ISSN : 1411-1624
e-ISSN: 2621-8089

Perancangan Sistem Informasi Kerja Paruh Waktu Mahasiswa dengan Lingkup Internal Program Studi X

Jonathan Chandra[1*], Jimmy Agustian Loekito[2]
Universitas Kristen Maranatha[1, 2]
Jalan Prof. drg. Surya Sumantri, M.P.H. No. 65 Bandung, Indonesia
Sur-el : jonathan.chandra@eng.maranatha.edu[1], jimmy.al@eng.maranatha.edu[2]
[*)] Corresponden Author

Received: 16 Juni 2025 Reviewed: 19 June 2025 Accepted: 23 June 2025

Abstract : *The information system in the form of part-time job vacancies in the scope of study programme X is one of the containers that is needed both from employers (lecturers) and job seekers (students). The information system regarding part-time work in study programme X is able to provide convenience for employers in finding workers who are in accordance with what they need and also provide opportunities in the form of experience for job seekers in study programme X. This research work begins with a system design design with predetermined limits based on observations of the X study programme which is used as a benchmark. The development of this information system is determined using the prototype development method because of the short development time and must be able to be used while the development itself takes place. After obtaining the limitations from the observation results, a database system design will be carried out for the purposes of this research information system application which will be applied to the system. The next step is the information system design process which is broken down into two parts, namely the front-end and back-end parts. For testing this information system itself using the black box testing method with the main target, namely the original user as the respondent.*

Keywords: *Information system, part-time, student, study programme.*

Abstrak : *Sistem informasi berupa lowongan pekerjaan paruh waktu di lingkup program studi X merupakan salah satu wadah yang sangat dibutuhkan baik dari para pemberi kerja (dosen) dan juga para pencari kerja (mahasiswa). Sistem informasi mengenai kerja paruh waktu di program studi X mampu memberikan kemudahan untuk para pemberi kerja dalam mencari tenaga kerja yang sesuai dengan yang dibutuhkannya dan juga memberi kesempatan berupa pengalaman untuk para pencari kerja di program studi X. Pengerjaan penelitian ini dimulai dengan desain perancangan sistem dengan batasan yang telah ditentukan berdasarkan hasil pengamatan terhadap program studi X yang dijadikan sebagai tolak ukurnya. Pengembangan sistem informasi ini ditentukan menggunakan metode pengembangan prototype karena waktu pengembangan yang singkat dan harus dapat digunakan selagi dalam pengembangan itu sendiri berlangsung. Setelah didapatkan batasan dari hasil pengamatan maka akan dilakukan perancangan sistem basis data guna keperluan aplikasi sistem informasi penelitian ini yang akan diaplikasikan ke dalam sistem. Langkah berikutnya adalah proses perancangan sistem informasi yang dipecah menjadi dua bagian yaitu bagian front-end dan juga bagian back-end. Untuk pengujian sistem informasi ini sendiri menggunakan metode black box testing dengan target utama yaitu pengguna asli sebagai respondennya.*

Kata kunci*: Sistem informasi, kerja paruh waktu, mahasiswa, program studi.*

1. PENDAHULUAN

Pengembangan sistem informasi untuk lowongan pekerjaan paruh waktu dapat sangat berguna untuk memfasilitasi interaksi efisien antara dosen sebagai pemberi kerja dan mahasiswa sebagai pencari kerja. Sistem ini dapat membantu mahasiswa untuk mendapatkan pengalaman kerja yang memungkinkan penerapan pengetahuan akademik dan non-

ISSN : 1411-1624
e-ISSN: 2621-8089

akademik, sekaligus mempersiapkan mereka untuk dunia kerja profesional, serta dosen yang memerlukan tenaga kerja berkompetensi untuk mendukung tugas akademik dan non-akademik [1]. Keterbatasan sumber daya manusia dan waktu memperparah tantangan ini, mendorong pengembangan sistem berbasis teknologi yang fleksibel dan mudah diakses [2]. *Platform* berbasis *web*, sebagaimana ditunjukkan dalam penelitian sebelumnya, memungkinkan pengelolaan terpusat informasi lowongan pekerjaan paruh waktu, mempermudah mahasiswa mengakses peluang yang kompatibel dengan jadwal akademik mereka [3].

Pengalaman kerja paruh waktu telah terbukti memberikan kontribusi signifikan terhadap peningkatan kinerja akademik dan kesiapan karier mahasiswa. Studi empiris menunjukkan bahwa mahasiswa yang terlibat dalam penempatan kerja memperoleh peningkatan skor akademik sebesar 3,22 poin persentase dan memiliki peluang dua kali lipat untuk kemajuan akademik [4]. Untuk mendukung hal ini, digitalisasi melalui *platform* berbasis *web* meningkatkan efisiensi dan aksesibilitas dalam pengelolaan penempatan kerja, menciptakan ekosistem yang responsif terhadap kebutuhan akademik dan profesional[5].

Penelitian sebelumnya seperti penerapan *Progressive Web Apps*, memungkinkan aksesibilitas luas untuk lowongan pekerjaan dosen melalui perangkat umum [6], serta sistem manajemen penempatan berbasis *web* dengan fitur pengelolaan profil dan pemantauan *real-time* untuk meningkatkan interaktivitas [7]. Lebih lanjut, teknik penyaringan hibrid dalam sistem rekomendasi penempatan magang memastikan kecocokan optimal antara pelamar dan lowongan [8], sementara platform terintegrasi mengoptimalkan proses rekrutmen kampus melalui pengelolaan data dan komunikasi yang efisien [9] [10].

Kemajuan teknologi terkini memberikan peluang signifikan untuk meningkatkan efektivitas sistem informasi lowongan kerja di lingkungan akademik. Analitik data real-time memungkinkan pemantauan tren penempatan melalui dashboard interaktif, mendukung pengambilan keputusan berbasis data [11]. Antarmuka pengguna yang dirancang secara intuitif memastikan aksesibilitas bagi pengguna dengan berbagai tingkat keahlian teknis [12], sedangkan pendekatan gamifikasi meningkatkan keterlibatan pengguna melalui elemen interaktif [13]. Penelitian ini mengadopsi teknologi seperti Node.js, ExpressJS, dan MySQL dalam kerangka *Multi-page Application (MPA)* untuk memenuhi kebutuhan program studi, dengan mengintegrasikan fitur penilaian dan rekomendasi kerja guna memperkuat relevansi sistem dalam konteks akademik [14] [15]. Namun, tantangan seperti skalabilitas dan pemeliharaan sistem tetap memerlukan sumber daya teknis yang memadai [16]. Untuk memaksimalkan dampaknya, integrasi sistem ini ke dalam kurikulum akademik dapat memperkaya pembelajaran melalui pengalaman praktis, sekaligus mendukung tujuan pendidikan jangka panjang [17].

ISSN : 1411-1624
e-ISSN: 2621-8089

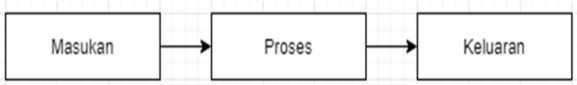
2. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode *Prototype Development Model* sebagai pendekatan utama dalam membangun sistem informasi lowongan pekerjaan paruh waktu berbasis web. Model ini dipilih karena mampu memberikan fleksibilitas tinggi dalam pengembangan sistem yang responsif terhadap kebutuhan pengguna melalui tahapan iteratif [18]. Pendekatan ini juga dinilai efektif dalam menghasilkan sistem yang dapat segera diujicobakan dan diperbaiki berdasarkan masukan pengguna, terutama dalam konteks sistem berbasis *web* yang memerlukan validasi antarmuka.

Tahapan dalam metode *prototyping* yang diterapkan pada penelitian ini mencakup:

1. Identifikasi kebutuhan sistem melalui studi literatur terhadap sistem serupa yang telah dikembangkan sebelumnya, yang menjadi dasar dalam merumuskan fitur utama seperti pencarian lowongan, manajemen akun pengguna, dan pengelolaan kompetensi.
2. Perancangan awal sistem, meliputi desain arsitektur sistem *client-server*, perancangan basis data, dan pemetaan alur proses antara *front-end* dan *back-end*. Arsitektur ini dirancang agar mendukung proses autentikasi, pengelolaan data secara dinamis, serta koneksi antara komponen *front-end* dengan sistem basis data MySQL melalui *API back-end*.
3. Implementasi prototipe awal dilakukan menggunakan teknologi *web modern*, yaitu Node.js dan Express.js untuk sisi *back-end*, MySQL untuk basis data, HTML/CSS dan JavaScript untuk *front-end*, serta PassportJS sebagai mekanisme autentikasi.
4. Pengujian sistem dilaksanakan menggunakan metode *black-box testing*, untuk mengevaluasi fungsi utama tanpa memperhatikan struktur internal kode. Uji coba mencakup proses *login*, registrasi, manajemen lowongan, serta *input* dan validasi data.
5. Evaluasi dan revisi prototipe dilakukan berdasarkan hasil pengujian dan masukan dari pengguna uji coba. Proses ini memungkinkan peningkatan iteratif terhadap sistem untuk mencapai tingkat fungsionalitas dan kegunaan yang optimal.

2.1 Arsitektur Sistem



Gambar 1. Diagram sistem secara umum.

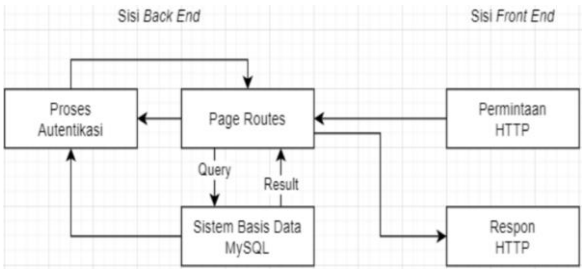
Gambar 1 menampilkan merupakan blok diagram sistem sederhana yang menjelaskan alur kerangka dasar dalam pengembangan sistem informasi, yang terdiri dari tiga komponen utama yaitu Masukan (*Input*), Proses, dan Keluaran (*Output*).

Bagian Masukan merupakan segala bentuk data atau informasi awal yang diterima sistem dari pengguna atau sumber data lainnya. Dalam konteks sistem informasi lowongan pekerjaan paruh waktu, masukan bisa berupa data lowongan dari dosen, data profil mahasiswa, atau permintaan pencarian pekerjaan. Bagian Proses merupakan inti dari sistem, tempat di mana data yang diterima diolah menggunakan

ISSN : 1411-1624
e-ISSN: 2621-8089

logika sistem, aturan akses, serta fungsi-fungsi lainnya untuk menghasilkan informasi yang dibutuhkan. Contohnya mencakup pencocokan antara kriteria lowongan dengan profil mahasiswa, validasi akun, dan pengelolaan basis data. Terakhir, Keluaran adalah hasil akhir dari proses tersebut yang ditampilkan kembali kepada pengguna, seperti daftar lowongan pekerjaan yang sesuai, notifikasi, atau status lamaran pekerjaan. Diagram ini menggambarkan prinsip kerja sistem secara umum dan digunakan sebagai dasar pemahaman dalam perancangan sistem informasi ini.

Arsitektur sistem yang dikembangkan berbasis *client-server*, terdiri dari dua komponen utama yaitu *front-end* (*client-side*) dan *back-end* (*server-side*). Komponen *front-end* bertugas menampilkan antarmuka pengguna dan menangani Masukan, sedangkan *back-end* bertugas memproses permintaan dari *front-end*, mengelola autentikasi, dan menjalankan *query* ke basis data. Sistem ini juga menggunakan *HTTP request-response* sebagai protokol komunikasi antar lapisannya.



Gambar 2. *Block Diagram* Arsitektur *Front-end* dan *Back-end*.

Gambar 2 menjelaskan alur komunikasi dan komponen utama dari sistem ini. Pengguna mengakses aplikasi melalui *browser* yang mengirimkan permintaan *HTTP* ke *back-end* (*server*). *Back-end* akan memproses permintaan tersebut, melakukan autentikasi menggunakan *PassportJS* sebagai *middleware*, lalu meneruskan *query* ke basis data MySQL yang telah dirancang. Data yang diperoleh dari basis data tersebut akan dikirimkan kembali dalam bentuk hasil dan diterima oleh *front-end* lalu menjadi tampilan yang dapat dimengerti oleh pengguna. Struktur ini mendukung pengelolaan rute (*page routes*) yang terorganisir, serta memisahkan tanggung jawab setiap komponen secara modular.

Dalam penelitian ini, pendekatan *UI/UX* digunakan sebagai bagian integral dari metodologi pengembangan sistem informasi untuk memastikan antarmuka yang mudah dipahami dan digunakan oleh seluruh jenis pengguna, yaitu Dosen dan Mahasiswa. Desain antarmuka dirancang agar setiap peran dapat berinteraksi dengan sistem tanpa mengalami kebingungan, serta memudahkan navigasi dalam menjalankan tugas masing-masing. *UI (User Interface)* adalah antarmuka visual yang digunakan untuk mengoperasikan sistem, sedangkan *UX (User Experience)* adalah pengalaman pengguna saat menggunakan suatu produk. Evaluasi pengalaman pengguna dilakukan melalui survei kepuasan berdasarkan beberapa parameter, seperti kesesuaian tampilan antarmuka *UI (User Interface)* dengan kebutuhan, serta kemudahan dalam navigasi dan operasional sistem. Dari hasil survei, diperoleh skor kepuasan keseluruhan antara 73,33% hingga 100%, menunjukkan bahwa sistem telah cukup memenuhi ekspektasi penggunanya [3].

Pendekatan *UI/UX* dalam penelitian ini tidak hanya berperan dalam menciptakan

ISSN : 1411-1624
e-ISSN: 2621-8089

antarmuka yang mudah dipahami dan digunakan oleh seluruh jenis pengguna, tetapi juga menjadi dasar dalam pemilihan arsitektur sistem berbasis *MPA (Multi-page Application)*. Desain *UI/UX* yang terstruktur membantu memastikan bahwa setiap pengguna dapat berinteraksi dengan sistem sesuai perannya, baik dalam melihat, menciptakan, mengubah, maupun menghapus data seperti lowongan pekerjaan atau informasi profil[19]. Untuk mendukung pengalaman pengguna yang optimal dan menjamin keamanan serta efisiensi pengelolaan akses, dipilihlah *MPA* sebagai bentuk aplikasi *web* yang memisahkan halaman berdasarkan fungsi dan peran pengguna. *MPA* memungkinkan penjelasan kewenangan secara lebih jelas melalui tampilan yang disesuaikan, menjaga efektivitas dalam manajemen data dengan membatasi akses hanya kepada pengguna yang berhak, serta memberikan fleksibilitas tinggi dalam pengembangan karena halaman baru dapat ditambahkan atau disunting tanpa memengaruhi bagian sistem lainnya [18]. Dengan integrasi antara *UI/UX* yang baik dan struktur *MPA*, sistem informasi ini mampu memberikan pengalaman yang aman, efisien, dan mudah digunakan bagi seluruh pengguna.

Untuk mendukung arsitektur *MPA* yang terintegrasi dengan prinsip *UI/UX*, sistem ini dikembangkan dengan membagi struktur kerja menjadi dua bagian utama, yaitu *front-end* dan *back-end*. *Front-end* merupakan bagian sistem yang menampilkan antarmuka visual dan menyediakan interaksi langsung dengan pengguna melalui elemen-elemen grafis yang responsif. Elemen-elemen seperti form pendaftaran, tampilan daftar lowongan pekerjaan, serta navigasi peran pengguna (Dosen dan Mahasiswa) dibangun menggunakan teknologi berbasis HTML, CSS, dan JavaScript yang responsif agar dapat diakses dengan mudah di berbagai perangkat. Sebaliknya, *back-end* berperan sebagai penyedia data yang mendukung kebutuhan *front-end*. Teknologi *back-end* yang digunakan meliputi bahasa pemrograman *server-side* seperti PHP atau Node.js dan sistem manajemen basis data seperti MySQL. Integrasi antara *front-end* dan *back-end* dilakukan melalui *API (Application Programming Interface)* yang memungkinkan komunikasi data secara aman dan efisien antara tampilan pengguna dan server. Pemisahan ini tidak hanya meningkatkan skalabilitas dan keamanan sistem, tetapi juga memudahkan proses pemeliharaan dan pengembangan lanjutan seiring kebutuhan pengguna yang terus berkembang [20][21].

Dalam penelitian ini, dibutuhkan serangkaian sumber daya berbasis JavaScript untuk mendukung pengembangan sistem *back-end* yang efisien, aman, dan mudah dikembangkan. Node.js dipilih sebagai *platform runtime* untuk menjalankan skrip *server-side* berbasis JavaScript, yang memberikan kemampuan tinggi dalam menangani permintaan *HTTP/HTTPS* secara asinkron dan efisien [22]. Untuk mempermudah pembangunan *back-end*, digunakan Express.js sebagai *framework web* yang berjalan di atas Node.js. Express.js menyederhanakan pembuatan rute, pengelolaan *middleware*, serta integrasi dengan basis data, sehingga mempercepat proses pengembangan dan pemeliharaan sistem [23]. Di sisi keamanan, Passport.js diimplementasikan sebagai

ISSN : 1411-1624
e-ISSN: 2621-8089

middleware untuk menangani proses autentikasi pengguna. Dengan Passport.js, sistem dapat menyediakan mekanisme *login* dan *logout* yang aman, sekaligus mendukung berbagai strategi autentikasi seperti *username* dan *password*, OAuth, dan juga lainnya [24]. Ketiga sumber daya ini bekerja secara terintegrasi untuk memastikan sistem informasi yang dikembangkan tidak hanya fungsional dan mudah digunakan, tetapi juga responsif dan aman dari akses yang tidak sah.

Setelah fondasi *back-end* dibangun menggunakan Node.js, Express.js, dan Passport.js, komponen penting berikutnya dalam pengembangan sistem informasi adalah pengelolaan data melalui basis data relasional. Dalam penelitian ini, digunakan MySQL sebagai sistem manajemen basis data (*DBMS*) untuk menyimpan dan mengatur data yang bersifat terstruktur, seperti data pengguna, lowongan pekerjaan, kompetensi, dan aktivitas sistem. MySQL dipilih karena keandalannya, kinerja tinggi dalam menangani *query* kompleks, serta kompatibilitasnya yang luas dengan berbagai platform dan bahasa pemrograman, termasuk Node.js melalui pustaka seperti mysql2 ataupun Sequelize [25]. MySQL menggunakan model relasional berbasis tabel dengan skema yang ketat, memungkinkan integritas data terjaga melalui *constraint* seperti *primary key*, *foreign key*, dan *indexing*. Hal ini menjadikan MySQL sangat cocok untuk sistem yang memerlukan konsistensi data dan relasi antarentitas, seperti hubungan antara mahasiswa dan lowongan kerja yang dilamar dan juga hubungan antara mahasiswa dan kompetensinya masing-masing. Dengan integrasi yang solid antara lapisan *back-end* dan MySQL sebagai basis datanya, sistem ini mampu menyediakan layanan yang cukup andal, aman, dan terstruktur dengan baik dalam pengelolaan informasi.

Desain basis data disusun untuk menangani kebutuhan sistem yang melibatkan entitas pengguna, lowongan pekerjaan, dan kompetensi. Desain ini disusun berdasarkan prinsip normalisasi untuk mengurangi redundansi data dan meningkatkan efisiensi penyimpanan serta akses data [26].

Struktur basis data terdiri dari tiga tabel utama, yaitu:

1. Tabel Pengguna: Menyimpan informasi pengguna seperti nama, *email*, kata sandi (yang disimpan dalam bentuk terenkripsi), serta peran masing-masing pengguna seperti admin, mitra, pencari kerja / mahasiswa.
2. Tabel Pekerjaan: Menyimpan data lowongan pekerjaan yang ditawarkan oleh mitra, seperti nama posisi pemberi kerja, deskripsi pekerjaan, syarat kompentensi yang dibutuhkan, dan beberapa jenis tanggal (unggah, waktu tenggat, dan sebagainya), dan juga referensi ke pengguna yang terkait.
3. Tabel Kompetensi: Menyimpan informasi mengenai keterampilan atau keahlian yang relevan, serta hubungan antar keterampilan dengan pekerjaan tertentu maupun hubungan dengan pencari kerja / mahasiswa.

Sistem dimulai ketika pengguna mengakses halaman utama melalui *browser*. Pengguna dapat melakukan registrasi atau *login*. Proses *login* akan memicu permintaan

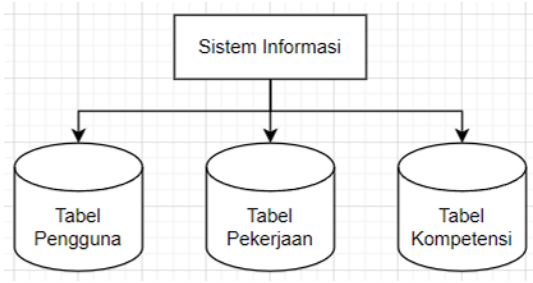
ISSN : 1411-1624
e-ISSN: 2621-8089

autentikasi ke *back-end*, yang memvalidasi kredensial terhadap data pengguna di basis data. Jika valid, pengguna diarahkan ke *dashboard* sesuai peran (mitra atau pencari kerja). Mitra dapat mengunggah lowongan pekerjaan beserta kompetensi yang dibutuhkan, sementara pencari kerja dapat menelusuri dan melamar pekerjaan.

Setiap interaksi *front-end* (seperti pengisian formulir atau pencarian data) dikirim sebagai HTTP request ke *back-end*. *Back-end* menangani validasi, meneruskan *query* ke basis data, kemudian mengembalikan hasil yang akan ditampilkan oleh *front-end*. Alur ini menjamin keamanan dan efisiensi pengelolaan data. Selain itu, sistem mendukung pembatasan hak akses berdasarkan peran pengguna yang diterapkan secara sistematis melalui *middleware* autentikasi. Pengujian sistem dilakukan menggunakan metode *black-box*, untuk memastikan bahwa setiap fitur berjalan sesuai spesifikasi tanpa mengevaluasi kode internal. *Black Box Testing* adalah metode pengujian perangkat lunak yang fokus pada fungsi dan output dari suatu sistem tanpa memperhatikan struktur internal, kode sumber, atau cara kerja sistem secara teknis. Penguji hanya mengetahui spesifikasi atau kebutuhan sistem dan melakukan pengujian berdasarkan *input* dan *output* yang diharapkan. Penelitian ini menggunakan *Black Box Testing* karena metode ini sangat sesuai untuk menguji sistem informasi dari sisi fungsi dan kebutuhan pengguna, tanpa perlu melihat atau memahami struktur kode sistem. Fokus utama dari penelitian ini adalah memastikan bahwa setiap fitur dan layanan sistem bekerja sesuai dengan yang diharapkan oleh pengguna, baik itu Admin, Mitra maupun Mahasiswa sebagai pencari kerja. Pengujian mencakup beberapa hal yaitu : Proses login dan registrasi, Input dan pengelolaan data pekerjaan, Validasi kompetensi, Navigasi antarmuka pengguna, dan Proses filtering dan pencarian pekerjaan.

Setiap skenario diuji untuk memastikan respon sistem sudah sesuai terhadap masukan yang benar maupun salah, serta stabilitas sistem dalam menangani permintaan secara simultan. Pengujian ini penting dalam menjamin keandalan sistem sebelum diterapkan secara nyata.

3. HASIL DAN PEMBAHASAN



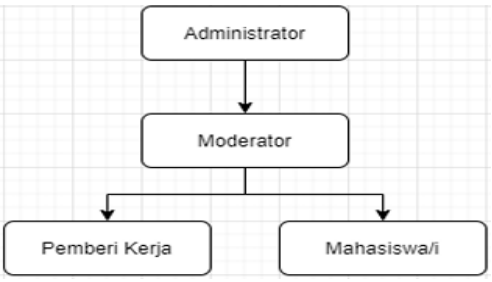
Gambar 3. Tabel basis data yang mewakili tiga bidang informasi.

Gambar 3 menampilkan hasil implementasi tabel basis data yang mewakili tiga bidang utama dalam sistem informasi kerja paruh waktu untuk Program Studi X, yaitu pengguna, kompetensi, dan pekerjaan. Ketiga tabel ini merupakan hasil analisis dari kebutuhan sistem yang sudah dilakukan sebelumnya. Tabel pengguna berisi data seperti ID, nama, peran (Admin, Moderator, Dosen/Mitra, atau Mahasiswa), dan status akun. Tabel kompetensi mencantumkan keterampilan atau keahlian yang dimiliki mahasiswa, contohnya seperti pemrograman, desain grafis, atau analisis data,

ISSN : 1411-1624
e-ISSN: 2621-8089

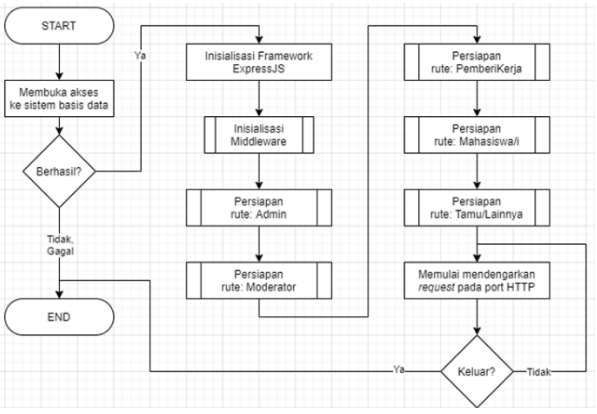
beserta tingkat kemahiran dan juga pengalamannya. Sementara itu, tabel pekerjaan menyimpan informasi lowongan yang ditawarkan dosen/mitra, meliputi judul pekerjaan, deskripsi, syarat kompetensi, dan status aktif/non-aktif. Hubungan antar-tabel ini memungkinkan sistem untuk mencocokkan mahasiswa dengan pekerjaan berdasarkan kompetensi yang relevan, memfasilitasi pencarian lowongan yang sesuai, serta memastikan manajemen data yang terstruktur. Desain basis data ini menjadi fondasi dasar bagi fungsionalitas sistem, seperti pendaftaran pekerjaan, pencarian kandidat yang sesuai, pelaporan pekerjaan, hingga pelacakan aktivitas antar pengguna.

Untuk menjaga konsistensi dan efisiensi, proses pengembangan tampilan *front-end* dimulai dari pengguna dengan fitur yang paling lengkap yaitu Admin, sehingga halaman untuk jenis pengguna lainnya (dengan posisi di bawah Admin) dapat dengan mudah dimodifikasi dari halaman tersebut karena sebagai referensi dasarnya.



Gambar 4. *Flowchart* pengembangan sisi *front-end*.

Flowchart pada Gambar 4 menggambarkan prioritas pengembangan tampilan sisi *front-end*. Proses pengembangan dimulai dari tingkat pengguna dengan fitur paling lengkap, yaitu Admin, yang memiliki akses penuh terhadap seluruh fitur sistem, seperti manajemen pengguna, kompetensi, hingga pekerjaan. Setelah halaman untuk Admin selesai dikembangkan, tim kemudian beralih ke pengguna dengan hak akses lebih terbatas, seperti Moderator, yang bertugas memediasi antara pemberi kerja (Dosen/Mitra) dan para pencari kerja (Mahasiswa). Tahap berikutnya adalah pengembangan halaman untuk Pemberi Kerja (Dosen/Mitra) dan Mahasiswa, yang memiliki fitur lebih spesifik sesuai dengan perannya masing-masing. Pendekatan ini memastikan konsistensi desain dan efisiensi pengembangan, karena halaman untuk pengguna dengan fitur terbatas dapat dimodifikasi dari *template* yang sudah dibuat untuk Admin. Selain itu, *flowchart* tersebut juga menekankan pentingnya pemisahan antarmuka berdasarkan peran pengguna untuk meningkatkan keamanan data dan memudahkan pemeliharaan sistem di masa yang akan datang. Dengan demikian, Gambar 4 secara visual menjelaskan strategi pengembangan front-end yang terstruktur dan berorientasi pada kebutuhan penggunanya.



Gambar 5. *Flowchart* pengerjaan sisi *back-end*.

Flowchart pada Gambar 5 menggambarkan alur kerja sisi *back-end* sistem informasi kerja paruh waktu berbasis *web*. Proses inisialisasi sistem

ISSN : 1411-1624
e-ISSN: 2621-8089

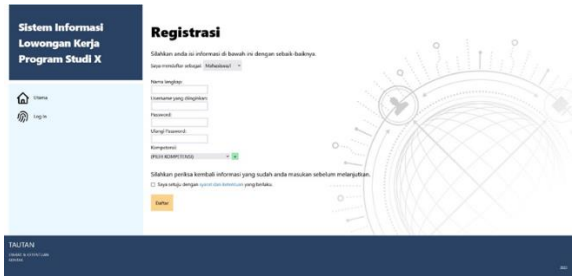
dimulai dari pembukaan akses terhadap basis data MySQL lalu dilanjutkan dengan inisialisasi beberapa *middleware* lalu diikuti oleh persiapan rute untuk masing-masing peran pengguna dalam sistem dan diakhiri dengan menunggu interaksi pengguna pada *server HTTP*. Sedangkan untuk proses interaksi dimulai ketika pengguna mengirimkan permintaan (*request*) melalui *front-end*, seperti *login*, pendaftaran, atau pencarian lowongan kerja. *Back-end* kemudian memproses permintaan tersebut dengan memeriksa kredensial pengguna terlebih dahulu dan dilanjutkan pengecekan validitas data dan pemilihan rute. Jika permintaan dan rute telah cocok, sistem akan berinteraksi dengan basis data MySQL untuk mengambil, menyimpan, atau memodifikasi data sesuai permintaan pengguna. Sebagai contoh, saat mahasiswa melamar pekerjaan, sistem akan mengecek kompetensi mereka di basis data sebelum mencocokkannya dengan lowongan yang tersedia. Hasil operasi ini kemudian dikirim kembali ke *front-end* sebagai respon, yang akhirnya ditampilkan kepada pengguna dalam bentuk informasi berupa tabel sebagai contohnya. *Flowchart* ini juga menekankan peran *framework* ExpressJS dan *middleware* PassportJS dalam mengelola rute, autentikasi, dan sesi pengguna. Node.js bertindak sebagai *server* utama yang menghubungkan seluruh komponen tersebut, memastikan komunikasi terjadi secara lancar antara *front-end*, basis data, dan komponen-komponen sistem tersebut sesuai dengan penjelasan proses inisialisasi sistem. Dengan struktur yang jelas dan terarah, *flowchart* ini menunjukkan bagaimana *back-end* dapat bekerja secara efisien untuk mendukung fungsionalitas sistem, mulai dari segi keamanan hingga pengolahan data yang kompleks sesuai dengan berbagai permintaan pengguna dan perannya masing-masing.

3.1 Desain dan Pengujian Sistem Informasi

Berikut ini adalah cuplikan tampilan halaman utama yang telah berhasil diimplementasikan dari rancangan yang sudah dibuat.

Gambar 6. Halaman tampilan utama sistem informasi.

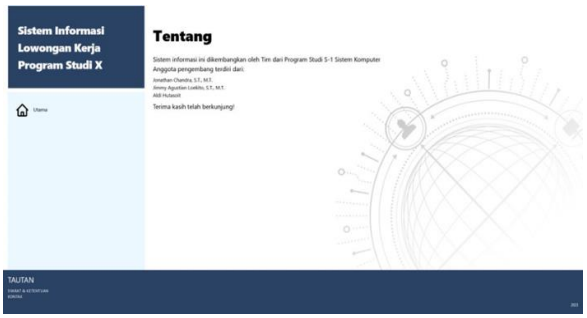
Selain itu terdapat juga hasil tampilan dari desain beberapa halaman yang telah berhasil diimplementasikan ke dalam sistem informasi penelitian ini dan berfungsi dengan baik pada bagian *front-end*.



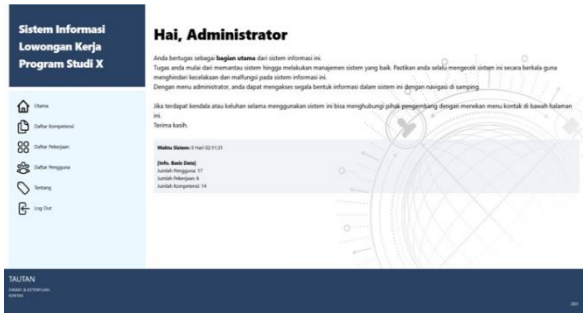
Gambar 7. Halaman Registrasi pengguna baru.

Gambar 8. Halaman *Log in* pengguna terdaftar.

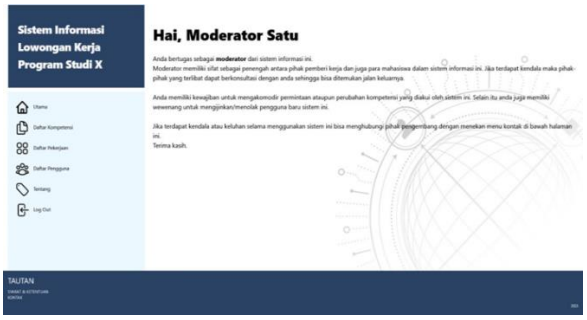
ISSN : 1411-1624
e-ISSN: 2621-8089



Gambar 9. Halaman tentang sistem informasi.



Gambar 10. Tampilan halaman utama pengguna peran Admin.



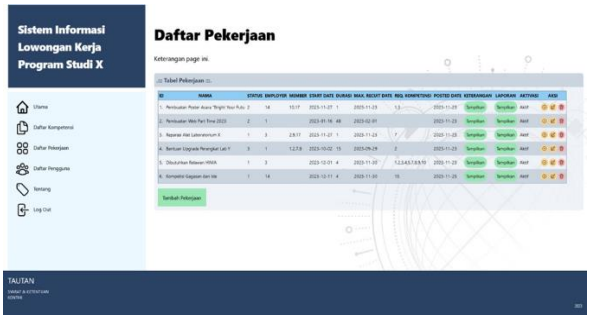
Gambar 11. Tampilan halaman utama pengguna peran Moderator.

Gambar 12. Tampilan halaman utama pengguna peran Pemberi Kerja.

Gambar 13. Tampilan halaman utama pengguna peran Mahasiswa/i.

Gambar 14. Cuplikan tabel kompetensi dalam sistem informasi.

Gambar 15. Cuplikan tabel pengguna dalam sistem informasi.



Gambar 16. Cuplikan tabel pekerjaan dalam sistem informasi.

ISSN : 1411-1624
e-ISSN: 2621-8089

Sistem informasi kerja paruh waktu ini dibangun sebagai *Multi-page Application (MPA)* dengan antarmuka terpisah untuk setiap peran pengguna (Admin, Moderator, Dosen/Mitra/Pemberi Kerja, dan Mahasiswa). Halaman utama (Gambar 6) dirancang secara minimalis dengan navigasi intuitif, berfungsi untuk menyediakan akses cepat ke halam *Log In* dan Registrasi. Tampilan halaman Registrasi (Gambar 7) memungkinkan pengguna baru untuk mendaftarkan diri dengan mengisi formulir berdasarkan peran yang diinginkannya, sementara halaman *Log In* (Gambar 8) mengijinkan pengguna yang sudah terdaftar sebelumnya untuk melakukan autentikasi secara aman melalui *middleware* PassportJS pada sistem ini.

Untuk menambah informasi pengguna, sistem informasi ini juga menyertakan halaman "Tentang" (Gambar 9) yang memuat penjelasan tentang sistem informasi ini dan juga tim pengembang dari sistem ini. Untuk peran tertentu seperti Admin maka dengan rute khusus sistem akan memberikan akses dashboard secara khusus (Gambar 10) untuk mengelola data pengguna, kompetensi, dan pekerjaan secara menyeluruh, hingga fitur aktivasi/deaktivasi akun. Halaman untuk pengguna dengan peran Moderator (Gambar 11) memiliki antarmuka serupa seperti Admin namun dengan fitur yang lebih terbatas dengan fokus utama pada pemantauan Dosen/Mitra maupun Mahasiswa dan juga dapat melakukan verifikasi kompetensi mahasiswa hingga melakukan moderasi pada pekerjaan yang sedang berlangsung.

Mahasiswa dan Dosen/Mitra diberikan halaman/tampilan terpisah (Gambar 12 dan 13). Mahasiswa dapat melihat daftar lowongan pekerjaan yang sesuai dengan kompetensi mereka (Gambar 14) dan mengajukan lamaran pekerjaan, sedangkan Dosen/Mitra mampu membuka lowongan baru serta melakukan pelaporan pekerjaan yang sedang berlangsung. Seluruh data pengguna (Gambar 15) dan lowongan kerja (Gambar 16) dikelola melalui tabel terstruktur dalam basis data MySQL, memastikan konsistensi dan kemudahan pemutakhiran jika terjadi perubahan maupun kendala selama sistem berjalan.

Tabel 1. Hasil pengujian *Black Box (B.B) Testing* terhadap sistem informasi.

No.	*Jenis Pengguna*	*Halaman*	*Aksi*	*Hasil Uji B.B.*
1	Admin	*Log In*	*Log In*	✓
2		Indeks	Navigasi	✓
3		Daftar Kompetensi	Mendaftar	✓
4			Mengubah	✓
5			Menghapus	✓
6			Aktivasi/Deaktivasi	✓
7		Daftar Pekerjaan	Mendaftar	✓
8			Mengubah	✓
9			Menghapus	✓
10			Aktivasi/Deaktivasi	✓
11		Daftar Pengguna	Mendaftar	✓
12			Mengubah	✓
13			Menghapus	✓
14			Aktivasi/Deaktivasi	✓
15	Moderator	*Log In*	*Log In*	✓
16		Indeks	Navigasi	✓
17		Daftar Kompetensi	Mendaftar	✓
18			Mengubah	✓
19			Menghapus	✓
20			Aktivasi/Deaktivasi	✓
21		Daftar Pekerjaan	Mengubah	✓
22			Aktivasi/Deaktivasi	✓
23		Daftar Pengguna	Aktivasi/Deaktivasi	✓
24	Pemberi Kerja	*Log In*	*Log In*	✓
25		Indeks	Navigasi	✓
26		Pekerjaan	Mendaftar	✓
27			Mengubah miliknya	✓

ISSN : 1411-1624
e-ISSN: 2621-8089

No.	*Jenis Pengguna*	*Halaman*	*Aksi*	*Hasil Uji B.B.*
28	Mahasiswa/i	Kompetensi	Melihat	✓
29		Akun	Mengubah	✓
30		*Log In*	*Log In*	✓
31		Indeks	Navigasi	✓
32		Pekerjaan	Melihat	✓
33			Mengubah (Melamar)	✓
34		Akun	Mengubah	✓
35			Melihat Kompetensi	✓

Sumber : Hasil Pengolahan Data

Tabel 1 menyajikan hasil pengujian *black-box testing* terhadap seluruh fitur sistem berdasarkan berbagai peran pengguna. Pengujian mencakup total 35 skenario yang terbagi dalam empat kategori peran pengguna: Admin, Moderator, Pemberi Kerja (dosen/mitra), dan Mahasiswa. Setiap skenario menguji fungsionalitas kritis seperti *login*, navigasi, manajemen data (pendaftaran, pengeditan, penghapusan), hingga aktivasi/deaktivasi akun maupun lowongan kerja. Hasilnya menunjukkan bahwa seluruh fitur utama telah berhasil beroperasi sesuai dalam ekspektasi (ditandai dengan centang/✓), membuktikan bahwa sistem telah memenuhi kebutuhan dasar pada setiap perannya. Sebagai contoh, Admin mampu mengelola seluruh aspek sistem (Baris 1-14), sementara Mahasiswa mampu mengakses seluruh fitur terbatas contoh salah satunya seperti melamar pekerjaan (Baris 30-35).

Tabel 2. Respon pengguna berupa timbal balik terhadap sistem.

No Pengguna.	*Respon %*	*Kesesuaian %*	*Pengoperasian %*	*Skor Keseluruhan*
1	100	80	80	86,67
2	100	100	100	100
3	100	100	100	100
4	100	100	100	100
5	100	80	100	93,33
6	100	100	100	100
7	80	100	80	86,67
8	100	100	100	100
9	100	100	100	100
10	80	80	60	73,33

Sumber : Hasil Pengolahan Data

Tabel 2 melengkapi Tabel 1 dengan menampilkan respons pengguna nyata terhadap sistem melalui survei. Data diukur berdasarkan tiga indikator: respon sistem (kecepatan dan konsistensi), kesesuaian fitur dengan kebutuhan, dan kemudahan pengoperasian. Secara keseluruhan, sistem mendapat skor tinggi (73–100%), dengan mayoritas pengguna (6 dari 10 responden) memberikan nilai sempurna (100%). Sedangkan terdapat beberapa catatan minor muncul pada respon sistem dan pengoperasian (contoh: Poin 7 dan 10 dengan skor 80% dan 73%), yang mengindikasikan perlunya optimalisasi performa antarmuka untuk pengguna tertentu. Kedua tabel ini secara kolektif membuktikan bahwa sistem tidak hanya berfungsi teknis (Tabel 1), tetapi juga dapat diterima dengan baik oleh para penggunanya dengan peran yang berbeda-beda (Tabel 2).

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem informasi kerja paruh waktu untuk Program Studi X, dapat disimpulkan bahwa arsitektur sistem telah berhasil diimplementasikan dan dapat diterima baik oleh pengguna dengan struktur basis data yang mewakili tiga aspek utama: pengguna, kompetensi, dan pekerjaan. Desain antarmuka yang telah dibangun berdasarkan peran pengguna, menggunakan pendekatan pengembangan secara bertahap yang dimulai dari fitur terlengkap (Admin) hingga fitur spesifik (Moderator, Dosen/Mitra, dan

ISSN : 1411-1624
e-ISSN: 2621-8089

Mahasiswa) juga dapat diterima baik oleh para pengguna sistem ini sesuai dengan hasil evaluasi pada tabel respon pengguna. Sedangkan untuk implementasi sisi *front-end* dan *back-end* pada tabel pengujian juga dapat menunjukkan keberhasilan arsitektur dengan penggunaan beberapa teknologi seperti Node.js, ExpressJS, dan PassportJS yang mendukung keamanan serta integritas proses sistem informasi ini.

Meskipun masih terdapat kekurangan seperti keterbatasan fitur dan penurunan performa pada kondisi tertentu seperti yang terlihat pada tabel respon pengguna, hal ini masih dapat ditoleransi oleh pengguna dengan catatan tanpa mengganggu fungsi utama sistem dan mengurangi efektifitas selama dalam penggunannya.

Dengan demikian, sistem ini dinyatakan layak untuk digunakan dan mampu memenuhi kebutuhan pengguna dalam memfasilitasi kerja paruh waktu berbasis kompetensi di lingkungan Program Studi X.

Untuk pengembangan sistem berikutnya perlu mempertimbangkan beberapa aspek penting, seperti penambahan fitur sesuai kebutuhan pengguna secara khusus, pemisahan modul khusus sehingga memudahkan dalam pengelolaan maupun pengembangan, serta melakukan integrasi sistem ke dalam kurikulum program studi dan alangkah baiknya juga terintegrasi dengan sistem akademik yang sudah berjalan. Langkah ini bertujuan agar sistem tidak hanya fungsional, tetapi juga dapat dimanfaatkan sebagai alat evaluasi proses pembelajaran di program studi tersebut.

UCAPAN TERIMA KASIH

Ucapan terima kasih yang ditujukan kepada Universitas Kristen Maranatha sebagai institusi yang telah mendukung penelitian ini dari awal hingga selesai dan juga pihak-pihak yang telah berkontribusi dalam penelitian, selain dari para penulis.

DAFTAR PUSTAKA

[1] G. G. Maulana, “Perancangan aplikasi ‘PARTIMA’ untuk sistem informasi lowongan kerja berbasis Android,” *Ensiklopedia J.*, vol. 1, no. 3, 2019. https://jurnal.ensiklopediaku.org/ojs-2.4.8-3/index.php/ensiklopedia/article/view/120

[2] K. Anita, A. D. Wahyudi, and E. R. Susanto, “Aplikasi lowongan pekerjaan berbasis web pada SMK Cahaya Kartika,” *J. Teknol. dan Sist. Inf.*, vol. 1, no. 1, pp. 75–80, 2020. https://jim.teknokrat.ac.id/index.php/sisteminformasi/article/view/213

[3] D. C. Murti, “*Sistem informasi pencari pekerjaan part time*,” Skripsi, Universitas Islam Indonesia, Yogyakarta, Indonesia, 2021. https://dspace.uii.ac.id/bitstream/handle/123456789/37447/14523158.pdf

[4] R. Brooks, D. J. Youngson, and N. Wilkin, “Undergraduate work placements: An analysis of the effects on career progression,” *Stud. High. Educ.*, vol. 41, no. 9, pp. 1563–1578, 2016. https://www.tandfonline.com/doi/full/10.1080/03075079.2015.1073249

[5] S. Shanmugam, S. S. K. Deepak, and V. S. Kumar, “Enhancing student placement preparation through web application,” *SSRN Electron. J.*, 2024. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4834698

[6] C. C. B. Bahari and Y. Sumaryana, “Penerapan progressive web apps pada aplikasi lowongan pekerjaan dosen Universitas Perjuangan,” *Inform. Digit. Expert (INDEX)*, vol. 1, no. 1, 2019. https://www.e-journal.unper.ac.id/index.php/informatics/article/view/285

ISSN : 1411-1624
e-ISSN: 2621-8089

[7] I. D. Lestari, S. Samsugi, and Z. Abidin, "Rancang bangun sistem informasi pekerjaan part time berbasis mobile di wilayah Bandar Lampung," *TELEFORTECH: J. Telemat. Inf. Technol*., vol. 1, no. 1, pp. 18–21, 2020. https://telefortech.tifauzi.com/index.php/telefortech/article/view/5

[8] S. Aastha, S. K. Sharma, and R. K. Gupta, "Web-based placement management system: A digital approach to campus hiring," J. *Adv. Database Manag. Syst*., vol. 2, no. 1, pp. 1–10, 2025. https://journals.stmjournals.com/joadms/article=2025/view=208286/

[9] R. K. Sharma, P. Jain, and S. Gupta, "A comprehensive campus recruitment and placement system for optimizing the hiring process," *Int. J. Recent Adv. Sci. Eng. Technol*., vol. 11, no. 6, pp. 123–130, 2023. https://www.ijraset.com/research-paper/comprehensive-campus-recruitment-and-placement-system

[10] A. Kumar and S. Singh, "Development of a web-based placement portal for university recruitment," *J. Inf. Syst. Technol*., vol. 5, no. 3, pp. 89–97, 2023. https://www.researchgate.net/publication/373155123_Development_of_a_Web-Based_Placement_Portal_for_University_Recruitment

[11] P. V. Rao and M. K. Desai, "Real-time analytics for campus placement systems," *Int. J. Data Sci. Anal*., vol. 9, no. 4, pp. 201–210, 2024. https://www.springer.com/journal/41060

[12] S. M. Patel and R. N. Shah, "User-centric design in web-based recruitment platforms," *J. Hum.-Comput. Interact*., vol. 12, no. 2, pp. 34–42, 2023. https://www.tandfonline.com/journals/hhci20

[13] E. Y. Huang, S. C. Lin, and T. K. Huang, "Increasing student motivation and awareness towards career opportunities through gamification," *Int. J. Inf. Educ. Technol*., vol. 9, no. 10, pp. 701–706, 2019. https://www.researchgate.net/publication/335455939_Increasing_Student_Motivation_and_Awareness_Towards_Career_Opportunities_Through_Gamification

[14] M. S. Khan and A. R. Ali, "Web-based systems for academic job placement: A case study," *J. Educ. Comput. Res*., vol. 62, no. 3, pp. 456–472, 2024. https://journals.sagepub.com/doi/10.1177/07356331231212345

[15] L. T. Nguyen and H. P. Tran, "Recommendation systems for academic job placement," *Int. J. Artif. Intell. Appl*., vol. 14, no. 5, pp. 78–90, 2023. https://www.aircccse.org/journal/ijaia/ijaia.html

[16] World Bank, "*Digital job matching platforms*," S4YE Discussion Note, 2023. https://thedocs.worldbank.org/en/doc/ceb5c5792ad0d874e9b1c3cc71362f46-0460012023/original/Digital-Job-Matching-Platforms-S4YE-Draft-Note-for-Discussion.pdf

[17] H. M. Tran and T. T. Nguyen, "The employability skills needed to face the demands of work in the future: Systematic literature reviews," *Open Eng*., vol. 10, no. 1, pp. 794–802, 2020. https://www.degruyter.com/document/doi/10.1515/eng-2020-0072/html

[18] J. Neagu, "What is a Multi Page Application (MPA) Exactly?," *Medium*, Jan. 29, 2024. [Online]. Available: https://medium.com/@julianneagu/multi-page-application-mpa-a-good-business-fit-36029c7be9f0

[19] Wahyuni, "Metode Design Thinking dalam Perancangan UI/UX Aplikasi Mobile Koperasi Modern", *SmartComp*, vol. 2024, no. 1, pp. 1-7, [Online]. Available: https://ejournal.poltekharber.ac.id/index.php/smartcomp/article/view/5493.

[20] M. A. Aziz, "Perancangan dan Implementasi Sistem Informasi Menggunakan Konsep Front-End dan Back-End," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 2, pp. 123–130, 2021.

[21] M. Rouse, "Front-end and back-end development," *TechTarget*, 2020. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/front-end-and-back-end

[22] Stack Overflow, "Node.js makes fullstack programming easy with server-side JavaScript," *Stack Overflow Blog*, Oct. 25, 2021. [Online]. Available: https://stackoverflow.blog/2021/10/25/node-js-makes-fullstack-programming-easy-

ISSN : 1411-1624
e-ISSN: 2621-8089

with-server-side-javascript/
[23] Express.js, "*Express - Node.js web application framework*," [Online]. Available: https://expressjs.com/
[24] Passport.js, "*Authentication middleware for Node.js*," [Online]. Available: https://www.passportjs.org/
[25] Oracle Corporation, "*MySQL : MySQL 8.0 Reference Manual*," [Online]. Available: https://dev.mysql.com/doc/
[26] R. Elmasri and S. B. Navathe., *Fundamentals of Database Systems*, 7th ed., Boston: Pearson, 2016.