



Integrasi Google OAuth dan Payment Gateway Midtrans dalam KosConnect

Fathya Fathimah Azzahra¹, Balqis Rosa Sekamayang², Roni Habibi³

^{1,2,3}Program Studi Sarjana Terapan Teknik Informatika, Sekolah Vokasi, Universitas Logistik & Bisnis Internasional, Bandung.

*Penulis Korespondensi, Email: fathyafathazzra@gmail.com

Abstrak- KosConnect adalah aplikasi inovatif yang memfasilitasi pencarian, pemesanan, dan pembayaran kos secara digital. Dengan meningkatnya permintaan kos-kosan serta tantangan dalam akses informasi yang efisien, KosConnect hadir dengan mengintegrasikan teknologi modern seperti Google OAuth dan Midtrans untuk meningkatkan pengalaman pengguna. Google OAuth digunakan sebagai metode autentikasi Single Sign-On (SSO) yang lebih praktis, sementara Midtrans sebagai payment gateway mendukung transaksi digital yang aman dan cepat. Aplikasi ini dikembangkan menggunakan bahasa pemrograman Go dan basis data MongoDB untuk memastikan performa optimal dalam pengelolaan data. Selain itu, KosConnect menawarkan fitur personalisasi yang memungkinkan pengguna menemukan kos sesuai preferensi mereka. Proyek ini bertujuan untuk menyederhanakan proses login, registrasi, dan pembayaran dengan menyediakan layanan terintegrasi yang mencakup seluruh tahapan, mulai dari pencarian hingga transaksi akhir. Hasil pengembangan menunjukkan bahwa KosConnect mampu mengatasi keterbatasan sistem informasi kos sebelumnya, seperti autentikasi berbasis email dan kata sandi tanpa SSO, pembayaran transfer bank yang memerlukan unggah bukti manual, serta kurangnya fitur personalisasi.

Kata Kunci: Kos; Google OAuth; Midtrans; Go; MongoDB.

Abstract- KosConnect is an innovative application that facilitates boarding house search, booking, and payment by integrating modern technologies to enhance user experience. As the demand for boarding houses increases and access to reliable information remains a challenge, KosConnect leverages Google OAuth for Single Sign-On (SSO) authentication, providing a more efficient and secure login process, while Midtrans serves as a payment gateway to enable fast and secure digital transactions. Developed using the Go programming language and MongoDB database, the application ensures optimal performance in data management while also offering personalization features to help users find accommodations that match their preferences. This project aims to streamline the login, registration, and payment processes by providing an integrated service that covers all stages from searching for boarding houses to completing transactions. The development results show that KosConnect effectively addresses the limitations of previous boarding house information systems, such as authentication that relies solely on email and password without SSO, bank transfers requiring manual proof uploads, and the lack of personalization features.

Keywords: Boarding House; Google OAuth; Midtrans; Go; MongoDB.

1. PENDAHULUAN

Sektor penyewaan kos-kosan merupakan salah satu bidang yang memiliki potensi ekonomi signifikan dengan permintaan yang terus meningkat[1][2]. Pada tahun 2022, Pemerintah Kota Bandung mencatat pertumbuhan pendapatan pajak sebesar Rp 2,5 miliar dari kategori kos-kosan, mencerminkan kontribusi besar sektor ini terhadap perekonomian[3][4].

Namun, kendala dalam proses pemesanan kos masih menjadi permasalahan, terutama bagi mereka yang tinggal di luar kota atau daerah yang jauh dari tempat kos yang diinginkan[5]. Metode konvensional, di mana calon penyewa harus datang langsung atau menghubungi pemilik kos melalui telepon, menjadi hambatan dalam mengakses informasi dan melakukan pemesanan secara efisien[6][7][8].

Para pencari kos seringkali menghadapi kesulitan dalam mencari informasi yang sesuai dengan preferensi mereka[9][10][11] karena akses informasi yang terbatas. Mereka cenderung mengandalkan informasi yang didapat dari warga sekitar atau kerabat yang tinggal di wilayah tersebut[7][10] yang seringkali tidak mencakup semua opsi yang tersedia. Oleh karena itu, dibutuhkan solusi yang lebih efisien dan terkoneksi untuk menyediakan informasi dan memudahkan proses pemesanan kos-kosan.

Aplikasi berbasis web telah menjadi solusi potensial untuk menjawab permasalahan tersebut. Meskipun demikian, banyak sistem informasi kos yang ada masih menggunakan metode autentikasi berbasis email dan kata sandi tanpa integrasi teknologi modern, seperti *Single Sign-On* (SSO). Proses ini sering kali memerlukan verifikasi tambahan melalui email, yang dapat memakan waktu lebih lama dibandingkan metode autentikasi yang lebih

praktis dan efisien. Hal ini berpotensi mengurangi kenyamanan pengguna dalam proses registrasi dan akses layanan.

Menurut penelitian oleh Chalidazia Nizar, sistem e-kost berbasis web memberikan solusi untuk mempermudah promosi dan pemesanan kamar kost secara online[5]. Di sisi lain, pembayaran dalam sistem ini masih bergantung pada transfer bank dan memerlukan unggah bukti pembayaran untuk konfirmasi. Penelitian oleh Novrini Hasti dan Muhammad Inal Zaelani menunjukkan bahwa sistem informasi rumah kost mempermudah penghuni dalam pembayaran dan penyampaian keluhan[12]. Meskipun demikian, proses pembayaran juga menggunakan metode transfer bank dengan mekanisme unggah bukti transfer. Penelitian Agung Rahmatillah, Tomi Saputra, Wuwuh Bakti Hartiningsih mengungkapkan bahwa sistem berbasis web mereka memungkinkan penyewaan secara online tanpa harus datang langsung[13]. Walaupun begitu, pembayaran dalam sistem tersebut masih menggunakan transfer bank tanpa mekanisme unggah bukti.

Di sisi lain, penelitian oleh Muhamad Iqbal Baehaqi dan Taryana Suryana menggunakan *Midtrans* sebagai solusi pembayaran *online* dalam aplikasi mereka. Hal ini memungkinkan transaksi yang lebih cepat dan akurat[14]. Namun, proses registrasi dalam sistem tersebut masih menggunakan *input* manual, di mana pengguna harus mengetikkan nama, email, dan kata sandi secara langsung tanpa opsi *Single Sign-On* (SSO) seperti *Google*.

Penelitian Ricky Sepjo, Muhammad Ami dan Heri Kurniawan menunjukkan bahwa sistem informasi berbasis web dapat meningkatkan efisiensi pengelolaan properti[15]. Sebaliknya, *login* dan registrasi dalam sistem mereka hanya tersedia dalam bentuk *input* manual tanpa dukungan teknologi modern seperti *Single Sign-On*. Selain itu, meskipun beberapa penelitian menyebutkan kemudahan pembayaran, metode spesifik yang digunakan tidak dijelaskan secara detail, sehingga implementasinya kurang jelas.

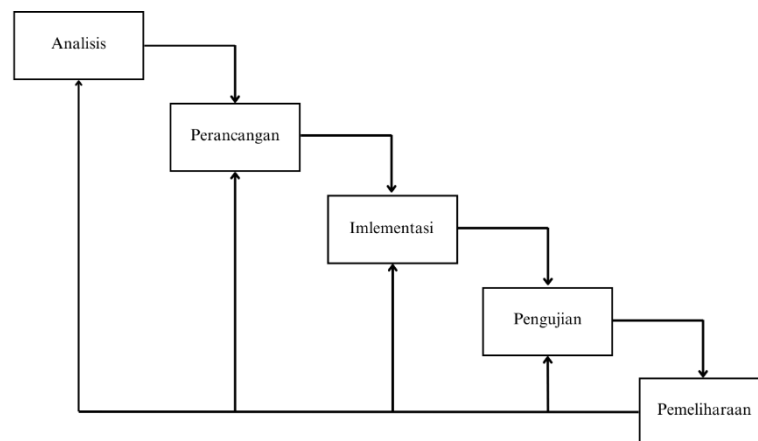
KosConnect hadir untuk mengatasi keterbatasan tersebut dengan mengadopsi teknologi modern seperti *Google OAuth* dan *Midtrans*. Integrasi *Google OAuth* memungkinkan proses *login* dan registrasi yang lebih cepat serta aman melalui *Single Sign-On*. Sementara itu, *Midtrans* sebagai *payment gateway* mendukung transaksi digital yang lebih efisien, aman, dan mudah diakses. Dengan pendekatan ini, KosConnect menawarkan solusi yang lebih komprehensif untuk kebutuhan penyewa dan pemilik kos.

2. METODOLOGI PENELITIAN

2.1 Metode dan Tahapan Pengembangan Sistem

Metode penelitian merupakan metode eksperimen yang menggunakan metode sistem pendukung keputusan dalam kegiatannya[16]. Penelitian ini menggunakan metode *waterfall* dalam pengembangan sistem, yang merupakan salah satu metode pengembangan perangkat lunak dengan alur kerja yang terstruktur dan sistematis [17]. Gambar

1 berikut menyajikan alur metode *waterfall* yang diterapkan dalam penelitian ini:



Gambar 1. Alur Metode *Waterfall*.

Metode ini dipilih karena memungkinkan pengembangan sistem dilakukan secara bertahap dan terorganisir dari awal hingga akhir. Tahapan metode *waterfall* yang diterapkan dalam penelitian ini meliputi:

a. Analisis

Tahapan ini bertujuan untuk memahami kebutuhan sistem secara mendalam. Data kebutuhan diperoleh melalui studi literatur dan observasi terhadap sistem informasi kos yang sudah ada untuk menemukan

kekurangan yang dapat diperbaiki. Pada tahap ini, kebutuhan seperti integrasi *Google OAuth* dan *Midtrans*, serta fitur pencarian kos, didefinisikan dengan jelas.

b. Perancangan

Pada tahap ini, rancangan sistem dibuat untuk memberikan gambaran yang jelas mengenai alur kerja sistem. Diagram seperti *use case*, *activity*, *sequence*, dan *class diagram* digunakan untuk memodelkan sistem, sementara desain antarmuka pengguna dirancang untuk memberikan pengalaman pengguna yang optimal. Setelah perancangan selesai, sistem dikembangkan berdasarkan rancangan yang telah dibuat.

c. Implementasi

Teknologi yang digunakan meliputi bahasa pemrograman *Go* untuk pengembangan *backend*, *MongoDB* untuk manajemen basis data, dan *Vercel* untuk *deployment* aplikasi. Selain itu, integrasi *Google OAuth* dan *Midtrans* API diterapkan untuk memenuhi kebutuhan autentikasi dan pembayaran digital.

d. Pengujian

Sistem diuji untuk memastikan bahwa semua fitur berjalan sesuai spesifikasi. Pengujian dilakukan dengan metode *black-box testing* untuk menguji fungsionalitas fitur utama, serta pengujian *backend* menggunakan Postman untuk memverifikasi respons setiap *endpoint* API. *Endpoint* yang diuji mencakup fitur seperti autentikasi menggunakan *Google OAuth*, proses transaksi dengan *Midtrans*, manajemen data kos, dan manajemen data pengguna. Hasil pengujian menunjukkan bahwa sistem berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan.

e. Pemeliharaan

Tahapan terakhir adalah pemeliharaan, di mana sistem dipantau secara berkala untuk memastikan stabilitasnya. Jika ditemukan kesalahan atau kebutuhan tambahan, perbaikan dilakukan untuk menjaga kualitas sistem. Secara keseluruhan, metode *waterfall* dalam penelitian ini mendukung proses pengembangan sistem yang terstruktur, sehingga hasilnya mampu memenuhi kebutuhan pengguna dalam hal autentikasi, transaksi, dan manajemen data secara efektif.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Analisis

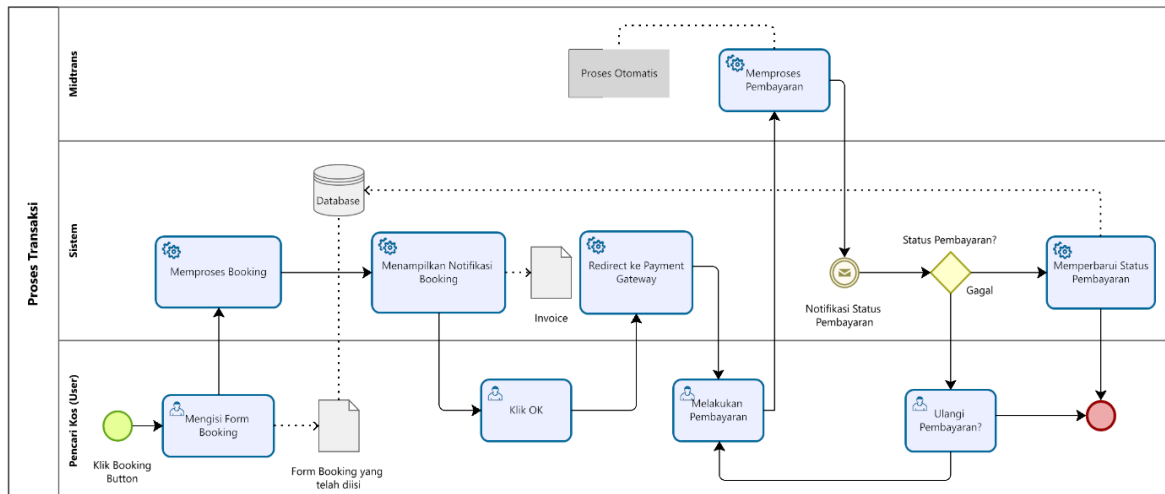
Berdasarkan hasil analisis literatur yang telah dilakukan, dapat disimpulkan bahwa melalui integrasi *Google OAuth* untuk autentikasi dan *Midtrans* sebagai sistem pembayaran, pendekatan ini mampu meningkatkan keamanan, kenyamanan, dan efisiensi dalam penggunaan sistem dibandingkan metode sebelumnya. Hasil analisis dapat dilihat pada Table 1.

Tabel 1. Hasil Analisis.

Penelitian	Judul	<i>Google OAuth</i>	<i>Midtrans</i>
[5]	Rancang Bangun Sistem Informasi Sewa Rumah Kost (E-Kost) Berbasis <i>Website</i>	×	×
[12]	Sistem Informasi Pengelolaan Rumah Kost Budi Sari Berbasis <i>Website</i>	×	×
[13]	Sistem Informasi Rumah Kost Berbasis Web Studi Kasus pada Paviliun Sejahtera	×	×
[14]	Penerapan Teknologi <i>Geofence</i> pada Aplikasi Pemesanan Kamar Kosan	×	✓
[15]	Rancang Bangun Sistem Sewa Rumah Kontrakan Berbasis Web	×	×
Proposed method	KosConnect dengan Metode <i>Google OAuth</i> dan <i>Payment Gateway Midtrans</i>	✓	✓

3.2 Diagram BPMN

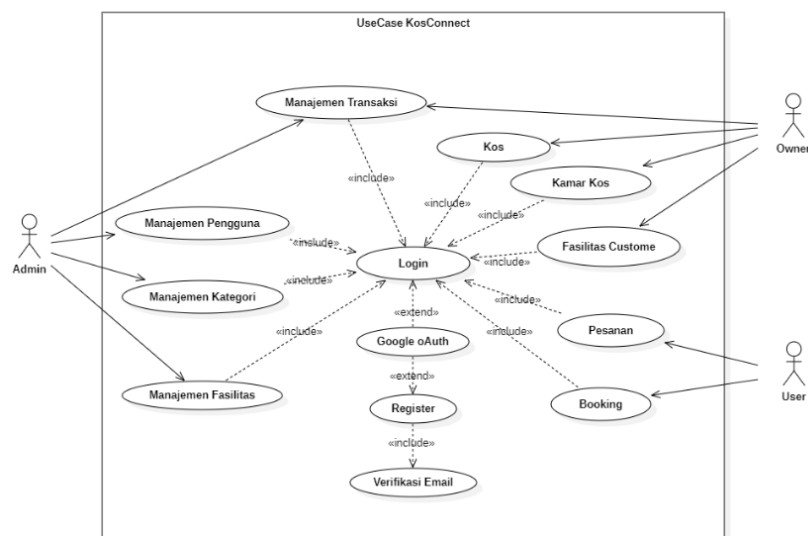
BPMN adalah standar pemodelan bisnis berbasis *flowchart* yang memvisualisasikan alur kerja, aktivitas, dan kontrol proses secara terstruktur, serta didukung oleh berbagai alat pemodelan[18]. Diagram BPMN pada Gambar 2 ini menggambarkan alur utama transaksi yang digunakan dalam sistem ini. Diagram ini menunjukkan proses utama yang dialami oleh pengguna, mulai dari pencarian kos, pemesanan kos, hingga pembayaran melalui *Midtrans*.



Gambar 2. Diagram BPMN.

3.3 Use Case Diagram

Use Case merupakan salah satu *tools* yang digunakan untuk membuat pemodelan interaksi *user* dengan sistem [19]. Tujuan dari pemodelan *use case* adalah untuk mendefinisikan kebutuhan fungsional dan operasional sistem dengan menentukan skenario penggunaan yang disepakati antara pengguna dan pengembang.[20] Diagram ini menunjukkan tiga aktor utama yaitu Pencari Kos(User) yang dapat melakukan pencarian kos, memesan kamar, dan melakukan pembayaran. Pemilik Kos(Owner) dapat mengelola data kos, kamar kos dan fasilitas *custom*. Admin dapat memanajemen transaksi, manajemen pengguna, manajemen kategori dan manajemen fasilitas. Gambar 3 menunjukkan diagram *use case* yang menggambarkan interaksi pengguna dengan sistem.



Gambar 3. Use Case Diagram.

3.3 Tools yang Digunakan

Dalam penelitian ini, penggunaan *tools* didasarkan pada kebutuhan untuk meningkatkan efisiensi, keamanan, dan kinerja sistem. *Google OAuth* digunakan sebagai solusi autentikasi *Single Sign-On* untuk mempermudah dan mengamankan proses login pengguna, sementara *Midtrans* diintegrasikan sebagai *payment gateway* untuk mendukung sistem pembayaran yang lebih andal dan efisien. Pemilihan kedua *tools* ini dilakukan berdasarkan analisis terhadap metode sebelumnya yang memiliki keterbatasan dalam implementasi autentikasi dan sistem pembayaran.

3.3.1 Go Language untuk Backend Development

Go Language atau *Golang* adalah bahasa pemrograman yang dikembangkan oleh *Google* pada tahun 2007 dan dirilis pada 2009 oleh Ken Thompson, Robert Griesemer, dan Rob Pike. Bahasa ini dirancang agar efisien, sederhana, dan memiliki performa tinggi seperti *C* dan *C++*, dengan tipe data statis serta kemampuan kompilasi langsung ke kode biner [21]. Dengan sintaks yang sederhana dan kecepatan kompilasi tinggi, *Golang* mengatasi tantangan seperti kompilasi lambat dan kompleksitas kode yang sering ditemukan dalam bahasa seperti *Java* atau *C++* [22]. Selain itu, *Golang* memiliki keunggulan dalam dukungan terhadap *concurrency*, yang memungkinkan beberapa proses berjalan secara paralel dalam satu waktu, serta dilengkapi dengan *Garbage Collector* untuk mengelola penggunaan memori agar lebih optimal dan efisien [23].

3.3.2 MongoDB sebagai Basis Data

MongoDB adalah basis data *NoSQL* yang bersifat *document-based*, yang berarti tidak memiliki tabel, kolom, atau baris seperti basis data relasional. Data disimpan dalam bentuk *dokumen JSON* yang disebut *BSN* (Binary JSON), memungkinkan fleksibilitas dalam penyimpanan data yang tidak terstruktur. Versi terbaru *MongoDB* telah meningkatkan performa dengan *throughput* baca hingga 36% lebih cepat serta fitur *Queryable Encryption* untuk keamanan data [24].

3.3.3 Autentikasi dengan Google OAuth

OAuth 2.0 adalah protokol otorisasi yang memungkinkan aplikasi pihak ketiga mengakses sumber daya pengguna tanpa membagikan kredensial [25][26]. Proses ini menggunakan *access token* dan mendukung berbagai mekanisme otorisasi seperti *Authorization Code Flow* [25].

Google OAuth merupakan implementasi *OAuth 2.0* yang memberikan akses terbatas ke layanan *Google*, seperti *Gmail* dan *Google Cloud Platform (GCP)*. Dengan *Google OAuth*, pengguna dapat *login* menggunakan akun *Google* mereka, sehingga proses autentikasi menjadi lebih mudah tanpa perlu membuat akun baru. Selain itu, *Google OAuth* mendukung atribut profil dasar dan tambahan dengan izin eksplisit serta memiliki proses verifikasi ketat untuk menjamin keamanan data pengguna [27].

3.3.4 Sistem pembayaran dengan Midtrans

Midtrans adalah *platform payment gateway* yang menghubungkan transaksi antara pembeli dan penjual [28]. *Platform* ini dapat diintegrasikan dengan *e-commerce* dan mendukung berbagai metode pembayaran, termasuk kartu debit, kartu kredit, serta layanan penarikan dan pengiriman uang tunai [28][29].

Dengan menggunakan *Midtrans*, sistem dapat menangani pembayaran secara lebih aman dan efisien, mengurangi risiko kesalahan dalam transaksi, serta memberikan pengalaman pembayaran yang lebih nyaman bagi pengguna.

3.4 Implementasi Aplikasi

KosConnect dikembangkan sebagai aplikasi berbasis *website* dengan tiga jenis pengguna utama:

- Admin: Mengelola data pengguna, transaksi, dan kategori kos serta melihat data kos, data kamar kos dan fasilitas *custom*.
- Owner: Menambahkan dan mengelola data kos dan kamar kos serta fasilitas *custom* yang ditawarkan.
- User: Mencari, memesan, melakukan pembayaran kos dan melihat *history* pemesanan.

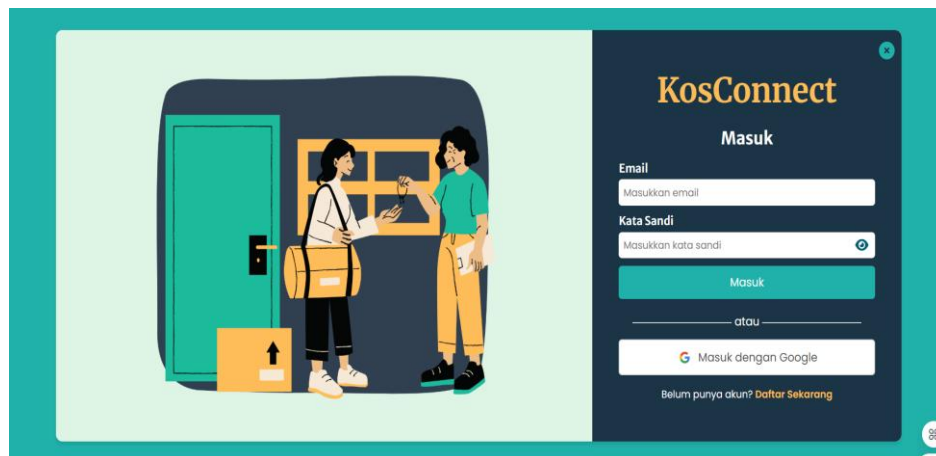
Bagian berikut menampilkan hasil implementasi KosConnect, termasuk tampilan dan fitur utama dari setiap peran pengguna dalam sistem.

3.4.1 Halaman Login

Halaman *login* yang ditunjukkan pada gambar 4 berfungsi untuk mengautentikasi pengguna yang telah terdaftar agar dapat mengakses sistem. Formulir *login* terdiri dari kolom *input* untuk email dan kata sandi, dengan opsi menampilkan atau menyembunyikan karakter demi kenyamanan pengguna.

Selain itu, sistem menyediakan opsi *login* dengan *Google OAuth*, memungkinkan pengguna masuk menggunakan akun *Google* mereka secara cepat dan praktis. Setelah kredensial diverifikasi, sistem memberikan akses sesuai

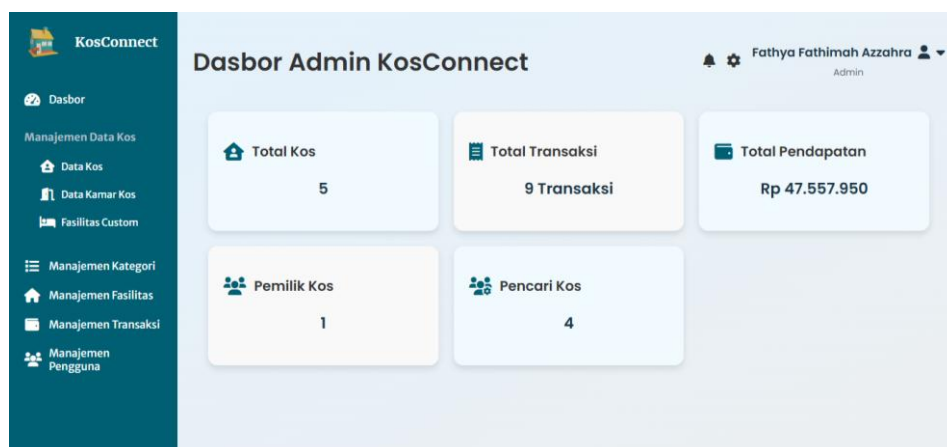
peran pengguna yaitu Pemilik Kos (Owner), atau Pencari Kos (User)



Gambar 4. Halaman *Login*.

3.4.2 Halaman Dashboard Admin

Halaman *dashboard* admin pada gambar 5 berfungsi sebagai pusat kendali utama untuk mengelola sistem.



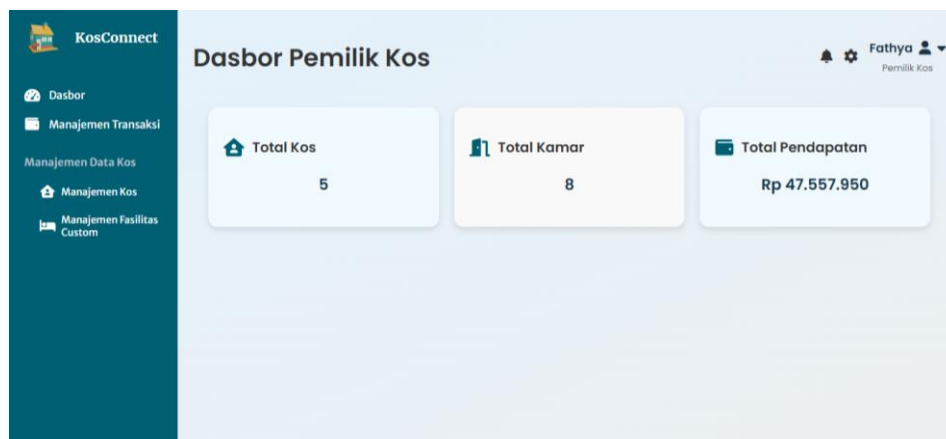
Gambar 5. Halaman *Dashboard* Admin.

Dashboard menyajikan ringkasan informasi dalam bentuk *widget*, termasuk total kos, transaksi, pendapatan, pemilik kos, dan pencari kos. Pada panel navigasi, terdapat beberapa fitur utama, dimulai dari Dasbor, kemudian diikuti oleh manajemen data kos yang mencakup data kos, data kamar kos, dan fasilitas *custom* di mana ketiga menu ini hanya memungkinkan admin untuk melihat data tanpa melakukan perubahan. Selanjutnya, terdapat fitur manajemen kategori, manajemen fasilitas, manajemen transaksi, dan manajemen pengguna, yang dilengkapi dengan fungsi CRUD (Create, Read, Update, Delete) untuk menambah, mengubah, dan menghapus data sesuai kebutuhan.

3.4.3 Halaman Dashboard Owner/Pemilik kos

Halaman *Dashboard Owner* berfungsi sebagai pusat kendali bagi pemilik kos dalam mengelola properti mereka. *Dashboard* menyajikan informasi penting seperti total kos, jumlah kamar, dan transaksi untuk membantu pemilik memantau bisnisnya secara cepat.

Pada panel navigasi, terdapat beberapa fitur utama, mulai dari Dasbor, manajemen transaksi yang hanya memungkinkan pemilik kos untuk melihat data tanpa melakukan perubahan, serta manajemen data kos, yang mencakup pengelolaan kos dan fasilitas *custom*. Setiap kos yang terdaftar juga memiliki manajemen data kamar, dilengkapi fitur CRUD (Create, Read, Update, Delete) untuk menambah, mengubah, dan menghapus data kamar sesuai kebutuhan. Tampilan halaman *dashboard owner* dapat dilihat pada Gambar 6.



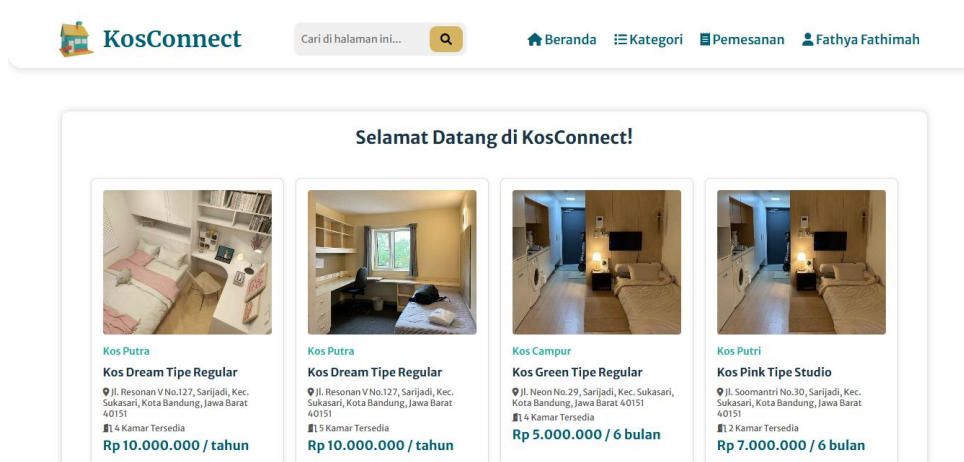
Gambar 6. Halaman *Dashboard Owner*.

3.4.4 Halaman *Home KosConnect*

Halaman *Home* KosConnect berfungsi sebagai pusat informasi dan navigasi bagi pengguna dalam mencari kos yang sesuai dengan kebutuhan mereka. Tampilan halaman ini bersifat dinamis, menyesuaikan dengan status pengguna, di mana navbar akan berbeda sebelum dan sesudah *login*.

Pada bagian atas, terdapat kolom pencarian yang memungkinkan pengguna mencari kos berdasarkan nama atau lokasi tertentu. *Navbar* juga menyediakan akses cepat ke berbagai halaman utama, seperti Beranda, Kategori, Pemesanan, serta profil pengguna bagi mereka yang telah login.

Bagian utama halaman menampilkan daftar kos yang tersedia, lengkap dengan gambar, tipe kos, alamat, jumlah kamar tersedia, serta harga sewa. Informasi ini membantu pengguna dalam membandingkan pilihan sebelum mengakses halaman detail kos untuk informasi lebih lanjut. Tampilan halaman home KosConnect dapat dilihat pada Gambar 7.



Gambar 7. Halaman *Home KosConnect*.

3.4.5 Halaman *Transaksi User*

Halaman *Transaksi User* merupakan bagian penting dalam proses pemesanan kamar kos di aplikasi KosConnect. Halaman ini memungkinkan calon penyewa untuk mengajukan pemesanan kamar setelah memilih kos yang diinginkan.

Sebelum masuk ke halaman transaksi, pengguna telah menelusuri daftar kos yang tersedia, memilih salah satu kamar kos sesuai dengan kebutuhannya. Setelah itu, pengguna diarahkan ke halaman transaksi untuk mengisi formulir pemesanan. Formulir ini mengumpulkan berbagai informasi penting, seperti nama lengkap, email, dan nomor WhatsApp yang digunakan untuk komunikasi dengan pemilik kos. Pengguna juga harus memilih tanggal masuk sesuai dengan rencana mereka, serta memiliki opsi untuk menambahkan fasilitas tambahan jika tersedia.

Setelah semua data diisi, sistem akan menampilkan rincian pesanan yang mencakup detail kamar yang dipesan, tanggal masuk yang dipilih, serta total biaya yang harus dibayarkan. Informasi ini ditampilkan untuk memastikan bahwa pengguna telah mengisi data dengan benar sebelum mengajukan permohonan sewa. Jika semua

sudah sesuai, pengguna dapat menekan tombol "Ajukan Sewa Sekarang" untuk melanjutkan proses pemesanan. Saat tombol diklik, sistem akan menampilkan notifikasi konfirmasi yang memberi tahu apakah pemesanan berhasil diproses atau tidak. Jika berhasil, notifikasi akan menampilkan tombol "OK", yang ketika diklik akan mengarahkan pengguna ke halaman pembayaran *Midtrans*. Jika terjadi kesalahan, notifikasi akan memberikan informasi bahwa transaksi gagal dan pengguna dapat mencoba kembali. Tampilan halaman transaksi dapat dilihat pada Gambar 8 di bawah ini.

Gambar 8. Halaman Transaksi *User*.

3.4 Pengujian Sistem

Pengujian merupakan proses yang dilakukan untuk mencari kesalahan atau *error* dengan menjalankan program[30]. Pengujian sistem dilakukan untuk memastikan bahwa semua fitur yang ada berjalan sesuai dengan rencana dan dapat memenuhi kebutuhan pengguna. Beberapa aspek yang diuji meliputi autentikasi pengguna, proses pemesanan, serta transaksi pembayaran.

3.4.1 Hasil Pengujian Black-box testing

Metode *black-box testing* adalah pengujian yang melihat hasil eksekusi melalui data uji dan memastikan fungsi dari *software* [31]. *Black-box testing* dilakukan untuk memastikan setiap fitur dalam aplikasi berjalan sesuai fungsinya. Pengujian ini dilakukan untuk menilai keandalan dan efektivitas setiap fitur dalam sistem, serta mengidentifikasi potensi kesalahan atau kekurangan dalam implementasinya. Berikut adalah hasil pengujian dari berbagai *use case* yang diuji:

Table 2. Pengujian *Black-box testing*.

No	No.Uji	Proses	In	Output	Hasil Uji
1.	UC.01	Register	Fullname : Fathya Azzahra Email : fathyafathazzra@gmail.com Role : Pilih Password : akukamusatu	Error, 'Please select an item in the list'	Tidak Berhasil
			Full name : Fathya Fathimah Azzahra Email : fathyafathazzra@gmail.com Role : Pencari kos(user) Password : akukamusatu	Halaman Login	Berhasil
2.	UC.02	Login	Email : fathyafathazzra@gmail.com Password : akukamusatu	Halaman Home	Berhasil

			Email : balqisrosa@gmail.com Password : akukamusatu	Error, 'Silakan cek email Anda untuk verifikasi sebelum login.'	Tidak Berhasil
3.	UC.03	Login Google OAuth	Klik Button "Masuk dengan Google"	Halaman Utama	Berhasil
			Klik Button "Masuk dengan Google" lalu Menolakan Izin	Halaman Callback Error	Gagal
			Klik Button "Masuk dengan Google" lalu Memberikan Izin	Halaman Select Role	Berhasil
4.	UC.04	Manajemen Pengguna	Menghapus User	Data Pengguna Terhapus	Berhasil
5.	UC.05	Kategori	Nama Kategori: Kos Putri	Halaman Data Kategori	Berhasil
6.	UC.06	Fasilitas	Nama Fasilitas: Parkir Motor Tipe Fasilitas: boarding_house	Halaman Data Fasilitas	Berhasil
7.	UC.07	Manajemen Kos	Nama Kos : Kos Pink Kategori: Kos Putri Alamat: Jl. Soomantri No.30, Sarijadi, Kec. Sukasari, Kota Bandung, Jawa Barat 40151 Deskripsi: Terdapat petugas kebersihan yang bertugas seminggu sekali Peraturan: Tidak Boleh Membawa Teman Laki-Laki Menginap Fasilitas Kos: - Dapur Umum - Kamar Mandi Luar - Parkir Motor	Halaman Data Kos	Berhasil
8.	UC.08	Manajemen Kamar Kos	Nama Kos: Kos Pink Tipe Kamar: Studio Ukuran: 5 x 5 meter Harga Sewa: Rp 7.000.000 / 6 bulan Rp 13.000.000 / tahun Fasilitas Kamar Kos: - Kamar Mandi Dalam - Meja Belajar Lesehan - Cermin Berdiri Fasilitas Custom: - 4 Galon Air Aqua Tiap Bulan – Rp 100.000 - Lemari Rp 100.000 Jumlah Kamar Tersedia: 5	Halaman Data Kamar Kos	Berhasil
9.	UC.09	Fasilitas Custom	Nama: Lemari Harga: Rp 100.000	Halaman Data Fasilitas Custom	Berhasil
10	UC.10	Booking	Nama Pemesan: Fathya Fathimah Azzahra Whatsapp: +621234567890 Tanggal Pindah: 08/02/2025 Fasilitas Custom: 4 Galon Air Tiap Bulan – Rp 100.000 Lemari Rp 100.000 Harga Sewa: Rp 5000.000 / Per 6 Bulan	Halaman Pembayaran Midtrans	Berhasil

Hasil pengujian menunjukkan bahwa sebagian besar fitur dalam sistem telah berfungsi sesuai dengan harapan, dengan beberapa skenario yang gagal dikarenakan kesalahan input atau ketidaksesuaian dengan aturan sistem. Pengujian ini memastikan bahwa sistem mampu menangani berbagai kasus penggunaan secara optimal.

3.4.2 Hasil Pengujian Postman

Postman adalah sebuah *platform* yang memudahkan dalam menguji API. Salah satu fitur *Postman* yaitu dapat membuat pengujian API dengan baik dan cepat. Cara kerja *Postman* dengan mengklasifikasi *request* berdasarkan metode *request* dan URL [32].

Pengujian dilakukan untuk memastikan bahwa fitur pembuatan pesanan dan proses pembayaran pada aplikasi KosConnect berjalan dengan baik. Pengujian pertama dilakukan pada fitur pembuatan pesanan dengan metode *POST*. Permintaan API dikirim ke *endpoint* yang memerlukan parameter dinamis, seperti *room_id*, *boarding_house_id*, *owner_id*, dan *user_id*, yang diisi berdasarkan data dari sistem. Permintaan ini juga dilengkapi dengan *headers*, yaitu *Content-Type: application/json* dan *Authorization: Bearer*. Hasil pengujian menunjukkan bahwa permintaan berhasil dengan mengembalikan respons *201 Created*. Selain itu, jika terdapat data yang kurang, sistem memberikan respons *400 Bad Request*, sedangkan jika token tidak valid, sistem mengembalikan respons *401 Unauthorized*.

Selanjutnya, pengujian dilakukan pada proses pembayaran. Permintaan API dikirim menggunakan metode *POST* ke *endpoint* yang memerlukan *transaction_id* sebagai parameter dinamis berdasarkan transaksi yang sedang diproses. Permintaan ini juga menggunakan *header Authorization: Bearer* untuk memastikan keamanan akses. Sistem kemudian memproses pembayaran sesuai dengan metode yang dipilih oleh pengguna. Hasil pengujian menunjukkan bahwa sistem berhasil memproses pembayaran dan mengembalikan respons *200 OK* beserta detail transaksi yang telah diperbarui. Pengujian juga memastikan bahwa transaksi hanya dapat diproses jika masih dalam status yang memungkinkan untuk dilakukan pembayaran. Jika token tidak valid, sistem mengembalikan respons *401 Unauthorized*.

3.4.3 Hasil Pengujian JMeter

Apache JMeter dapat digunakan untuk menguji kinerja baik pada sumber daya statis dan dinamis, dan aplikasi dinamis web. Aplikasi ini digunakan untuk melakukan simulasi beban berat pada server, sekelompok server, jaringan atau objek untuk menguji kekuatannya atau untuk menganalisis kinerja keseluruhan di bawah jenis beban yang berbeda [33].

Pengujian performa dilakukan untuk menguji kinerja *endpoint /api/rooms/home* menggunakan *JMeter*. Tujuan pengujian adalah untuk mengukur waktu respons, mengidentifikasi potensi *bottleneck*, serta memastikan *endpoint* dapat menangani sejumlah *request* tanpa gangguan. Pengujian difokuskan pada metode *GET* tanpa penggunaan *header*. Pengujian ini dilakukan sebagai *baseline* untuk mengukur performa umum server, bukan dalam konteks transaksi atau pembayaran.

Dalam pengujian ini, *JMeter* digunakan sebagai alat uji beban dengan konfigurasi yang mencakup *Test Plan* bernama *KosConnect Home* dan *Thread Group* bernama *Thread Group Home*, yang terdiri dari 100 pengguna virtual dengan *ramp-up period* 10 detik dan satu kali pengulangan. *Request* dikirim menggunakan protokol HTTPS dengan *path /api/rooms/home*.

Hasil pengujian menunjukkan bahwa waktu respons rata-rata adalah 1126 ms, dengan waktu minimum 698 ms dan maksimum 3726 ms. Standar deviasi mencapai 642.28 ms, yang mengindikasikan variasi waktu respons yang cukup besar. Meskipun demikian, tidak ada *error* yang tercatat selama pengujian dengan *error rate* 0.00%. *Throughput* tercatat sebesar 9.2 *request* per detik, yang menunjukkan kapasitas server dalam menangani *request*, meskipun masih perlu dibandingkan dengan target *throughput* yang diharapkan. Selain itu, ukuran rata-rata *response body* adalah 6355.0 bytes.

Berdasarkan hasil pengujian, dapat disimpulkan bahwa *endpoint /api/rooms/home* memiliki performa yang cukup baik dalam menangani *request*. Waktu respons rata-rata masih dalam batas wajar, meskipun terdapat beberapa request dengan waktu respons yang tinggi. *Throughput* yang diperoleh masih perlu dievaluasi lebih lanjut untuk memastikan apakah sudah memenuhi target yang diharapkan. Namun, tidak adanya *error* selama pengujian menjadi indikator positif terhadap kestabilan *endpoint* ini.

4 KESIMPULAN

Berdasarkan hasil pengembangan dan pengujian, KosConnect berhasil meningkatkan efisiensi autentikasi dan keamanan dalam transaksi digital. Integrasi *Google OAuth* mempermudah pengguna dalam proses *login*, mengurangi ketergantungan pada metode tradisional berbasis email dan kata sandi, serta mengurangi kendala lupa

kredensial. Implementasi *Midtrans* sebagai *payment gateway* memungkinkan proses pembayaran berjalan lebih cepat dan terstruktur, sehingga pengguna tidak perlu melakukan transfer manual atau konfirmasi pembayaran secara terpisah. Hasil pengujian menunjukkan bahwa kedua fitur ini dapat berfungsi dengan baik sesuai spesifikasi yang dirancang, mendukung kelancaran autentikasi pengguna serta transaksi dalam sistem KosConnect.

REFERENCES

- [1] M. Rahmawati and E. Harahap, "Analisis Keuntungan Usaha Kos-Kosan Menggunakan Program Linear Dengan Aplikasi Geogebra Profit Analysis of a Boarding House Business using Linear Programming with Geogebra Application," *Jurnal Matematika*, vol. 20, no. 1, 2021.
- [2] E. Nurluthfiyah, "Analisis potensi pajak rumah kos pada Kelurahan Cipadung Kecamatan Cibiru Kota Bandung," UIN Sunan Gunung Djati Bandung, 2020.
- [3] Rifat Alhamidi - detikJabar, "Cuan Menggiurkan dari Pajak Kos-Kosan di Bandung," Detik jabar.
- [4] Nyaringindonesia.com, "Pajak Kos-Kosan di Kota Bandung Mencapai Miliaran Rupiah," Nyaringindonesia.com. Accessed: Feb. 19, 2025. [Online]. Available: <https://nyaringindonesia.com/pajak-kos-kosan-di-kota-bandung-mencapai-miliaran-rupiah/>
- [5] P. Studi Sistem Informasi Universitas Trilogi Jakarta Jl TMP Kalibata No, D. Tiga Kec Pancoran, K. Kunci, R. Kost, and S. Informasi, "RANCANG BANGUN SISTEM INFORMASI SEWA RUMAH KOST (E-KOST) BERBASIS WEBSITE CHALIDAZIA NIZAR," *Jurnal Sistem Informasi dan Sains Teknologi*, vol. 3, no. 1, 2021.
- [6] G. T. F. Zeck, "Jurnal Onlie Mahasiswa Sistem Informasi dan Akuntansi (ONESISMIK) 226 AMIK Dian Cipta Cendikia RANCANG BANGUN SISTEM INFORMASI KOS-KOSAN MENGGUNAKAN FRAMEWORK RAPID APPLICATION DEVELOPMENT".
- [7] F. Y. Nursita, A. D. Rizki, and others, "Pengembangan Antarmuka Pengguna untuk Aplikasi Pencari Kost (Gkost)," *Instink: Inovasi Pendidikan, Teknologi Informasi dan Komputer*, vol. 3, no. 2, pp. 96–104, 2024.
- [8] R. MULIANI, "RANCANG BANGUN SISTEM INFORMASI PENYEWAAN KAMAR KOST BERBASIS WEBSITE (STUDI KASUS PADA KOST BU NUR DESA PETERONGAN)," Universitas Pesantren Tinggi Darul'Ulum, 2022.
- [9] D. Ambarwati and U. Teknologi Yogyakarta Jl Ringroad Utara Jombor Sleman Yogyakarta, "SISTEM INFORMASI KOS DAN HOMESTAY BERBASIS WEBSITE."
- [10] S. Sutariyani, W. Wendanto, and A. Yulianto, "Sistem Informasi Kos Area Surakarta Berbasis Android," *Go Infotech: Jurnal Ilmiah STMIK AUB*, vol. 26, no. 1, p. 12, Jun. 2020, doi: 10.36309/goi.v26i1.119.
- [11] J. L. Harefa and A. P. Harianja, "Rancang Bangun Ruteakost Sistem Informasi Rumah Kost Berbasis Web di Kota Medan," *SATESI: Jurnal Sains Teknologi dan Sistem Informasi*, vol. 4, no. 2, pp. 164–171, 2024.
- [12] N. Hasti and Muhammad Inal Zaelani, "Sistem Informasi Pengelolaan Rumah Kost Budi Sari Berbasis Website," *Jurnal Pengabdian Masyarakat dan Riset Pendidikan*, vol. 2, no. 2, pp. 26–32, Nov. 2023, doi: 10.31004/jerkin.v2i2.202.
- [13] A. Rahmatillah, T. Saputra, and W. B. Hartiningsih, "Sistem Informasi Rumah Kost Berbasis Web Studi Kasus pada Paviliun Sejahtera," vol. 11.
- [14] M. I. Baehaqi and T. Suryana, "PENERAPAN TEKNOLOGI GEOFENCE PADA APLIKASI PEMESANAN KAMAR KOSAN," *KOMPUTA : Jurnal Ilmiah Komputer dan Informatika*, vol. 10, no. 1, 2021.
- [15] R. Sepjo, M. Amin, and H. Kurniawan, "Rancang Bangun Sistem Sewa Rumah Kontrakan Berbasis Web," *JOURNAL ZETROEM*, vol. 5, no. 2, pp. 174–179, 2023.
- [16] Bendra Wardana, Roni Habibi, and M. Harry K Saputra, "Comparison of SAW Method and Topsis in Assesing The Best Area Using HSE Standards," *EMITTER International Journal of Engineering Technology*, vol. 8, no. 1, pp. 126–139, Jun. 2020, doi: 10.24003/emitter.v7i2.423.
- [17] R. As and M. Shalahudin, "Rekayasa perangkat lunak terstruktur dan berorientasi objek," 2021.
- [18] A. Firdaus, "Pemodelan Proses Bisnis Konveksi di Tasikmalaya dengan Business Process Model and Notation (BPMN)," *Jurnal Ekonomi dan Bisnis Digital*, vol. 1, no. 3, pp. 133–142, 2022.
- [19] L. Setiyani, "Desain Sistem: Use Case Diagram," in *Prosiding Seminar Nasional Inovasi dan Adopsi Teknologi (INOTEK)*, 2021, pp. 246–260.
- [20] S. Sulistijono, M. Pradana, I. Hasbi, D. Wahyu Nugraha, M. Amran Hakim Siregar, and R. Habibi Prodi, "Web-Based Application of High School Laboratory Administration: Case Study at SMA Pasundan 8, Bandung, Indonesia."
- [21] M. R. Adani, "Golang: Pengertian, Fungsi, Jenis Framework, dan Kelebihan," Mar. 2021. [Online]. Available: <https://www.sekawanmedia.co.id/blog/belajar-golang/>

- [22] digitalskola, “Golang Adalah: Definisi, Keunggulan, dan Contoh Terlengkap,” Mar. 2024. [Online]. Available: <https://digitalskola.com/blog/home/golang-adalah>
- [23] L. dan Distribusi Konten Referensi Belajar Lainnya i, “Table of Contents.”
- [24] E. Kurniawanto Putra, F. Rahmayeni, D. Jurusan, T. Informatika, and M. Jurusan, “IMPLEMENTASI DATABASE MONGODB UNTUK SISTEM INFORMASI BIMBINGAN KONSELING BERBASIS WEB (Studi Kasus: SMP N 1 Sawahlunto),” vol. 4, no. 1, 2016.
- [25] P. Philippaerts, D. Preuveneers, and W. Joosen, “OAuth: Exploring Security Compliance in the OAuth 2.0 Ecosystem,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Oct. 2022, pp. 460–481. doi: 10.1145/3545948.3545955.
- [26] D. A. Kusnadi, M. Iqbal, and B. Prabowo, “Perancangan dan Implementasi Protokol OAuth 2.0 pada Pengembangan Produk Coofis Verse di PT ARM Solusi,” *eProceedings of Applied Science*, vol. 10, no. 6, p. 1472, 2024.
- [27] by Srivathsan Morkonda Gnanasekaran, “USER PRIVACY IN OAUTH-BASED SINGLE SIGN-ON SYSTEMS,” 2024.
- [28] G. K. Wardana, B. Rahayudi, and W. H. N. Putra, “Pengembangan E-Commerce dengan Integrasi API Payment Gateway Midtrans,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 11, pp. 4770–4774, 2021.
- [29] F. R. Pratama, N. Santoso, and L. Fanani, “Pengembangan Aplikasi E-Commerce Menggunakan Payment Gateway Midtrans,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 4, pp. 1133–1140, 2020.
- [30] A. Suryansah, R. Habibi, R. M. Awangga, R. Nuraini, and S. Fatonah, “Implementasi Face Recognition Untuk Mengakses Ruangan,” vol. 3, no. 3, p. 25, 2020.
- [31] V. Febrian, M. R. Ramadhan, M. Faisal, and A. Saifudin, “Pengujian pada Aplikasi Penggajian Pegawai dengan menggunakan Metode Blackbox,” *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 1, pp. 61–66, 2020.
- [32] A. S. Sari and R. Hidayat, “Designing website vaccine booking system using golang programming language and framework react JS,” *JISICOM (Journal of Information System, Informatics and Computing)*, vol. 6, no. 1, pp. 22–39, 2022.
- [33] A. C. Barus, J. Harungguan, and E. Manulu, “Pengujian api website untuk perbaikan performansi aplikasi ditenu,” *Journal of Applied Technology and Informatics Indonesia*, vol. 1, no. 2, pp. 14–21, 2021.