Jln Raya Olat Maras, Batu Alang, Moyo Hulu, Kabupaten Sumbawa, and Nusa Tenggara Barat. 2021. "Rekayasa Aplikasi Center Rumah Kost Berbasis Web Di Kabupaten Sumbawa." *Jurnal Manajemen Informatika & Sistem Informasi* 4(2):141–48.
- Maniah dan Hamidini, D. (2017), Analisis dan Perancangan Sistem Informasi Pembahasan Secara Praktis dengan Contoh Kasus, Yogyakarta: Deepublish.
- Murad. Dina Fitria, Kusniawati. Nia, Asyanto. Agus. 2013. Aplikasi IntelligenceWebsite

- Nugroho, Adi Sulistyono. 2017, Analisis dan Perancangan Sistem Informasi. Yogyakarta : Trans Media.
- Oktavian, Diar Puji. 2010, Menjadi Programmer Jempolan Menggunakan PHP. Yogyakarta : Mediakom.
- Palevi, A. R. & Krisnawati, 2013. Analisa dan Perancangan Sistem Informasi Penerimaan Peserta Didik Baru Berbasis Website Pada Smp N 2 Mojosoong Boyolali. Jurnal Ilmiah Dasi, 14(ISSN : 1411- 3201), pp. 1-6.
- Pratama, I Putu Agus Eka. 2014, Sistem Informasi Dan Implementasinya. Bandung : Informatika Bandung.
- Purnia, Dini Silvi, Ratningsih Ratningsih, Mumun Surahman, and Widhiani Agustin. 2021. "Implementasi Metode Prototyping Pada Rancang Marketplace Rumah Kost Berbasis Mobile." *EVOLUSI : Jurnal Sains Dan Manajemen* 9(1):1–11. doi: 10.31294/evolusi.v9i1.10145.
- Ramdan, D. S. 2020. "Aplikasi Desktop Multi Platform Untuk Redis Client Berbasis Teknologi Web Menggunakan Framework ElectronJS Dan ReactJS." *Jurnal TEDC* 14(3):226–31.
- Rob, Peter dan Coronel, Carlos (2009). Database system. Salemba Empat. Jakarta.
- Rosa, A. , & Shalahuddin, M. 2014, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung : Informatika.
- Rudiyanto, A. , 2011. Pemrograman Web Dinamis Menggunakan PHP dan *MySQL*. Yogyakarta: Andi.
- Satzinger, Ja ckson, Burd. (2010). System Analisis and Design with the Unified Process.
- Sugiarti, Yuni. 2013, Analisis dan Perancangan UML (Unified Modelling Language) Generate VB 6. Bandung : Informatika.
- Sutarman. 2012. Buku Pengantar Sistem Informasi. Yogyakarta : Graha Ilmu.
- Systems. South Western : Cengage Learning.
- Tangerang: Perguruan Tinggi Raharja. Vol. 7, No. 1, September 2013.
- Studi, Program, Sistem Informasi, Fakultas Sains, and U. I. N. Suska Riau. 2020. "Rancang Bangun Aplikasi Pencarian Rute Kos-Kosan." 6(1):16–25.
- Susilo, Muhammad. 2018. "Rancang Bangun Website Toko Online Menggunakan Metode Waterfall." *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)* 2(2):98–105. doi: 10.30743/infotekjar.v2i2.171.
- Wahid, Aceng Abdul. 2020. "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi." *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK* (November):1–5.

Yakub. (2012). Data dan Deskripsi. Graha ilmu. Yogyakarta.

Yogyakarta.

Yuhfizar (2013), Mudah Membangun Web Profil Multibahasa, Jakarta: Elex Media
Komputindo.

Zeck Gian T.F. 2019. “Rancang Bangun Sistem Informasi Kos-Kosan Menggunakan
Framework Rapid Application Development.” Pp. 226–33 in Vol. 3.



DAFTAR RIWAYAT HIDUP



Data Pribadi

Nama Lengkap : Stefusus Kristianto
Tempat, Tanggal Lahir : Belitung, tanjung pandan, 22 September 2000
Jenis Kelamin : Laki-laki
Alamat : Jl. Iman Bonjol, Gg. bansin 7A, Karawaci, Sukajadi,
Tangerang
Agama : Protestan
No Kontak : 087748588585
eMail : stefunusk@gmail.com

Pendidikan Formal

2018 - Sekarang : Strata Satu Universitas Buddhi Dharma, Tangerang
2015 - 2018 : Sekolah Menengah Kejuruan Yaperbel 2 Tanjung
Pandan, Belitung
2012 - 2015 : Sekolah Menengah Pertama Pgri 2 Tanjung pandan,
Belitung
2007- 2012 : Sekolah Dasar Negeri 5 Tanjung pandan Belitung

Pengalaman Kerja

2021 - Sekarang : Pratama Tax Consultant (Accounting Staff)

Tangerang, 04 Agustus 2022

A handwritten signature in black ink, appearing to read 'Stefusus', written in a cursive style.

Stefusus Kristianto

No.	Analisis Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat :	
	Menarik	✓
	kelamian	✓
	update	✓
	mudah	✓
	bisa login	✓
	bisa register	✓
	bisa mencari	✓
	bisa menambahkan	✓
	bisa menghapus	✓
	bisa transaksi	✓

Tangerang, 04 Maret 2022

Pembimbing,



Riki, M. Kom

NIDN. 0431128204

Responden,



Priscillia .A.A

Mahasiswa,



Stefunus Kristianto
NIM : 20180700040

No.	Analisis Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat :	
	menampilkan isi least	✓
	Bisa login	✓
	Bisa register	✓
	Bisa dual alan	✓
	Bisa menambahkan produk	✓
	Bisa delete	✓
	user friendly	✓
	ubah	✓
	online	✓
	Bisa Transaksi	✓

Tangerang, 04 Maret 2022

Pembimbing,



Riki. M. Kom
NIDN. 0431128204

Responden,



Uesiana

Mahasiswa,



Stefunus Kristianto
NIM : 20180700040

No.	Analisis Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat :	
	Bisa Menampilkan halaman Awal	✓
	Bisa Menampilkan Menu Login	✓
	Bisa menampilkan Menu Daftar	✓
	Bisa Dual Akun dalam Satu website	✓
	Bisa menambahkan data Produk	✓
	Bisa Menghapus Produk	✓
	Bisa Mencari produk	✓
	Bisa layout	✓
	Tampilan Admin	x
	online	

Tangerang, 04 Maret 2022

Pembimbing,



Riki M. Kom
NIDN. 0431128204

Responden,


Rohman

Mahasiswa,



Stefunus Kristianto
NIM : 20180700040

No.	Analisis Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat :	
	Aman	✓
	Mudah	✓
	Simple	✓
	Bisa login	✓
	Bisa daftar	✓
	Bisa multi akun	✓
	Bisa transaksi	✓
	Bisa menampilkan invoice	✓
	Bisa logout	✓
	Bisa Delete	

Tangerang, 04 Maret 2022

Pembimbing,



Riki, M Kom
NIDN. 0431128204

Responden,



Helmi

Mahasiswa,



Stefusus Kristianto
NIM : 20180700040

No.	Analisis Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat :	
1.	men cari kost-an	✓
2.	Bisa login	✓
3.	Transaksi	✓
4.	menampilkan invoice	✓
5.	menjelaskan detail kost-an	✓
6.	Bisa menambahkan kost-an	✓
7.	mendaftarkan diri / registrasi	✓
8.	online	x
9.	Tampilan menarik	✓
10.	kelunian	✓

Tangerang, 04 Maret 2022

Pembimbing,



Riki, M Kom
NIDN. 0431128204

Responden,



Herman

Mahasiswa,



Stefunus Kristianto
NIM : 20180700040

No.	Analisis Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat :	
	Bisa menampilkan halaman awal	✓
	Bisa menampilkan halaman login	✓
	Bisa menampilkan halaman Register	✓
	Bisa menampilkan halaman pencatatan	✓
	Bisa menampilkan checkout	✓
	Bisa menampilkan invoice	✓
	Bisa logout	✓
	Bisa menambahkan produk	✓
	Bisa menghapus produk	✓
	Tampilan simple	✓

Pembimbing,



Riki. M. Kom
NIDN. 0431128204

Responden,



Dessy Mutiara

Tangerang, 03 Maret 2022
Mahasiswa,



Stefunus Kristianto
NIM : 20180700040

No.	Analisis Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat :	
	memiliki 2 akses akun	✓
	Bagian admin dapat melihat data pelanggan	✓
	admin bisa mengedit data produk	✓
	Bisa login	✓
	Bisa register	✓
	Tampilan friendly	✓
	Bisa logout	✓
	Pelanggan bisa melihat pesannya	✓
	menampilkan Daftar Koston	✓
	Jenis pembayaran	✓

Tangerang, 04 Maret 2022

Pembimbing,



Riki, M. Kom

NIDN. 0451128204

Responden,



Jonathan,

Mahasiswa,



Stefusus Kristianto
NIM : 20180700040

No.	Analisis Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat :	
1.	User Friendly	✓
2.	Terupdate	✓
3.	Bisa login	✓
4.	Bisa register	✓
5.	Bisa transaksi	✓
6.	Cepat	✓
7.	Menampilkan daftar kost-an	✓
8.	Online	✗
9.	Notif kalau ada pesanan	✓
10.	Jenis pembayaran mudah dan banyak	✓

Tangerang, 04 Maret 2022

Pembimbing,



Riki, M. Kom

NIDN. 0431128204

Responden,



EMMA

Mahasiswa,



Stefusus Kristianto

NIM : 20180700040

No.	Analisis Kebutuhan Sistem	
	Saya ingin sistem dapat :	Keterangan
	Bisa login	✓
	Bisa register	✓
	menambahkan user	✓
	Delete user	✓
	mencari user	✓
	logout	✓
	Bisa Transaksi	✓
	menampilkan Invoice	✓
	menghapus	✓
	lebih efisien dan efektif	✓

Tangerang, 04 Maret 2022

Pembimbing,

Responden,

Mahasiswa,



Riki, M. Kom
NIDN. 0431128204



Agus Saputra



Stefunus Kristianto
NIM : 20180700040

No.	Analisis Kebutuhan Sistem	Keterangan
	Saya Ingin sistem dapat :	
1	MUDAH	Bisa
2	TAMPILAN MENARIK	Bisa
3	BISA LOGIN	Bisa
4	BISA REGISTER	Bisa
5	WARNA BERAGAM	Bisa
6	MENAMPILKAN ISI KOST	Bisa
7	HARGA PROMO	Tidak
8	BANYAK CASHBACK	Tidak
9	BISA MENAMBAHKAN KOST SENDIRI	Bisa
10	TERPERCAYA	Bisa

Tangerang, 04 Maret 2022

Pembimbing,



Riki, M. Kom
NIDN. 0431128204

Responden,



Harto Mandra

Mahasiswa,



Stefunus Kristianto
NIM : 20180700040

LAMPIRAN CODE PROGRAM WEBSITE

Auth.tsx

```
import { AxiosError, AxiosResponse } from "axios";
import { useState } from "react";
import { useMutation } from "react-query";
import { useNavigate } from "react-router-dom";
import { toast } from "react-toastify";
import { Link } from "../components/Links";
import { useStore } from "../store/store";
import { ILoginParam, login, setAuthToken } from "../utils/authenticate";

export interface ILaravelApiErrorReturn {
  message?: string;
  errors?: {
    email?: Array<string>;
    password?: Array<string>;
  };
}

const Auth = () => {
  const [form, setForm] = useState<ILoginParam>({
    email: "",
    password: "",
  });
  const handleChange = (
    label: string,
    e: React.ChangeEvent<HTMLInputElement>
  ) => {
    setForm({ ...form, [label]: e.target.value });
  };

  const [formError, setFormError] = useState<ILaravelApiErrorReturn>({});

  const setUserProfile = useStore((state) => state.setUserProfile);

  const navigate = useNavigate();
  const { isLoading: isPosting, mutate: loginPost } = useMutation(
    async () => {
      return login(form);
    },
    {
      onSuccess: (res: AxiosResponse) => {
        const result = {
          status: res.status + "-" + res.statusText,
          headers: res.headers,
          data: res.data,
        };
        setUserProfile({
          email: res.data.content.user_email,
          name: res.data.content.user_name,

```

```

    role: res.data.content.role,
    join: new Date(res.data.content.user_join),
  });
  setAuthToken(res.data.content.access_token);
  setFormError({ });
  if (res.data.content.role === 1) {
    navigate("/admin/product");
  } else {
    navigate("/");
  }
},
onError: (err: AxiosError) => {
  const errors = err.response?.data as ILaravelApiErrorReturn;
  setFormError(errors);
  toast("Failed login " + err.message, {
    position: "top-right",
    autoClose: 5000,
    hideProgressBar: false,
    closeOnClick: true,
    pauseOnHover: true,
    draggable: true,
    progress: undefined,
  });
},
}
);

const onSubmit = (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault();
  loginPost();
};
return (
  <div className="flex items-center justify-center min-h-full px-4 py-12 sm:px-6 lg:px-8">
    <div className="w-full max-w-md space-y-8">
      <div>
        <div className="w-auto h-12 mx-auto text-5xl font-extrabold text-center">
          <h1>Kost Gilak</h1>
        </div>
        <h2 className="mt-6 text-3xl font-extrabold text-center text-gray-900">
          Sign in to your account
        </h2>
      </div>
      <form
        className="mt-8 space-y-6"
        onSubmit={(event) => onSubmit(event)}
        method="POST"
      >
        <input type="hidden" name="remember" value="true" />

```

```

<div className="-space-y-px rounded-md shadow-sm">
  <div>
    <label className="sr-only">Email address</label>
    <input
      id="email-address"
      name="email"
      type="email"
      autoComplete="email"
      required
      className="relative block w-full px-3 py-2 text-gray-900 placeholder-
gray-500 border border-gray-300 rounded-none appearance-none rounded-t-md
focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 focus:z-10
sm:text-sm"
      placeholder="Email address"
      onChange={(event) => handleChange("email", event)}
    />
    { formError.errors?.email?.map((item: string, i: number) => (
      <div key={i} className="text-sm text-center text-red-600">
        {item}
      </div>
    ))}
  </div>
  <div>
    <label htmlFor="password" className="sr-only">
      Password
    </label>
    <input
      id="password"
      name="password"
      type="password"
      autoComplete="current-password"
      required
      className="relative block w-full px-3 py-2 text-gray-900 placeholder-
gray-500 border border-gray-300 rounded-none appearance-none rounded-b-md
focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 focus:z-10
sm:text-sm"
      placeholder="Password"
      onChange={(event) => handleChange("password", event)}
    />
    { formError.errors?.password?.map((item: string, i: number) => (
      <div key={i} className="text-sm text-center text-red-600">
        {item}
      </div>
    ))}
  </div>
</div>

<div className="flex items-center justify-between">
  <div className="flex items-center">
    <input

```

```

        id="remember-me"
        name="remember-me"
        type="checkbox"
        className="w-4 h-4 text-indigo-600 border-gray-300 rounded
focus:ring-indigo-500"
    />
    <label className="block ml-2 text-sm text-gray-900">
        Remember me
    </label>
</div>

<div className="text-sm">
    <a
        href="#"
        className="font-medium text-indigo-600 hover:text-indigo-500"
    >
        Forgot your password?
    </a>
</div>
</div>

<div>
    <button
        type="submit"
        className="relative flex justify-center w-full px-4 py-2 text-sm font-
medium text-white bg-indigo-600 border border-transparent rounded-md group
hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-offset-2
focus:ring-indigo-500"
    >
        <span className="absolute inset-y-0 left-0 flex items-center pl-3">
            <svg
                className="w-5 h-5 text-indigo-500 group-hover:text-indigo-400"
                xmlns="http://www.w3.org/2000/svg"
                viewBox="0 0 20 20"
                fill="currentColor"
                aria-hidden="true"
            >
                <path
                    fillRule="evenodd"
                    d="M5 9V7a5 5 0 0110 0v2a2 2 0 012 2v5a2 2 0 01-2 2H5a2 2 0 01-
2-2v-5a2 2 0 012-2zm8-2v2H7V7a3 3 0 016 0z"
                    clipRule="evenodd"
                />
            </svg>
        </span>
        Sign in
    </button>
    <div className="mt-2 text-sm text-center">
        <span className="mr-2"> Don't have a account?</span>
        <Link

```

```

        href="/register"
        className="font-medium text-indigo-600 hover:text-indigo-500"
      >
        Create new account
      </Link>
    </div>
  </div>
</form>
</div>
</div>
);
};

```

export default Auth;

Detail.tsx

```

import { useQuery } from "react-query";
import { useParams } from "react-router-dom";
import Card from "../components/custom/Card";
import { Link } from "../components/Links";
import { Header } from "../components/Navbar/Header";
import { getDetailProduct, getListShop } from "../utils/data";
import { formatNumber } from "../utils/helper";

const Detail = () => {
  const { id } = useParams();
  const { data, error, isError, isLoading, isFetching, refetch } = useQuery(
    "detailProduct",
    () => getDetailProduct(id),
    { keepPreviousData: true, enabled: true }
  );

  const { data: dataProduct } = useQuery(["product"], () => getListShop(""));

  return (
    <div className="h-screen">
      <Header></Header>
      <div className="flex flex-col w-full h-full px-10 mt-4">
        <div className="font-bold">Product Detail</div>
        <div className="flex justify-center">
          <div className="flex flex-col w-6/12 py-8">
            <div className="flex justify-center w-full h-full">
              <div className="w-6/12 h-full border">
                <img
                  src={data?.data.photo}
                  alt="image"
                  className="w-full h-full"
                />
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

```



```

<div className="w-6/12 h-full px-3 py-5 border">
  <div className="text-xl bold">{data?.data.name}</div>
  <div>
    Rp.{data?.data.price ? formatNumber(data?.data.price) : 0}
  </div>
  <div>{data?.data.location}</div>
  <Link
    href={`\checkout/${id}`}
    className="flex items-center justify-center px-2 py-1 mt-10 mb-4
text-sm text-white bg-black border border-gray-700 cursor-pointer whitespace-
nowrap hover:bg-gray-500 focus:outline-none focus:bg-gray-500"
  >
    Make Appointment
  </Link>
</div>
</div>
<div className="px-2 py-2 w-full min-h-[5rem] h-full border overflow-
auto">
  <div className="font-bold">Description</div>
  <span className="whitespace-pre">{data?.data.description}</span>
</div>
</div>
<div className="font-bold">Recommendation</div>
<div className="flex justify-center w-full">
  <div className="grid grid-cols-6 gap-4 p-3 mt-1 h-80">
    {dataProduct?.data.map((row: any, index: number) => (
      <Card
        id={row.id}
        src={row.photo}
        title={row.name}
        price={row.price}
        description={row.location}
        detail={row.description}
      ></Card>
    ))}
  </div>
</div>
</div>
);
};

```

export default Detail;

Checkout.tsx

```

import { AxiosError } from "axios";
import React, { useState } from "react";
import {

```

```

SubmitHandler,
useForm,
UseFormRegister,
UseFormRegisterReturn,
} from "react-hook-form";
import { useMutation, useQuery } from "react-query";
import { useNavigate, useParams } from "react-router-dom";
import { toast } from "react-toastify";
import { Header } from "../components/Navbar/Header";
import { getDetailProduct } from "../utils/data";
import { formatNumber } from "../utils/helper";
import { ICheckoutParam, storeCheckout } from "../utils/postData";

const FileUpload = ({ register }: { register: UseFormRegisterReturn }) => {
  let [fileData, setFile] = useState({} as File);

  return (
    <div className="mb-2">
      <div className="rounded-lg shadow-xl bg-gray-50">
        <div className="p-4">
          <label className="inline-block mb-2 text-gray-500">
            Upload Image(jpg,png,svg,jpeg)
          </label>
          <div className="relative flex items-center justify-center w-full">
            {fileData && fileData.name && (
              <img
                src={URL.createObjectURL(fileData)}
                className="absolute h-full mx-auto rounded-lg "
                style={{
                  height: "100px",
                }}
              ></img>
            )}
            <label className="flex flex-col w-full h-32 border-4 border-dashed
            hover:bg-gray-100 hover:border-gray-300">
              <div className="flex flex-col items-center justify-center pt-7">
                <svg
                  xmlns="http://www.w3.org/2000/svg"
                  className="w-8 h-8 text-gray-400 group-hover:text-gray-600"
                  fill="none"
                  viewBox="0 0 24 24"
                  stroke="currentColor"
                >
                  <path
                    strokeLinecap="round"
                    strokeLinejoin="round"
                    strokeWidth="2"
                    d="M7 16a4 4 0 0 1-.88-7.903A5 5 0 1 115.9 6L16 6a5 5 0 0 11
                    9.9M15 13l-3-3m0 0l-3 3m3-3v12"

```

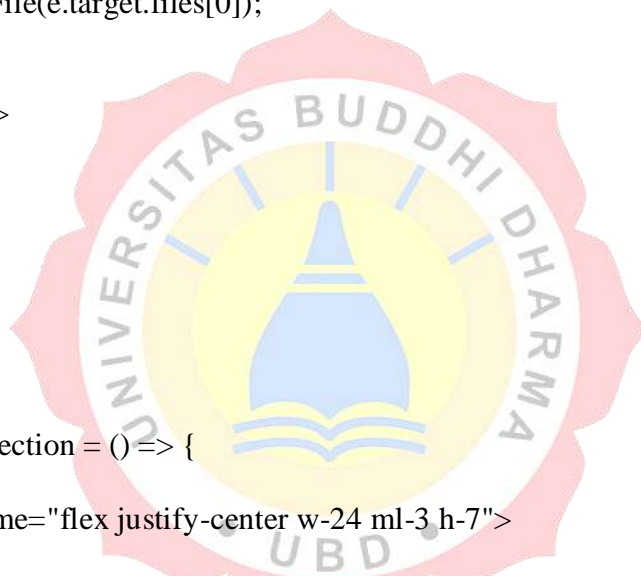
```

        />
      </svg>
      <p className="pt-1 text-sm tracking-wider text-gray-400 group-
hover:text-gray-600">
        { fileData && fileData.name
          ? fileData.name
            : "Select a photo" }
      </p>
    </div>
    <input
      type="file"
      className="opacity-0"
      required
      {...register}
      onChange={(e) => {
        if (!e.target.files) return;
        setFile(e.target.files[0]);
      }}
    />
  </label>
</div>
</div>
</div>
</div>
</>
);
};

export const Selection = () => {
  return (
    <div className="flex justify-center w-24 ml-3 h-7">
      <select
        className="block w-full px-3 m-0 text-base font-normal text-gray-700
transition ease-in-out bg-white bg-no-repeat border border-gray-300 border-solid
rounded appearance-none form-select bg-clip-padding focus:text-gray-700
focus:bg-white focus:border-blue-600 focus:outline-none"
        aria-label="Default select example"
      >
        <option value="1" className="text-xs">
          Month
        </option>
      </select>
    </div>
  );
};

export const Checkout = () => {
  let { productId } = useParams();
  let navigate = useNavigate();

```



```

let [day, setDuration] = useState(1);

const { data, error, isError, isLoading, isFetching, refetch } = useQuery(
  "detailProduct",
  () => getDetailProduct(productId),
  { keepPreviousData: true, enabled: true }
);
const handleDayPricing = (value: number) => {
  if (value > 0) {
    setDuration(value);
  }
};

const { isLoading: isPosting, mutate: checkoutPost } = useMutation(
  (postData: ICheckoutParam) => {
    const formData = new FormData();
    formData.append("id", data?.data.id);
    for (const property in postData) {
      if (property == "image") {
        formData.append(
          "image",
          postData[property as keyof ICheckoutParam][0]
        );
      } else {
        formData.append(property, postData[property as keyof ICheckoutParam]);
      }
    }
    return storeCheckout(formData);
  },
  {
    onSuccess: (res) => {
      console.log(res);
      navigate("/vendor");
    },
    onError: (err: AxiosError) => {
      toast("Failed Transaction " + err.message, {
        position: "top-right",
        autoClose: 5000,
        hideProgressBar: false,
        closeOnClick: true,
        pauseOnHover: true,
        draggable: true,
        progress: undefined,
      });
    },
  }
);

const { register, handleSubmit, reset } = useForm<ICheckoutParam>();

```

```

const onSubmit: SubmitHandler<ICheckoutParam> = (dataForm) => {
  checkoutPost(dataForm);

  console.log(dataForm);
};

const tax = (day * data?.data.price * 11) / 100;
console.log(data);
return (
  <div>
    <Header></Header>
    <div className="mt-20">
      <h1 className="flex items-center justify-center font-bold text-blue-600
text-md lg:text-3xl">
        Checkout Page
      </h1>
    </div>
    <div className="container p-12 mx-auto">
      <div className="flex flex-col w-full px-0 mx-auto md:flex-row">
        <div className="flex flex-col md:w-full">
          <h2 className="mb-4 font-bold md:text-xl text-heading ">
            Payment Detail
          </h2>
          <form
            onSubmit={ handleSubmit(onSubmit) }
            className="justify-center w-full mx-auto "
          >
            <div className="">
              <div className="space-x-0 lg:flex lg:space-x-4">
                <div className="w-full">
                  <label className="block mb-3 text-sm font-semibold text-gray-
500">
                    Payment Proof
                  </label>
                  <div className="mb-3">
                    Please transfer to BCA : 73123131
                  </div>
                  <div className="mb-3">
                    Please transfer to Mandiri : 33123131
                  </div>
                  <FileUpload register={register("image")}></FileUpload>
                </div>
              </div>
            <div className="flex items-center mt-4">
              <label className="flex items-center text-sm group text-heading">
                <input
                  type="checkbox"
                  className="w-5 h-5 border border-gray-300 rounded focus:outline-
none focus:ring-1 "
                />

```

```

    <span className="ml-2">
      Save this information for next time
    </span>
  </label>
</div>
<div className="relative pt-3 xl:pt-6">
  <label className="block mb-3 text-sm font-semibold text-gray-500">
    Notes (Optional)
  </label>
  <textarea
    name="note"
    className="flex items-center w-full px-4 py-3 text-sm border
border-gray-300 rounded focus:outline-none focus:ring-1 focus:ring-blue-600"
    rows={4}
    placeholder="Notes for delivery"
  ></textarea>
</div>
<div className="mt-4">
  <button className="w-full px-6 py-2 text-blue-200 bg-blue-600
hover:bg-blue-900">
    Process
  </button>
</div>
</div>
</form>
</div>
<div className="flex flex-col w-full ml-0 lg:ml-12 lg:w-2/5">
  <div className="pt-12 md:pt-0 2xl:ps-4">
    <h2 className="text-xl font-bold">Order</h2>
    <div className="mt-8">
      <div className="flex flex-col space-y-4">
        <div className="flex space-x-4">
          <div>
            <img
              src={data?.data.photo}
              alt="image"
              className="w-60"
            />
          </div>
          <div>
            <h2 className="text-xl font-bold">{data?.data.name}</h2>
            <p className="text-sm">{data?.data.description}</p>
            <span className="text-red-600">Price</span>
          </div>
          <div>
            Rp.{formatNumber(parseInt(data?.data.price))} Per Month
          </div>
        </div>
        <div className="flex items-center mt-2">
          <button
            onClick={() => handleDayPricing(day - 1)}

```



```

600" className="text-gray-500 focus:outline-none focus:text-gray-
>
<svg
  className="w-5 h-5"
  fill="none"
  strokeLinecap="round"
  strokeLinejoin="round"
  strokeWidth="2"
  viewBox="0 0 24 24"
  stroke="currentColor"
>
<path d="M15 12H9m12 0a9 9 0 11-18 0 9 9 0 0118
0z"></path>
</svg>
</button>
<input
  type="text"
  className="mx-2 text-center text-gray-700 border w-9"
  {...register("total")}
  onChange={(event) =>
    handleDayPricing(parseInt(event.target.value))
  }
  value={day}
></input>
<button
  onClick={() => handleDayPricing(day + 1)}
  className="text-gray-500 focus:outline-none focus:text-gray-
600"
>
<svg
  className="w-5 h-5"
  fill="none"
  strokeLinecap="round"
  strokeLinejoin="round"
  strokeWidth="2"
  viewBox="0 0 24 24"
  stroke="currentColor"
>
<path d="M12 9v3m0 0v3m0-3h3m-3 0H9m12 0a9 9 0 11-18 0
9 9 0 0118 0z"></path>
</svg>
</button>
</div>
</div>
<div>
<button
  onClick={() => navigate("/vendor")}

```

```

        className="text-gray-500 focus:outline-none focus:text-gray-
600"
        >
        <svg
        xmlns="http://www.w3.org/2000/svg"
        className="w-6 h-6"
        fill="none"
        viewBox="0 0 24 24"
        stroke="currentColor"
        >
        <path
        stroke-linecap="round"
        stroke-linejoin="round"
        stroke-width="2"
        d="M6 18L18 6M6 6 12 12"
        />
        </svg>
        </button>
        </div>
        </div>
        </div>
        </div>
        <div className="flex p-4 mt-4">
        <h2 className="text-xl font-bold">ITEMS</h2>
        </div>
        <div className="flex items-center w-full py-4 text-sm font-semibold
border-b border-gray-300 lg:py-5 lg:px-3 text-heading last:border-b-0 last:text-
base last:pb-0">
        Subtotal
        <span className="ml-2">
        { day } Month * Rp.{formatNumber(parseInt(data?.data.price))}
        </span>
        </div>
        <div className="flex items-center w-full py-4 text-sm font-semibold
border-b border-gray-300 lg:py-5 lg:px-3 text-heading last:border-b-0 last:text-
base last:pb-0">
        Tax
        <span className="ml-2">Rp.{formatNumber(tax)}</span>
        </div>
        <div className="flex items-center w-full py-4 text-sm font-semibold
border-b border-gray-300 lg:py-5 lg:px-3 text-heading last:border-b-0 last:text-
base last:pb-0">
        Total
        <span className="ml-2">
        Rp.{formatNumber(day * data?.data.price + tax)}
        </span>
        </div>
        </div>
        </div>
        </div>
        </div>

```

```
    </div>
  </div>
);
};
```

Home.tsx

```
import { Content } from "../components/Content/Content";
import { Navbar } from "../components/Navbar/Navbar";
import { useState } from "react";
import { useQuery } from "react-query";
import { getListShop } from "../utils/data";

const Home = () => {
  const [search, setSearch] = useState("");

  const { data, error, isError, isLoading, isFetching, refetch } = useQuery(
    ["home", search],
    () => getListShop(search)
  );
  const searchHandle = (value: string) => {
    setSearch(value);
  };
  console.log(data);
  return (
    <div className="flex flex-col">
      <Navbar searchHandle={searchHandle}></Navbar>
      <Content
        data={data}
        error={error}
        isLoading={isLoading}
        isError={isError}
        isFetching={isFetching}
      ></Content>
    </div>
  );
};

export default Home;
```

Profile.tsx

```
import React from "react";
import { ProfileForm } from "../components/custom/ProfileForm";
import SidebarProfile from "../components/layouts/SidebarProfile";
import { Header } from "../components/Navbar/Header";
import { useStore } from "../store/store";

let listMenu = [
```

```

    { label: "Personal Info", href: "/profile" },
    { label: "History Transaction", href: "/transaction" },
  ];

```

```

export const Profile = () => {
  const userProfile = useStore((state) => state.userProfile);
  const userJoin = new Date(userProfile.join);

  const ProfileCard = () => (
    <div className="p-3 bg-white border-t-4 border-blue-400 w-72">
      <div className="overflow-hidden image">
        <div className="flex items-center justify-center mx-auto border-2 rounded-
full w-28 h-28">
          <div className="pb-3 text-6xl font-bold">
            {userProfile.name.charAt(0)}
          </div>
        </div>
      </div>
      <h1 className="my-1 text-xl font-bold leading-8 text-gray-900">
        {userProfile.name}
      </h1>
      <h3 className="leading-6 text-gray-600 font-lg text-semibold">
        Status as{" "}
        {userProfile.role === 0 ? <span>Tenant</span> : <span>Renter</span>}
      </h3>
      <p className="text-sm leading-6 text-gray-500 hover:text-gray-600"></p>
      <ul className="px-3 py-2 mt-3 text-gray-600 bg-gray-100 divide-y rounded
shadow-sm hover:text-gray-700 hover:shadow">
        <li className="flex items-center py-3">
          <span>Status</span>
          <span className="ml-auto">
            <span className="px-2 py-1 text-sm text-white bg-green-500 rounded">
              Active
            </span>
          </span>
        </li>
        <li className="flex items-center py-3">
          <span>Login since</span>
          <span className="ml-auto">
            {userJoin.toLocaleDateString("us-EN", {
              weekday: "long",
              year: "numeric",
              month: "numeric",
              day: "numeric",
            })}
          </span>
        </li>
      </ul>
    </div>
  );

```

```

return (
  <div>
    <Header></Header>
    <div className="flex">
      <SidebarProfile listMenu={listMenu}></SidebarProfile>
      <div className="flex mx-auto">
        <div className="mr-3 ">
          <ProfileCard></ProfileCard>
        </div>
        <ProfileForm></ProfileForm>
      </div>
    </div>
  </div>
</div>
);
};

```

Request.tsx

```

import FormRequest from "../components/Content/FormRequest";
import Requester from "../components/Content/Requester";
import { Header } from "../components/Navbar/Header";

const Request = () => {
  return (
    <div className="h-screen overflow-scroll">
      <Header></Header>
      <div className="flex justify-center pt-6 space-x-4 ">
        <Requester></Requester>
        <FormRequest></FormRequest>
      </div>
    </div>
  );
};

```

export default Request;

Transaction.tsx

```

import React, { useState } from "react";
import { useQuery } from "react-query";
import { useNavigate } from "react-router-dom";
import { TableCustom } from "../components/custom/table/Table";
import SidebarProfile from "../components/layouts/SidebarProfile";
import { Header } from "../components/Navbar/Header";
import { useStore } from "../store/store";
import { getListTransaction } from "../utils/data";

```

```

let listMenu = [

```

```

    { label: "Personal Info", href: "/profile" },
    { label: "Transaction History", href: "/transaction" },
  ];

```

```

export const Transaction = () => {
  const userProfile = useStore((state) => state.userProfile);
  const userJoin = new Date(userProfile.join);

  const [currentPage, setCurrentPage] = useState(1);

  const { data, error, isError, isLoading, isFetching, refetch } = useQuery(
    ["trans", currentPage],
    () => getListTransaction(currentPage),
    { keepPreviousData: true }
  );

  const config: Array<{ title: string; key: string; type?: string }> = [
    { title: "Photo", key: "photoProduct", type: "image" },
    { title: "Name", key: "productName" },
    { title: "Date", key: "created_at" },
    { title: "Price", key: "productPrice" },
    { title: "Location", key: "location" },
    { title: "Total Payment", key: "total" },
  ];

  console.log(data);
  const handleChange = (value: number) => {
    setCurrentPage(value);
  };
  const navigate = useNavigate();

  return (
    <div>
      <Header></Header>
      <div className="flex">
        <SidebarProfile listMenu={listMenu}></SidebarProfile>
        <div className="flex mx-auto">
          <TableCustom
            data={data}
            config={config}
            currentPage={currentPage}
            error={error}
            isLoading={isLoading}
            isError={isError}
            isFetching={isFetching}
            subTitle="List Transaction History"
            title="Transaction history"
            handleChange={handleChange}
            handleCustom={(id, action) => navigate(`/invoice/${id}`)}
          ></TableCustom>
        </div>
      </div>
    </div>
  );
}

```



```
    </div>
  );
};
```

Vendor.tsx

```
import List from "../components/List/List";
import { Header } from "../components/Navbar/Header";
```

```
const Vendor = () => {
  return (
    <div className="flex flex-col ">
      <Header></Header>
      <List></List>
    </div>
  );
};
```

```
export default Vendor;
```

Report.tsx

```
import { AxiosError } from "axios";
import { useState } from "react";
import { useMutation, useQuery } from "react-query";
import { toast } from "react-toastify";
import { TableCustom } from "../../components/custom/table/Table";
import Admin from "../../components/layouts/Admin";
import { getListTransactionVendor } from "../../utils/data";
import { storeApprovalOrder } from "../../utils/postData";
const Order = () => {
  const [currentPage, setCurrentPage] = useState(1);

  const { data, error, isError, isLoading, isFetching, refetch } = useQuery(
    ["trans", currentPage],
    () => getListTransactionVendor(currentPage),
    { keepPreviousData: true }
  );

  const config: Array<{ title: string; key: string; type?: string }> = [
    { title: "Photo Product", key: "photo", type: "image" },
    { title: "Photo Payment", key: "photoPayment", type: "image" },
    { title: "Name Customer", key: "custName" },
    { title: "Total", key: "price" },
    { title: "Unit", key: "unit" },
  ];

  console.log(data);
  const handleChange = (value: number) => {
    setCurrentPage(value);
  };
};
```

```

const approvalPost = useMutation(
  (dataPost: { type: string; id: string }) => {
    return storeApprovalOrder(dataPost.type, dataPost.id);
  },
  {
    onSuccess: (res) => {
      refetch();
      toast("Success Submit ", {
        position: "top-right",
        autoClose: 5000,
        hideProgressBar: false,
        closeOnClick: true,
        pauseOnHover: true,
        draggable: true,
        progress: undefined,
      });
    },
    onError: (err: AxiosError) => {
      toast("Failed Submit " + err.message, {
        position: "top-right",
        autoClose: 5000,
        hideProgressBar: false,
        closeOnClick: true,
        pauseOnHover: true,
        draggable: true,
        progress: undefined,
      });
    },
  }
);

const handleApproval = (id: string, action: string) => {
  approvalPost.mutate({ type: action, id });
};

return (
  <Admin>
  <div className="flex-initial w-full overflow-auto">
    <TableCustom
      data={data}
      config={config}
      currentPage={currentPage}
      error={error}
      isLoading={isLoading}
      isError={isError}
      isFetching={isFetching}
      subTitle="List Order"
      title="Order"
      handleChange={handleChange}
      handleCustom={({ id: string, action: string }) =>
        handleApproval(id, action)
    }
    />
  </div>
)

```

```
    }  
  <</TableCustom>  
</div>  
</Admin>  
)  
};  
  
export default Order;
```



Tanggal : 18 Maret 2022
Waktu : 15.00 wib
Narasumber : Bapak Hermanto
Jabatan : Pemilik Kost-an
Tempat : Jalan Imam Bonjol, Gg.Bansin no.7A., Karawaci, Tangerang

P : Penulis N : Narasumber

- P : Selamat sore pak Hermanto, saya Stefunus Kristianto dari Universitas Buddhi Dharma ingin meminta waktu nya sebentar untuk mewawancari bapak mengenai Kost-an bapak ini.
- N : Iya Boleh, Silakan mau tanya tentang apa?
- P : Baik sebelumnya saya berterima kasih pak, Tujuan saya mewawancari bapak untuk memenuhi kelengkapan skripsi saya, dimana saya ingin bertanya tentang apa saja yang diperlukan seorang Pelanggan / pendatang untuk mulai menghuni di kost-an bapak?
- N : Yang diperlukan pertama untuk orang baru itu, Ktp saja yang paling utama ya. Terus untuk pembayaran itu bisa dari kesepakatan dari saya dengan pelanggan mau dibuat kan invoice atau tidak,begitu saja sih.
- P : Baik jadi untuk data yang wajib di siap kan sebelum masuk Cuma 2 ya pak, ktp sama kebutuhan invoice untuk pelanggan dengan catatan kesepakatan ber dua ya pak ingin dibuatkan atau tidak ya.
- N : Iya bener sekali, saya rasa rata – rata Kost-an sekitar yang biasa sampai elite juga begitu.
- P : Baik pak,terima kasih,selanjutnya apakah ada permasalahan selama menyewakan Kost-an ini pak?
- N : Permasalahan yang biasa saya dapati sih;
1. Banyak yang tidak tau kalau disini terdapat kost-an.
 2. Kost-an ini masih ada yang kosong atau tidak kamar yang tersedia.
 3. Banyak yang menebak-nebak harga bahkan gak tau,makanya tidak memberanikan diri untuk datang kesini.
 4. Kurang terkenal didaerah sini, palingan Cuma dari mulut ke mulut saja.
- P : Oke pak,terima kasih untuk jawabannya,dengan ini saya mengamati beberapa masalah tadi maka saya akan mencoba untuk mengimplementasikan pada judul penelitian saya. nanti saya akan memasukan ini dalam paper saya mengenai sistem informasi kost-an di tangerang ini dengan sistem berbasis website, dimana proses pencarian kost-an lebih cepat dan efektif,ada nya informasi lengkap tentang kost-an yang akan memudahkan para pelanggan,dan untuk mempromosikan kost-an secara online, mungkin dengan begini akan membantu proses penyewahan kost-an,dengan begini bapak mengijinkan untuk dimasukan ke dalam sistem website kami?
- N : Boleh silakan,semoga nanti penelitian ini membantu para penyewa kost-an yang lain juga termasuk membantu saya dalam menyewahkan kost-an ini.
- P : Saya harap demikian pak,sebelumnya saya mengucapkan terima kasih banyak atas waktu yang telah diberikan dalam wawancara ini dan sangat membantu dalam keperluan paper saya,sekali lagi saya berterima kasih pak.

Tangerang, 18 Maret 2022

Narasumber



Hermanto

LAMPIRAN SOFTWARE SCENARIO TESTING DOCUMENTATION

Universitas Buddhi Dharma

Software Scenario Testing Documentation

Nama Website : Kost Gilak
Platform : Website
Minimal H/W :
CPU : Intel(R) Celeron(R) N4000 CPU
@ 1.10GHz 1.10 GHz
RAM : 8 GB
OS : Win 10 64-bit
Minimal S/W :
Framework : Laravel 9
: ReachJS 18.2
Language : Es2021
: php 7.4
Database : Mysql 8
Database Tools : Http Server | Apache 2.4.52
Developer Tools : Php Myadmin 5.2.0
: Visual Studio Code 1.6.8
Icon Website : -

Scenario List :

Code	Scenario Name	Description
010	Tampilan home	Menampilkan tampilan awal pada website
110	Menu Login	Melakukan proses login dahulu saat ingin melakukan transaksi di website
112	Menu registrasi tenant	Menu form registrasi tenant
113	Menu registrasi renter	Menu form registrasi renter
120	Menu shop	Menampilkan produk yang ada diwebsite
101	Menu admin renter product	Menu untuk user menambahkan produk kost yang ia punya
102	Menu admin renter report	Menu bagi user untuk menerima data transaksi masuk
105	Menu tenant/report transaksi	Menu user untuk melihat transaksi yang sedang dilakukan
109	Menu profil	Menampilkan informasi user

Positive Scenario

No	Code	Information	Step	Expected Result	Actual Result	Pass/Fail
1	010	Sudah membuka website nya	Memasukan link website	Muncul tampilan website	Tampilan muncul	Pass
	110	sudah mengklik menu sign in di halaman awal	Mengklik sign in	Menampilkan form sign in	- muncul - bisa login	Pass
	112	Sudah membuka menu Registrasi	Mengklik pilihan registrasi tenant	Menampilkan form registrasi tenant	- muncul - bisa register	Pass
	113	Sudah membuka menu Registrasi	Mengklik pilihan registrasi renter	Menampilkan form registrasi renter	- muncul - bisa register	Pass
	120	User sudah login	Mengklik menu shop	Menampilkan halaman produk untuk di lakukan transaksi	- muncul - bisa transaksi	Pass
	101	Sudah login sebagai user Renter	Mengklik menu renter product	Menampilkan halaman form untuk menambahkan produk	- muncul - bisa menambah bahan	Pass
	102	Sudah login sebagai user Renter	Mengklik menu renter report	Menampilkan form transaksi	- muncul - bisa transaksi	Pass
	105	Sudah login sebagai user Tenant	Mengklik menu tenant report	Menampilkan history transaksi	- muncul -	Pass
	109	Sudah login sebagai user	Mengklik menu profil	Menampilkan data diri renter atau tentant	- muncul	Pass

Negative Scenario

No	Code	Information	Step	Negative Condition	Expected Result	Actual Result	Pass/Fail
1	010	Website sudah dibuka	Memasukan alamat link ke menu	Tidak ada jaringan internet	Tidak dapat mengakses website	error page	Pass

			pencarian browser				
110	User dalam menu login	Mengklik menu sign in pada website	Memasukan email dan password salah. Tidak memasukan email atau password atau Cuma salah satu aja.	Munculnya pop up kesalahan login	Tidak bisa login	pass	
112	User membuka form registrasi tenant	Sudah memilih level user sebagai Tenant	Tidak mengisi sesuai ketentuan form yang telah disediakan. Salah satu form tidak diisi.	Terjadi penolakan dan pop up error untuk mengisi ulang	Tidak bisa register	pass	
113	User membuka form registrasi renter	Sudah memilih level user sebagai Renter	Tidak mengisi sesuai ketentuan form yang telah disediakan. Tidak mengisi semua form yang tersedia.	Terjadi penolakan dan pop up error untuk mengisi ulang	registrasi gagal	pass	
120	User membuka halaman shop	User sudah login/register bisa mengakses	Tidak terdapat jaringan internet. Database rusak. Database produk error.	Halaman akan kosong. Muncul kode error di halaman website.	error page	pass	

101	User admin renter didalam menu product	Terdaftar sebagai user Renter	Mengisi form tidak sesuai. Angka nominal tidak ada tanda baca. Tidak bisa memasukkan photo. Alamat dan deskripsi terlalu panjang.	Data produk yang tidak sesuai akan ditolak dan memunculkan pop up error. Kesulitan memahami jumlah.	Tidak memunculkan Data	Pass
102	User admin renter membuka report transaksi	Terdaftar sebagai user Renter	Tidak ada koneksi internet/terputus dengan server. Tidak terdapat menu detail. Tidak ada notifikasi pembelian.	Data yang dikeluarkan akan error. Tidak bisa melihat secara besar detail transaksi yang sudah terjadi. Pembelian tertunda jika vendor belum menyetujui pembelian.	website - usony - Tidak ada notifikasi	Pass
105	User tenant membuka Menu history transaksi	Terdaftar sebagai user Tenant	Pembelian belum di accept sama penjual. Pembeli tidak mengetahui pembelian sudah diterima oleh vendor.	Tidak akan menampilkan data pembelian.	- Tidak Terdegar Transaksi	Pass

	109	User dalam menu profil	Sudah mempunyai akun	Akun tidak terdaftar.	Tidak bisa login.	Tidak seribuk ^a	pass
--	-----	------------------------	----------------------	-----------------------	-------------------	----------------------------	------

Tangerang, 03 Maret 2022



William Lim

Scenario List:

Code	Scenario Name	Description
010	Tampilan home	Menampilkan tampilan awal pada website
110	Menu Login	Melakukan proses login dahulu saat ingin melakukan transaksi di website
112	Menu registrasi tenant	Menu form registrasi tenant
113	Menu registrasi renter	Menu form registrasi renter
120	Menu shop	Menampilkan produk yang ada di website
101	Menu admin renter product	Menu untuk user menambahkan produk kost yang ia punya
102	Menu admin renter report	Menu bagi user untuk menerima data transaksi masuk
105	Menu tenant/report transaksi	Menu user untuk melihat transaksi yang sedang dilakukan
109	Menu profil	Menampilkan informasi user

Positive Scenario

No	Code	Information	Step	Expected Result	Actual Result	Pass/Fail
1	010	Sudah membuka website nya	Memasukan link website	Muncul tampilan website	Tampil	pass
	110	sudah mengklik menu sign in di halaman awal	Mengklik sign in	Menampilkan form sign in	tidak login	pass

112	Sudah membuka menu Registrasi	Mengklik pilihan registrasi tenant	Menampilkan form registrasi tenant	Bisa regis	pass
113	Sudah membuka menu Registrasi	Mengklik pilihan registrasi renter	Menampilkan form registrasi renter	Bisa regis	pass
120	User sudah login	Mengklik menu shop	Menampilkan halaman produk untuk di lakukan transaksi	Bisa transaksi transaksi	pass
101	Sudah login sebagai user Renter	Mengklik menu renter product	Menampilkan halaman form untuk menambahkan produk	Bisa menambahkan produk	pass
102	Sudah login sebagai user Renter	Mengklik menu renter report	Menampilkan form transaksi	Bisa transaksi	pass
105	Sudah login sebagai user Tenant	Mengklik menu tenant report	Menampilkan history transaksi	Bisa melihat	pass

	109	Sudah login sebagai user	Mengklik menu profil	Menampilkan data diri renter atau tentant	muncul	pass
--	-----	--------------------------	----------------------	-------------------------------------------	--------	------

Negative Scenario

No	Code	Information	Step	Negative Condition	Expected Result	Actual Result	Pass/Fail
1	010	Website sudah dibuka	Memasukan alamat link ke menu pencarian browser	Tidak ada jaringan internet	Tidak dapat mengakses website	Tidak ada	pass
	110	User dalam menu login	Mengklik menu sign in pada website	Memasukan email dan password salah. Tidak memasukan email atau password atau Cuma salah satu aja.	Munculnya pop up kesalahan login	pop up muncul	pass
	112	User membuka	Sudah memilih	Tidak mengisi	Terjadi penolakan		

		form resgistrasi tenant	level user sebagai Tenant	sesuai ketentuan form yang telah disediakan. Salah satu form tidak diisi.	dan pop up error untuk mengisi ulang	<i>harus mengisi semua bagian</i>	<i>pass</i>
113	User membuka form resgistrasi renter	Sudah memilih level user sebagai Renter	Tidak mengisi sesuai ketentuan form yang telah disediakan. Tidak mengisi semua form yang tersedia.	Terjadi penolakan dan pop up error untuk mengisi ulang	<i>harus mengisi dengan benar</i>	<i>pass</i>	
120	User membuka halaman shop	User sudah login/registre r bisa mengakses	Tidak terdapat jaringan internet.	Halaman akan kosong. Muncul kode error	<i>error page</i>	<i>pass</i>	

				Database rusak. Database produk error.	di halaman website.		
101	User admin renter didalam menu product	Terdaftar sebagai user Renter	Mengisi form tidak sesuai. Angka nominal tidak ada tanda baca. Tidak bisa memasukan photo. Alamat dan deskripsi terlalu panjang.	Data produk yang tidak sesuai akan ditolak dan memunculkan pop up error. Kesulitan memahami jumlah.	produk kosong	pass	
102	User admin renter membuka report transaksi	Terdaftar sebagai user Renter	Tidak ada koneksi internet/terputus dengan server. Tidak terdapat menu detail.	Data yang dikeluarkan akan error. Tidak bisa melihat secara besar detail transaksi	tidak bisa diakses	Fail	

				Tidak ada notifikasi pembelian.	yang sudah terjadi. Pembelian tertunda jika vendor belum menyetujui pembelian.		
105	User tenant membuka Menu history transaksi	Terdaftar sebagai user Tenant	Pembelian belum di accept sama penjual. Pembeli tidak mengetahui pembelian sudah diterima oleh vendor.	Tidak akan menampilkan data pembelian.	Pembelian kosong Tidak Terdaftar	Fail	
109	User dalam menu profil	Sudah mempunyai akun	Akun tidak terdaftar.	Tidak bisa login.	Tidak bisa diakses	Pass	

Tangerang, 03 Maret 2022


Hermanto



UNIVERSITAS BUDDHI DHARMA

Jl. Imam Bonjol No. 41 Karawaci Ilir, Tangerang
021 5517853 / 021 5586822 admin@buddhidharma.ac.id

KARTU BIMBINGAN TA/SKRIPSI

NIM : 20180700040
Nama Mahasiswa : STEFUNUS KRISTIANTO
Fakultas : Sains dan Teknologi
Program Studi : Sistem Informasi
jenjang : Strata Satu
Tahun Akademik/Semester : 2022/2023 Ganjil
Dosen Pembimbing : Riki, M.Kom
Judul Skripsi : ANALISIS DAN PENYEWAAN SISTEM INFORMASI KOST-AN E
KOTA TANGERANG

Tanggal	Catatan	Paraf
2022-03-19	Briefng: Prosedur bimbingan	
2022-03-26	BAB1: Isu/Masalah, Rumusan dan Objek/Tujuan	
2022-04-02	Review	
2022-04-09	BAB 2: Membuat Literature Review	
2022-04-16	BAB 3: Metode/Pendekatan dan Teknik Analisa	
2022-04-23	Review	
2022-04-30	BAB 4: Hasil Pendekatan dan Analisa	
2022-05-07	Review	
2022-05-14	BAB 4: Hasil Pendekatan dan Analisa	

Mengetahui

Ketua Program Studi

Benny Daniawan, M.Kom

Tangerang, 22 August 2022

Pembimbing

Riki, M.Kom