

Aplikasi Manajemen Kos-Kosan Berbasis Web Menggunakan Framework Express.js dengan Database MySQL

Tegar Robi Bakhtiar*¹, Khoiriya Latifah²

^{1,2} Program Studi Informatika, Universitas PGRI Semarang, Kota Semarang

*Email: tegarhemmm0@gmail.com

Abstract

In the digital era that continues to develop, information technology plays an important role in the transformation of various sectors of life, including property management such as boarding houses. The need for efficient temporary housing is increasing, especially among students, workers and migrating individuals. To overcome this problem, the development of a web-based boarding house management application is the main focus. Development of a boarding house management application using the Express.js framework for web development and MySQL as a relational database management system. The Express.js framework was chosen because of its ability to facilitate the development of fast and scalable server-side web applications, while MySQL was chosen because of its reliability and performance in storing and managing data. This application development was also carried out using the Agile method. The Agile method was chosen because of its iterative and gradual approach. This application is designed to provide various important features, from searching for boarding houses based on certain criteria to managing transactions such as recording payments and communication between boarding house owners and tenants. By utilizing web technology, this application not only increases operational efficiency for boarding house owners but also improves the user experience with a more user-friendly interface. The result of combining Express.js and MySQL is a comprehensive solution that not only simplifies the property management process, but also increases user satisfaction through efficient data accessibility and an overall better user experience.

Keywords: Express.js, Agile, MySQL

Abstrak

Di era digital yang terus berkembang, teknologi informasi memainkan peran penting dalam transformasi berbagai sektor kehidupan, termasuk manajemen properti seperti kos - kosan. Kebutuhan akan tempat tinggal sementara yang efisien semakin meningkat, terutama di kalangan mahasiswa, pekerja, dan individu yang merantau. Untuk mengatasi masalah tersebut, pengembangan aplikasi manajemen kos-kosan berbasis web menjadi fokus utama. Pengembangan aplikasi manajemen kos-kosan menggunakan framework Express.js untuk pengembangan web dan MySQL sebagai sistem manajemen basis data relasional. Framework Express.js dipilih karena kemampuannya dalam memfasilitasi pembangunan aplikasi web *server-side* yang cepat dan skalabel, sementara MySQL dipilih karena keandalan dan performanya dalam menyimpan dan mengelola data. Pengembangan aplikasi ini juga dilakukan dengan menggunakan metode *Agile Development*. Metode *Agile Development* dipilih karena pendekatannya yang iteratif dan bertahap. Aplikasi ini dirancang untuk menyediakan berbagai fitur penting, mulai dari pencarian kamar kos berdasarkan kriteria spesifik hingga manajemen transaksi seperti pencatatan pembayaran dan komunikasi antara pemilik kos dan penyewa. Dengan memanfaatkan teknologi web, aplikasi ini tidak hanya meningkatkan efisiensi operasional bagi pemilik kos-kosan tetapi juga memperbaiki pengalaman pengguna dengan antarmuka yang lebih ramah pengguna. Hasil dari penggabungan Express.js dan MySQL adalah solusi komprehensif yang tidak hanya mempermudah proses manajemen properti, tetapi juga meningkatkan kepuasan pengguna

melalui aksesibilitas data yang efisien dan pengalaman pengguna yang lebih baik secara keseluruhan.

Kata Kunci: Express.js, Agile, MySQL

1. Pendahuluan

Dalam era digital yang terus berkembang, teknologi informasi telah menjadi elemen penting dalam berbagai aspek kehidupan, termasuk sektor properti dan manajemen kos-kosan. Permintaan akan tempat tinggal sementara yang praktis dan efisien, terutama di kalangan mahasiswa dan pekerja, semakin meningkat. Hal ini mendorong perlunya sebuah sistem manajemen kos-kosan yang mampu mengoptimalkan proses pencarian, pemesanan, dan pengelolaan kos-kosan[1]

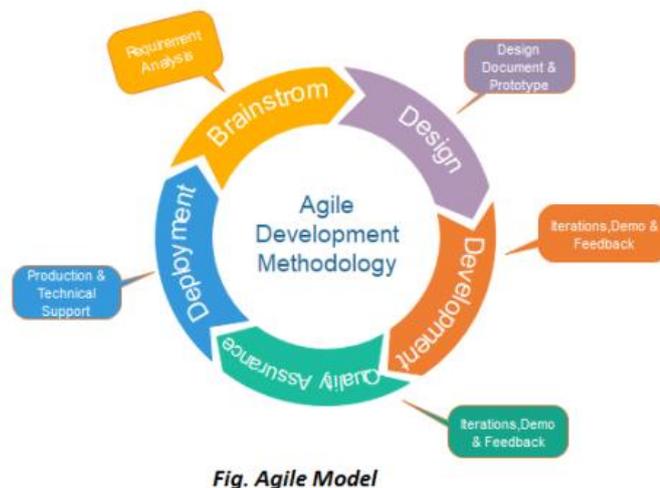
Bagi mahasiswa, pekerja, atau pelajar baru yang sedang merantau ke kota lain untuk bekerja maupun menempuh ilmu, mencari tempat tinggal merupakan prioritas utama sebelum mereka memperlengkapi kebutuhan lainnya. Tempat tinggal menjadi kebutuhan dasar yang harus dipenuhi untuk menjamin kenyamanan dan keamanan selama masa perantauan[2]. Para pencari kos memiliki berbagai cara untuk mendapatkan informasi tentang rumah kos yang sesuai dengan kebutuhan mereka. Beberapa mendapatkan informasi dari teman atau kenalan yang sudah lebih dulu mendapatkan tempat tinggal, sementara yang lain memilih untuk mencari informasi melalui media sosial seperti Facebook atau sumber informasi singkat yang tersedia di internet. Namun, metode tradisional ini seringkali tidak efisien dan kurang sistematis[3].

Aplikasi manajemen kos-kosan berbasis web hadir sebagai solusi inovatif yang menggabungkan teknologi terkini untuk memberikan kemudahan bagi pemilik kos-kosan dalam mengelola properti mereka, serta memudahkan penyewa dalam mencari dan memesan kamar kos. Penggunaan teknologi web memungkinkan aksesibilitas yang luas dan pengelolaan data yang efisien. Framework Express.js, salah satu framework JavaScript populer untuk pengembangan aplikasi web, dikenal karena kemampuannya dalam memfasilitasi pengembangan aplikasi server-side yang cepat dan efisien[4]. Express.js menyediakan berbagai fitur yang mendukung pembuatan aplikasi yang scalable dan maintainable[5]. Di sisi lain, MySQL sebagai sistem manajemen basis data relasional menawarkan keandalan dan kinerja tinggi dalam penyimpanan serta pengelolaan data yang kompleks.

Dengan mengintegrasikan Express.js dan MySQL, aplikasi manajemen kos-kosan ini menawarkan solusi lengkap, mulai dari pencarian kamar kos berdasarkan kriteria tertentu hingga fitur-fitur pengelolaan kos-kosan seperti pencatatan pembayaran, manajemen kamar, dan komunikasi antara pemilik kos dan penyewa. Implementasi ini tidak hanya meningkatkan efisiensi operasional tetapi juga meningkatkan kepuasan pengguna melalui pengalaman yang lebih terstruktur dan user-friendly.

2. Metode

Metode yang digunakan dalam pengembangan aplikasi Manajemen kos-kosan Berbasis Web yaitu menggunakan metode *Agile Development*. Metode *Agile Development* adalah kerangka kerja pengembangan perangkat lunak yang menekankan pada kolaborasi tim yang intensif dan pengiriman produk secara bertahap melalui serangkaian iterasi singkat yang disebut sprint[6]. Pendekatan ini telah terbukti efektif dalam mengatasi kompleksitas dan perubahan yang sering terjadi dalam pengembangan aplikasi. Fokus utama menggunakan framework Express.js dan Database MySQL untuk merancang dan mengimplementasikan solusi yang efisien dan responsif. Langkah-langkah pengembangan yang terperinci, mulai dari analisis kebutuhan hingga deployment, diterapkan dengan cermat. Dengan metode *Agile Development*, berupaya untuk memberikan panduan yang praktis dan terperinci. Berikut visualisasi model iteratif:

Gambar 1. Pengembangan Metode *Agile Development*

Berikut adalah tahapan-tahapan utama dalam metodologi ini:

1. *Deployment* (Penyediaan): Tahap awal dimulai dengan persiapan lingkungan pengembangan dan penerapan infrastruktur yang diperlukan untuk memulai proyek. Tahap ini meliputi penyiapan server, database, dan konfigurasi dasar aplikasi.
2. *Brainstorm* (Pemikiran Bersama): Tim melakukan sesi brainstorming untuk mengidentifikasi kebutuhan utama dan tujuan dari aplikasi manajemen kos-kosan ini. Ide-ide awal dikumpulkan dan dianalisis untuk membentuk dasar rencana pengembangan lebih lanjut.
3. *Design* (Perancangan): Dalam tahap ini, tim fokus pada perancangan arsitektur aplikasi dan pengembangan prototipe awal. Desain UI/UX dibuat untuk memastikan aplikasi memiliki antarmuka yang intuitif dan ramah pengguna.
4. *Development* (Pengembangan): Pada tahapan ini, tim mulai mengimplementasikan desain yang telah dirancang sebelumnya. Tim menggunakan Express.js untuk menulis route (routes), yang telah dirancang, memverifikasi pengguna dengan middleware (middleware), mengatur logika bisnis dalam kontroler (controller), dan mengintegrasikan model (models) dengan database MySQL. Fitur-fitur utama seperti pencarian kamar kos, tipe-tipe kos, manajemen pembayaran, dan komunikasi antara pemilik dan penyewa dikodekan dan diimplementasikan.
5. *Quality Assurance* (Jaminan Kualitas): Setiap tahap pengembangan diikuti oleh pengujian menyeluruh untuk memastikan kualitas dan konsistensi aplikasi. Uji fungsional, uji integrasi, dan uji performa dilakukan untuk menemukan dan memperbaiki bug serta memastikan aplikasi berjalan dengan lancar.

Metode *Agile Development* tidak hanya memungkinkan pengembangan yang iteratif dan responsif terhadap perubahan, tetapi juga memastikan bahwa aplikasi akhir memenuhi standar kualitas yang tinggi dan memuaskan kebutuhan pengguna dengan efektif. Dengan demikian, aplikasi manajemen kos-kosan ini dapat berhasil memenuhi tujuan pengelolaan properti yang efisien dan pengalaman pengguna yang lebih baik[7].

3. Hasil dan Pembahasan

Aplikasi Manajemen Kos-Kosan Berbasis Web Menggunakan Framework Express.js Dengan Database MySQL ini meliputi analisis kebutuhan, perancangan sistem, pengembangan perangkat lunak, dan pengujian aplikasi[8]. Fitur yang dibuat pada aplikasi ini yaitu Untuk pengguna, sistem menyediakan fitur pendaftaran dan otentikasi yang aman, memungkinkan mereka untuk login, mendaftar, dan mengelola profil mereka. Pengguna harus dapat mencari kamar kos berdasarkan berbagai kriteria tertentu seperti lokasi, harga, fasilitas,

melihat detail kamar, melakukan reservasi, dan mengelola pembayaran sewa secara online. Selain itu, pengguna harus dapat melihat riwayat transaksi dan menerima notifikasi terkait pembayaran. Bagi admin kos, sistem harus memungkinkan manajemen data kos-kosan, termasuk penambahan, penghapusan, dan pembaruan informasi kamar seperti harga, fasilitas, dan status ketersediaan.

3.1. Analisi Kebutuhan

Analisis kebutuhan yang pertama adalah Data kamar kos seperti, ketersediaan fasilitas kamar kos jumlah kamar kos, dan ukuran kamar kos. Selain itu analisis kebutuhan ini juga mencakup aspek kebutuhan fungsional dan non-fungsional yang meliputi berbagai fitur untuk dua peran utama, yaitu admin (pemilik kos) dan user (pengguna). Secara fungsional, admin dapat melakukan manajemen properti dan fasilitas seperti menambah, mengedit, dan menghapus informasi kos-kosan, serta melihat dan memproses permintaan penyewaan. Disisi pengguna, mereka dapat melakukan pencarian properti dan fasilitas berdasarkan kriteria tertentu, melihat detail fasilitas, melakukan pemesanan, dan membayar melalui sistem yang terintegrasi.

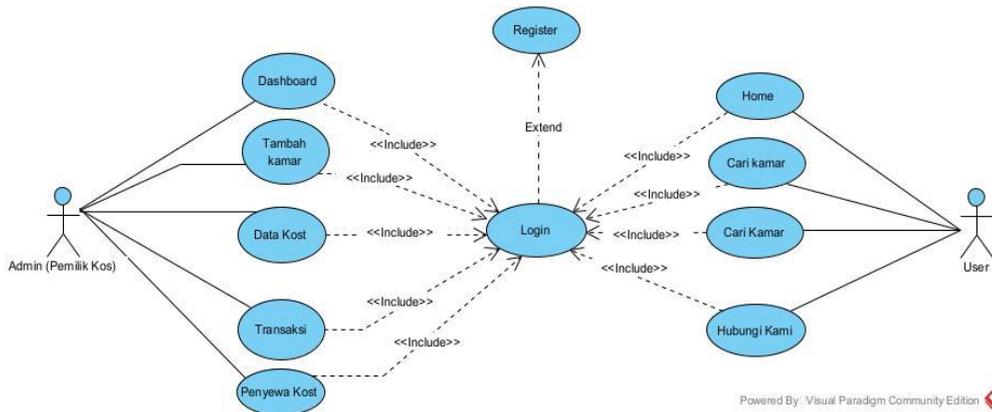
Kebutuhan non-fungsional termasuk keamanan data yang ketat untuk melindungi informasi pengguna dan transaksi, performa aplikasi berbasis web yang responsif agar pengguna yang intuitif untuk meningkatkan pengalaman pengguna secara keseluruhan.

3.2. Perancangan

Aplikasi Manajemen Kos-Kosan Berbasis Web Menggunakan Framework Express.js Dengan Database MySQL ini terdapat dua level pengguna didalamnya yaitu Admin dan User umum[9]. Adapun alur dari masing-masing level pengguna dan fungsinya bisa dilihat pada Use Case Diagram dibawah:

a. Use Case Diagram

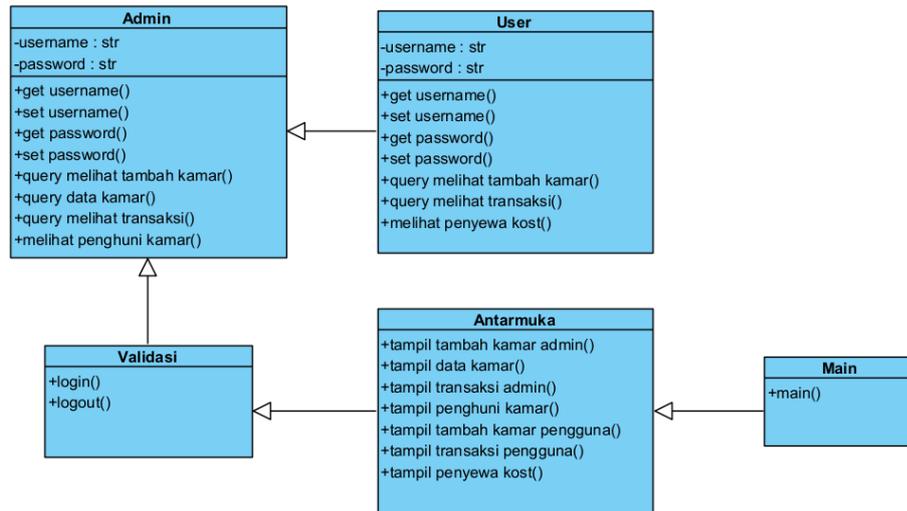
Use Case Diagram menggambarkan hubungan antara sistem dengan aktor yang memfokuskan pada fungsi apa yang ada pada aplikasi, bukan berdasarkan urutan kejadian[5]. Diagram ini menunjukkan fungsionalitas yang disediakan oleh sistem dan bagaimana aktor dapat berinteraksi dengan sistem tersebut.



Gambar 2. Use Case Diagram

b. Class Diagram

Class diagram adalah representasi visual dari struktur kelas dan hubungan antara kelas dalam sistem perangkat lunak. Biasanya, diagram kelas digunakan dalam pemrograman berorientasi objek untuk menggambarkan struktur kelas, properti, dan metode dalam sistem.



Gambar 3. Class Diagram

3.3. Pembembangan Aplikasi

Express.js adalah sebuah framework aplikasi web yang minimalis dan fleksibel, dirancang untuk Node.js. Dikenal karena kemampuannya yang menonjol dalam memfasilitasi pengembangan aplikasi web yang berjalan di sisi server, Express.js juga mempunyai komponen penting dari arsitektur dan fungsionalitas, berikut adalah beberapa Komponen penting yang diimplementasikan:

a. Router

Router adalah komponen dalam Express.js yang bertanggung jawab untuk menangani rute (routes) atau jalur permintaan HTTP. Dengan router, logika penanganan permintaan untuk berbagai jalur URL dapat dipisahkan

```

const router = express.Router();

router.get("/apply-rent", getApplyRent);
router.get("/apply-rent/:id", getApplyRentById);
router.post("/apply-rent", addApplyRent);
router.delete("/apply-rent/:id", deleteApplyRent);

export default router;

```

Gambar 4. Router

b. Controller

Controller adalah modul atau file yang berisi logika aplikasi untuk menangani permintaan dan menentukan respons yang sesuai. Controller biasanya memisahkan logika bisnis routing, membuat kode lebih terorganisir

```

export const getApplyRent = async (req, res) => {
  try {
    const response = await ApplyRent.findAll();
    res.json(response);
  } catch (error) {
    console.log(error.message);
  }
};

export const getApplyRentById = async (req, res) => {
  try {
    const response = await ApplyRent.findOne({
      where: {
        id: req.params.id,
      },
    });
    res.json(response);
  } catch (error) {
    console.log(error.message);
  }
}

```

Gambar 5. Controller

c. Middleware

Middleware adalah fungsi yang dipanggil selama siklus permintaan respons aplikasi. Dalam pengembangan aplikasi ini middleware digunakan untuk memverifikasi token JWT dari header otoritas dalam permintaan HTTP. Jika header atau token tidak ada, middleware mengembalikan status 401. Jika token tidak valid, middleware mengembalikan status 403. Jika valid data pengguna yang dikodekan disimpan di *req.user* dan middleware melanjutkan ke fungsi berikutnya dengan *next()*.

```
import jwt from "jsonwebtoken";

export const verifyToken = (req, res, next) => {
  console.log("Verifying token...");
  const authHeader = req.headers["authorization"];
  if (!authHeader) {
    console.log("No authorization header");
    return res.sendStatus(401); // Unauthorized
  }

  const token = authHeader.split(" ")[1];
  if (!token) {
    console.log("No token provided");
    return res.sendStatus(401); // Unauthorized
  }

  jwt.verify(token, process.env.ACCESS_TOKEN_SECRET, (err, decoded) => {
    if (err) {
      console.log("Failed to verify token", err);
      return res.sendStatus(403); // Forbidden
    }

    req.user = decoded;
    console.log("Token verified:", req.user);
    next();
  });
};
```

Gambar 6. Middleware

d. Model

Model adalah representasi dari struktur data aplikasi. Dalam konteks aplikasi web, model seringkali terkait dengan database dan digunakan untuk mendefinisikan skema data serta berinteraksi dengan basis data

```
let mysql = require('mysql');

let connection = mysql.createConnection({
  host: 'educalab.id',
  user: 'WfAjHTHdptCwP8FA',
  password: 'VxqvThhECCDRVavr',
  database: 'Ckeoq2LSZGLA110',
  port: 3307
});

connection.connect(function(err){
  if(!err) {
    console.log(err);
  } else {
    console.log('Terkoneksi Dengan Databse!')
  }
});

module.exports = connection;
```

Gambar 7. Model

3.4. Pengujian

Pada tahap pengujian sistem ini, dilakukan pengujian langsung pada infrastruktur backend dengan metode *blackbox* testing. Tujuannya adalah memastikan bahwa sistem berfungsi sesuai kebutuhan yang telah ditentukan[10]. Berikut beberapa hasil blackbox testing dari fitur utama:

a. Register

Tabel 1. Pengujian *Blackbox Register*

Kasus Uji	Pengujian yang dilakukan	Hasil Pengujian	Kesimpulan
Register	Memasukan username, email,	Data username, email, password yang di-hash. Dan	Berhasil

	password, confPassword	role tersimpan ke database MySQL	
Password tidak cocok	Memasukan password yang tidak sama	Pesan error "Passwords do not match"	Berhasil
Email sudah ada	Memasukan email yang sudah ada	Pesan error "Email already exists"	Berhasil

Berdasarkan tabel 1, menunjukkan bahwa proses registrasi berjalan sesuai dengan yang diharapkan. Data *username*, *email*, *password* yang di-hash, dan role berhasil disimpan ke database MySQL. Apabila password dan konfirmasi password tidak cocok, sistem memberikan pesan error yang sesuai, "Passwords do not match". Jika email yang dimasukkan sudah terdaftar, sistem memberikan pesan error yang sesuai, misalnya "Email already exists".

b. Login

Tabel 2. Pengujian *Blackbox* Login

Kasus Uji	Pengujian yang Dilakukan	Hasil Pengujian	Kesimpulan
Login Berhasil	Memasukan email dan Password yang benar	Akses token dan refres token berhasil dibuat, data user termasuk role muncul di response	Berhasil
User Tidak Ditemukan	Memasukan email yang tidak terdaftar	Pesan error "User not found" muncul	Berhasil
Password Salah	Memasukan password yang salah	Pesan error "wrong password"	Berhasil

Berdasarkan tabel 2, Pengujian login menunjukkan bahwa sistem dapat menangani berbagai skenario dengan benar. Ketika pengguna memasukkan email dan password yang benar, sistem berhasil menghasilkan akses token dan refresh token serta menampilkan data pengguna termasuk role di response. Jika email yang dimasukkan tidak terdaftar, sistem memberikan pesan error "User not found". Apabila password yang dimasukkan salah, sistem memberikan pesan error "Wrong password".

c. Manajemen Kamar

Tabel 4. Pengujian *Blaxbox* Manajemen Kamar

Kasus Uji	Pengujian yang dilakukan	Hasil pengujian	Kesimpulan
Medapatkan daftar semua kamar (<i>getRoom</i>)	Memanggil endpoint 'GET'	Mendapatkan response daftar semua kamar	Berhasil
Mendapatkan detail kamar berdasarkan ID (<i>getRoomById</i>)	Memanggil endpoint 'GET/room/:id'	Mendapatkan response dengan detail kamar	Berhasil
Menambahkan kamar baru (<i>addRoom</i>)	Mengirimkan data form termasuk gambar valid	Menerima pesan "Room Created Successfully"	Berhasil
Mengedit kamar yang ada (<i>editRoom</i>)	Mengirimkan data form termasuk gambar opsional	Menerima pesan "Room Update Successfully"	Berhasil
Menghapus kamar yang ada (<i>deleteRoom</i>)	Mengirimkan permintaan 'Delete/room/:id'	Menerima pesan "Room Delete Successfully"	Berhasil

Berdasarkan tabel 3, Semua skenario pengujian berhasil dengan respons yang sesuai yang menunjukkan fungsi dasar sistem bekerja dengan baik. Berdasarkan hasil pengujian *blackbox* ini, fitur manajemen kamar telah lulus pengujian dengan baik. Fungsionalitas seperti menampilkan daftar kamar, mengambil detail kamar berdasarkan ID, menambahkan kamar baru dengan validasi gambar, mengedit kamar dengan penggantian gambar opsional, dan menghapus kamar bersamaan dengan gambar terkait berjalan dengan baik.

d. ApplyRent (Ajukan Sewa)

Tabel 4. Pengujian *Blackbox* Ajukan Sewa

Kasus Uji	Pengujian yang dilakukan	Hasil Pengujian	Kesimpulan
Melihat semua data kamar kos (<i>getApplyRent</i>)	Memanggil endpoint <i>'GET/applyrent'</i>	Mendapatkan respons dengan daftar semua kamar kos	Berhasil
Mendapatkan detail sewa kamar berdasarkan ID (<i>getApplyRent</i>)	Memanggil endpoint <i>'GET/applyrent/:id'</i>	Mendapatkan respons dengan detail kamar kos	Berhasil
Mengajukan sewa kamar kos baru (<i>addapplyRent</i>)	Mengirimkan data form termasuk foto KTP penyewa	Menerima pesan <i>"Apply Rent Successfully"</i>	Berhasil
Menghapus atau membatalkan kamar kos yang telah di ajukan	Mengirimkan permintaan <i>'DELETE/applyrent/:id'</i>	Menerima pesan <i>"ApplyRent Delete Successfully"</i>	Berhasil

Berdasarkan tabel 4, Semua skenario pengujian berhasil dilalui dengan respons yang sesuai, menunjukkan bahwa fitur "Ajukan Sewa Kamar Kos" dapat melakukan operasi dasar seperti menampilkan, menambah, dan menghapus atau membatalkan sewa kamar kos dengan benar. Berdasarkan hasil pengujian *blackbox* ini, fitur "Ajukan Sewa Kamar Kos" pada aplikasi manajemen kos-kosan berbasis web menggunakan Express.js dengan database MySQL telah lulus pengujian dengan baik. Fungsionalitas dasar seperti CRUD (*Create, Read, Update, Delete*) aplikasi sewa kamar kos dan pengelolaan gambar terkait berjalan sesuai yang diharapkan.

e. Transaksi

Tabel 5. Pengujian *Blackbox* Transaksi

Kasus Uji	Pengujian yang Dilakukan	Hasil pengujian	Kesimpulan
Mendapatkan semua transaksi (<i>getTransaction</i>)	Memanggil endpoint <i>'GET/transaction'</i>	Mendapatkan respons dengan daftar semua transaksi	Berhasil
Mendapatkan detail transaksi berdasarkan ID	Memanggil endpoint <i>'GET/transaction/:id'</i>	Mendapatkan respons dengan detail transaksi	Berhasil
Menambahkan transaksi baru (<i>addTransaction</i>)	Mengirimkan data form termasuk gambar valid (bukti pembayaran untuk user)	Menerima pesan <i>"Transaction Successfully"</i>	Berhasil
Menghapus transaksi (<i>deleteTransaction</i>)	Mengirimkan permintaan <i>'DELETE/transaction/:id'</i>	Menerima pesan <i>"Transaction Deleted Successfully"</i>	Berhasil

Berdasarkan tabel 5, Semua skenario pengujian berhasil dilalui dengan respons yang sesuai, menunjukkan bahwa fitur "Transaksi" dapat melakukan operasi dasar seperti menampilkan, menambah, dan menghapus transaksi dengan benar. Hasil pengujian *blackbox* ini, fitur "Transaksi" pada aplikasi manajemen kos-kosan berbasis web menggunakan Express.js dengan database MySQL telah lulus pengujian dengan baik. Fungsionalitas dasar seperti CRUD (*Create, Read, Update, Delete*) transaksi dan pengelolaan gambar terkait berjalan sesuai yang diharapkan.

Secara keseluruhan semua fitur utama aplikasi manajemen kos-kosan telah lulus pengujian *blackbox*, menunjukkan bahwa sistem berfungsi sesuai dengan kebutuhan yang telah di tentukan secara 100%

4. Kesimpulan

Dengan adanya aplikasi manajemen kos-kosan berbasis web ini, dapat memberikan solusi yang signifikan bagi para pencari kos. Para pencari kos tidak lagi mengalami kesulitan dalam mencari informasi atau melakukan transaksi untuk mendapatkan kos-kosan yang sesuai. Fitur-fitur seperti manajemen kamar, ajukan sewa, dan transaksi berfungsi dengan baik, memberikan pengalaman yang lebih mudah, efisien, dan terorganisir bagi pengguna. Hal ini menggambarkan bahwa aplikasi berbasis web ini tidak hanya meningkatkan kualitas layanan dalam manajemen kos-kosan, tetapi juga memberikan solusi praktis bagi masalah yang sering dihadapi oleh para pencari kos.

5. Referensi

- [1] W. Darlin, A. Dwi Putra, N. Hendrastuty, N. Penulis, K.: Wayan, and D. Submitted, "Sistem Informasi Manajemen Kost Putra Trisula Berbasis Web (Studi Kasus: Asrama Putra Trisula)," *J. Teknol. dan Sist. Inf.*, vol. 4, no. 3, pp. 240–249, 2023.
- [2] S. Rahmadani, I. Gartina Husein, and W. Wikusna, "Aplikasi Pencarian Dan Pengelolaan Kos Di Daerah Universitas Telkom Modul Pemilik Kos," *eProceedings Appl. Sci.*, vol. 7, no. 6, pp. 2607–2612, 2021.
- [3] Nugroho, "Metode Extreme Programming Dalam Membangun Aplikasi Kos-Kosan Di Kota Bandar Lampung Berbasis Web," *Cendikia*, vol. XVIII, no. 2013, p. 4, 2009.
- [4] N. D. Putra, B. Dirgantoro, and R. A. Nugrahaeni, "Perancangan Back-End Aplikasi E-Kost Dengan Model Waterfall Berbasis Web," *e-Proceeding Eng.*, vol. 7, no. 2, pp. 4814–4821, 2020.
- [5] D. A. Muhammad Syah Reza Pahlevi, Bima Fadi Lana, "Merancang Dan Membangun Aplikasi Pencarian Kos Berbasis Web dan Android," vol. 2016, no. September, pp. 60–69, 2005.
- [6] M. Aldi, A. Ghiffari, I. B. Praptono, and H. Sagita, "Perancangan Sistem Informasi Manajemen Pengelolaan Hunian Sewa Berbasis Website Pada Kos-Kosan Pondok Salma," *Telkom Univ. Open Libr.*, vol. 7, no. 2, pp. 6106–6113, 2020.
- [7] H. Handayani, K. U. Faizah, A. Mutiara Ayulya, M. F. Rozan, D. Wulan, and M. L. Hamzah, "Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development Designing a Web-Based Inventory Information System Using the Agile Software Development Method," *J. Test. dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 29–40, 2023.
- [8] R. Lisgiani and S. Nurmajid, "Implementasi Autentikasi Dari Sisi Backend Pada Arsitektur Microservices Menggunakan Express Js," *Infotronik J. Teknol. Inf. dan Elektron.*, vol. 7, no. 1, p. 27, 2022, doi: 10.32897/infotronik.2022.7.1.1330.
- [9] C. Nizar, "Rancang Bangun Sistem Informasi Sewa Rumah Kost (E-Kost) Berbasis Website," *J. Sist. Inf. dan Sains Teknol.*, vol. 3, no. 1, pp. 1–10, 2021, doi: 10.31326/sistek.v3i1.852.
- [10] M. N. Saiholau, "Rancang Bangun Backend Website Pemungutan Suara Dengan Menggunakan Framework Express.js," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 1433–1442, 2024, doi: 10.23960/jitet.v12i2.4261.