

Pengembangan Aplikasi Mobile "MyAgenda" dengan *React Native*, *Node.js*, dan *MySQL*

Mochammad Rifky Andrianto¹, Bamaramzy Rakan Faishal², Ferdi Firdaus Ega Pratama³, Muhammad Muharrom Haromainy^{4*}

^{1,2,3,4} (Informatika, Universitas Pembangunan Nasional Veteran Jawa Timur)

¹122081010013@student.upnjatim.ac.id

²22081010056@student.upnjatim.ac.id

³22081010247@student.upnjatim.ac.id

⁴muhammad.muharrom.if@upnjatim.ac.id

⁴ (Informatika, Universitas Pembangunan Nasional Veteran Jawa Timur)

*Corresponding author email: muhammad.muharrom.if@upnjatim.ac.id

Abstrak— Di era digital saat ini, individu membutuhkan alat yang efektif untuk mengelola aktivitas sehari-hari mereka. Penelitian ini menyajikan pengembangan aplikasi "My Agenda," aplikasi mobile yang dirancang untuk membantu pengguna mengatur tugas pribadi seperti acara, tugas akademik, dan manajemen proyek. Aplikasi ini menggunakan *React Native* untuk pengembangan antarmuka lintas platform, *Node.js* dengan *Express.js* untuk layanan *backend*, dan *MySQL* sebagai *database* relasional. Sistem ini dikembangkan menggunakan model *Waterfall*, yang meliputi tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Fitur utama aplikasi ini mencakup otentikasi pengguna dengan *JSON Web Token (JWT)*, operasi *CRUD* untuk manajemen acara, serta pembaruan profil yang aman dengan validasi data dan enkripsi *password*. Pengujian fungsional dilakukan menggunakan *emulator Android* dan *Postman*, menunjukkan otentikasi yang berhasil, manajemen acara, dan pembaruan profil yang aman. Integrasi *frontend* dan *backend* melalui *RESTful API* dengan *Axios* memastikan komunikasi data yang lancar dan pengalaman pengguna yang responsif. Aplikasi ini mendukung manajemen tugas pribadi dengan antarmuka yang aman dan ramah pengguna. *React Native* memfasilitasi pengembangan lintas platform, sementara *Express.js* menyediakan struktur *backend* modular. Untuk perbaikan di masa depan, disarankan penambahan pemberitahuan pengingat, desain *UI/UX* yang lebih baik, cadangan data, dan otentikasi dua factor.

Kata Kunci: Aplikasi Mobile, *React Native*, *Node.js*, *MySQL*, Autentikasi *JWT*.

I. PENDAHULUAN

Dalam kehidupan modern yang dinamis, setiap individu dituntut untuk dapat mengelola waktu dan aktivitasnya secara efektif. Aktivitas sehari-hari yang beragam seperti pertemuan, tugas akademik, hingga pekerjaan proyek seringkali sulit dikelola tanpa sistem yang terorganisir. Permasalahan umum yang sering muncul adalah lupa jadwal, tumpang tindih kegiatan, dan ketidakteraturan dalam pencatatan aktivitas. Oleh karena itu, dibutuhkan suatu sistem digital yang dapat membantu pengguna dalam mencatat, mengatur, dan mengingatkan aktivitas penting secara efisien dan terstruktur. Penelitian oleh Suhartono (2016) menunjukkan bahwa penggunaan aplikasi agenda digital dapat membantu

mahasiswa dalam mengatur jadwal kegiatan akademik secara lebih efektif [1, 4].

Seiring dengan perkembangan teknologi *mobile*, kebutuhan akan aplikasi yang praktis dan mudah diakses semakin meningkat. Mutiwiayati. (2023) juga membuktikan bahwa aplikasi *mobile* berbasis *React Native* dapat berfungsi dengan baik dalam menyampaikan materi dan informasi secara efisien, mendukung urgensi pengembangan sistem yang responsif dan fleksibel. Hal ini menjadi dasar bagi pengembangan aplikasi MyAgenda sebagai solusi untuk membantu pengguna dalam pengelolaan aktivitas pribadi berbasis *mobile* [3, 5, 6]. Penelitian sebelumnya menunjukkan bahwa penggunaan aplikasi *mobile* dapat meningkatkan efisiensi dalam mengelola agenda pribadi. Suhartono (2016) mengembangkan aplikasi pengaturan agenda perkuliahan pribadi mahasiswa berbasis *Android* dan menemukan bahwa aplikasi tersebut memiliki tingkat kelayakan penggunaan sebesar 86,44% berdasarkan evaluasi dari 15 responden [1].

Dalam pengembangan aplikasi *mobile*, pemilihan teknologi yang tepat sangat krusial. *React Native*, sebagai framework *open-source* yang dikembangkan oleh Facebook, memungkinkan pengembangan aplikasi *mobile* lintas platform dengan efisiensi tinggi. Hal ini didukung oleh penelitian Mutiwiayati (2023), yang menunjukkan bahwa aplikasi berbasis *React Native* dapat berjalan dengan lancar pada berbagai perangkat *android* [3]. Selain itu, penggunaan *Node.js* dengan *Express* sebagai *backend* memberikan fleksibilitas dalam membangun API yang efisien dan skalabel [2, 10].

Praditta dan Voutama (2025) menunjukkan bahwa sistem *to-do list* berbasis *web* yang dikembangkan dengan *React.js* sebagai antarmuka dan *MongoDB* sebagai basis data menunjukkan peningkatan efisiensi dalam pengelolaan tugas dibandingkan metode konvensional. Hasil *usability testing* dengan 10 responden menghasilkan skor *SUS* sebesar 88%, yang mengindikasikan sistem memiliki tingkat *usability* yang sangat baik. Pengguna merasakan peningkatan efisiensi dalam pencatatan, pembaruan, dan pemantauan tugas [2].

Mengembangkan model sistem pengelolaan agenda pribadi berbasis *mobile* untuk mengatasi kendala dalam pencatatan dan pengaturan aktivitas harian pengguna. Sistem ini memungkinkan akses fleksibel melalui perangkat seluler serta integrasi dengan fitur autentikasi dan manajemen data, sehingga memudahkan pencatatan, pembaruan, dan

pengendalian aktivitas secara mandiri. Sistem serupa juga telah dikembangkan oleh Suhartono (2016), yang menunjukkan bahwa penggunaan aplikasi agenda berbasis Android dapat meningkatkan efisiensi pengelolaan waktu bagi mahasiswa [3]. Dengan mempertimbangkan studi-studi tersebut, pengembangan aplikasi "MyAgenda" bertujuan untuk menyediakan platform yang dapat membantu pengguna dalam mengelola berbagai aspek kehidupan mereka secara terorganisir dan efisien. Aplikasi ini diharapkan dapat menjadi alat yang berguna bagi individu dalam merencanakan dan melacak aktivitas mereka sehari-hari.

II. METODOLOGI PENELITIAN

2.1. KAJIAN PUSTAKA

Penelitian terkait pengembangan aplikasi manajemen agenda telah banyak dilakukan. Suhartono (2016) mengembangkan aplikasi pengelolaan agenda perkuliahan berbasis Android yang ditujukan untuk mahasiswa dalam mencatat dan mengatur jadwal kegiatan akademik. Aplikasi tersebut membantu pengguna mengelola aktivitas secara lebih efisien, namun hanya tersedia dalam platform Android dan menggunakan pendekatan prototyping [1, 8].

Praditta dan Voutama (2025) membahas sistem to-do list berbasis *website* menggunakan *React.js* dan *Express.js*. Sistem ini mendukung pencatatan tugas berbasis *web* dengan tampilan antarmuka yang interaktif serta arsitektur REST API untuk mendukung modularitas *backend* [2, 10].

Sementara itu, Mutiwiati (2023) mengembangkan aplikasi e-learning berbasis *React Native* dan *Express.js*. Penelitian ini menunjukkan bahwa penggunaan *stack* teknologi tersebut menghasilkan aplikasi yang berjalan baik pada berbagai perangkat Android, dengan performa yang stabil dan antarmuka responsif [3, 7].

Dari studi tersebut, dapat disimpulkan bahwa aplikasi agenda berbasis digital terbukti bermanfaat dalam membantu pengguna mengatur aktivitas. Namun, penelitian sebelumnya masih memiliki keterbatasan seperti cakupan platform yang terbatas, tidak adanya fitur manajemen tugas pribadi secara menyeluruh, atau belum terintegrasi dengan sistem autentikasi modern. Oleh karena itu, penelitian ini mengembangkan aplikasi "MyAgenda" berbasis mobile dengan pendekatan lintas platform menggunakan *React Native*, *backend Express.js*, dan basis data *MySQL* untuk memberikan solusi yang lebih fleksibel dan aman.

2.2. PENDEKATAN DAN METODE PENGEMBANGAN

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak untuk mengembangkan aplikasi My Agenda, yang berfungsi sebagai media pencatatan kegiatan, tugas proyek, dan aktivitas akademik. Untuk mendukung pengembangan yang terstruktur dan terarah, dipilih metode pengembangan sistem *Waterfall*. Metode ini dianggap tepat karena menyediakan tahapan yang sistematis dan mudah diikuti, mulai dari analisis kebutuhan hingga pemeliharaan sistem. *Waterfall* dipilih

karena memberikan struktur yang jelas dalam setiap tahap pengembangan, memungkinkan dokumentasi yang lengkap dan kontrol yang lebih baik pada setiap tahap. Selain itu, dalam proses pengembangan digunakan berbagai teknologi modern yang mendukung efisiensi dan kompatibilitas lintas platform.

1. Metodologi Pengembangan Sistem

Pengembangan aplikasi My Agenda menggunakan metode *Waterfall*, yang terdiri dari tahapan berurutan: analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Metode ini dipilih karena memberikan struktur yang jelas dalam setiap tahap pengembangan. Metode *Waterfall* juga digunakan dalam sistem informasi agenda pimpinan, karena cocok untuk sistem dengan ruang lingkup yang jelas dan kebutuhan yang terdokumentasi sejak awal.

2. Teknologi yang Digunakan

Frontend: Aplikasi dibangun menggunakan *React Native*, karena mampu mempercepat proses pengembangan aplikasi mobile lintas platform serta efisien dalam pengelolaan komponen *UI*. *Backend* dikembangkan dengan *Node.js* dan *Express.js* untuk membangun REST API yang ringan, fleksibel, dan mudah diintegrasikan. *Database* menggunakan *MySQL* dipilih karena memiliki keandalan tinggi, performa stabil, dan mendukung query yang kompleks untuk pengelolaan data agenda.

3. Tahapan Pengembangan

Dalam penerapan metode *Waterfall*, proses pengembangan sistem dilakukan secara bertahap dan berurutan. Setiap tahap memiliki peran penting dalam memastikan bahwa aplikasi yang dibangun sesuai dengan kebutuhan pengguna dan berfungsi dengan baik. Adapun tahapan pengembangan dalam proyek aplikasi My Agenda dimulai dari analisis kebutuhan hingga tahap pemeliharaan sistem. Berikut adalah penjelasan masing-masing tahapan:

- a. Analisis Kebutuhan: Proses analisis kebutuhan dilakukan dengan mengidentifikasi permasalahan umum yang dialami oleh pengguna dalam mengelola aktivitas harian, seperti jadwal perkuliahan, tugas proyek, dan agenda pribadi. Pengumpulan kebutuhan dilakukan melalui studi literatur terhadap penelitian sebelumnya dan observasi terhadap kebiasaan pencatatan manual yang kurang efisien. Hasil analisis ini menghasilkan kebutuhan utama berupa fitur pencatatan acara, manajemen tugas akademik dan proyek, serta sistem autentikasi pengguna yang dirancang dalam struktur sistem terintegrasi. [2, 8].
- b. Perancangan Aplikasi: Pada tahap perancangan aplikasi, dilakukan penyusunan model sistem menggunakan beberapa diagram untuk menggambarkan proses dan struktur yang akan dibangun, yaitu *use case diagram* untuk menunjukkan interaksi antara pengguna dan sistem, *activity diagram* untuk memodelkan alur aktivitas pengguna dalam menggunakan fitur aplikasi, serta *entity relationship*

diagram (ERD) untuk merancang struktur basis data yang digunakan untuk menyimpan data pengguna, tugas, dan jadwal kegiatan. Ketiga diagram ini berfungsi sebagai dasar perancangan sistem sebelum tahap implementasi dilakukan, agar pengembangan aplikasi dapat dilakukan secara sistematis, terarah, dan sesuai kebutuhan pengguna. [3, 5].

- c. Implementasi: Implementasi sistem dilakukan dengan pendekatan komponen *reusable* di *React Native* dan pemanggilan REST API menggunakan *Axios*.
- d. Pengujian: Menggunakan metode *Black-Box testing*, yaitu dengan menguji fungsionalitas aplikasi berdasarkan *input* dan *output* tanpa melihat struktur internal kode program. Pengujian ini bertujuan untuk memastikan bahwa setiap fitur dalam aplikasi, seperti pencatatan agenda, pengelolaan tugas, pengeditan, dan penghapusan data, dapat berjalan sesuai dengan rancangan. Selain itu, pengujian juga dilakukan pada berbagai perangkat Android untuk memastikan kompatibilitas dan stabilitas aplikasi di berbagai kondisi penggunaan.

Tabel. 1 *Black-Box Testing*

No	Nama Fitur	Input	Output yang Diharapkan	Hasil
1.	Registrasi Akun	Nama, email, dan kata sandi	Akun baru berhasil dibuat dan dapat <i>login</i>	Berhasil
2.	Login	Email dan kata sandi valid	Pengguna berhasil masuk ke dalam aplikasi	Berhasil
3.	Tambah Agenda	Judul, tanggal, waktu	Data agenda tersimpan dan tampil dalam daftar	Berhasil
4.	Edit Tugas Proyek	Perubahan judul atau <i>deadline</i>	Data tugas diperbarui di sistem	Berhasil
5.	Hapus Agenda	Tekan tombol hapus pada daftar agenda	Data agenda terhapus dari sistem	Berhasil
6.	Lihat Daftar Tugas	Akses halaman daftar tugas	Menampilkan semua tugas yang disimpan	Berhasil
7.	Ubah Kata Sandi	Kata sandi lama dan sandi baru valid	Kata sandi berhasil diperbarui dan dapat <i>login</i>	Berhasil

- e. Pemeliharaan: Setelah aplikasi dirilis, pemeliharaan dilakukan berdasarkan umpan balik pengguna untuk memperbaiki *bug* dan menambahkan fitur baru.

4. Integrasi dan Komunikasi Data

Integrasi *frontend* dan *backend* dilakukan dengan metode *REST API*. *Express.js* digunakan untuk membuat *endpoint* dan pengaturan *routing*, sedangkan *Axios* digunakan untuk pengambilan data di sisi *frontend*.

III. HASIL DAN PEMBAHASAN

3.1. HASIL PENGEMBANGAN SISTEM

Dalam pengembangan aplikasi My Agenda, fokus utama adalah menciptakan sebuah sistem pencatatan agenda pribadi yang mudah digunakan dan efisien untuk mendukung berbagai aktivitas pengguna sehari-hari. Aplikasi ini dirancang agar mampu mengelola data acara (*event*) secara dinamis melalui operasi CRUD, serta menyediakan fitur autentikasi pengguna dan kemampuan untuk memperbarui informasi profil dengan aman. Dengan memanfaatkan teknologi *React Native* pada sisi *frontend* dan *Node.js Express* pada *backend*, serta basis data *MySQL*, sistem ini diharapkan dapat memberikan pengalaman pengguna yang responsif dan handal. Berikut ini dijelaskan hasil implementasi dari sistem yang telah dikembangkan beserta pembahasannya secara terperinci.

1. Hasil Implementasi Sistem

Setelah melalui tahapan analisis, perancangan, implementasi, dan pengujian, aplikasi My Agenda berhasil dikembangkan dengan berbagai fitur utama yang mendukung kebutuhan pengguna dalam mengelola agenda harian. Implementasi sistem difokuskan pada fungsionalitas inti seperti pengelolaan *event*, autentikasi pengguna, serta pengaturan profil. Setiap fitur dirancang agar mudah digunakan, responsif, dan aman, dengan dukungan teknologi *frontend* dan *backend* yang telah dipilih. Berikut adalah hasil implementasi dari fitur-fitur utama dalam aplikasi:

a. Form Event Crud

Gbr. 2 *CRUD* Form Acara

Aplikasi My Agenda menyediakan fitur lengkap untuk mengelola agenda berupa form *event* yang memungkinkan pengguna melakukan operasi *CRUD* (*Create, Read, Update, Delete*). Pengguna dapat menambahkan *event* baru dengan mengisi nama acara, tanggal, lokasi, dan deskripsi. Data *event* ini kemudian disimpan ke dalam database *MySQL*. Pengguna juga dapat melihat daftar *event* yang tersimpan, memperbarui informasi *event* apabila terdapat perubahan,

dan menghapus *event* yang tidak lagi diperlukan. Implementasi *CRUD* ini didukung oleh *API RESTful* dengan metode Validasi data juga diterapkan untuk memastikan *input* yang benar dan mencegah error pada *database*.

b. Fitur Login

Gbr. 3 Login

Halaman *login* pada aplikasi ini berfungsi sebagai mekanisme autentikasi pengguna menggunakan email dan *password*. Ketika pengguna mengirimkan data *login*, *backend* memverifikasi kredensial tersebut dan mengembalikan *token JWT* jika valid. *Token* ini berfungsi sebagai bukti autentikasi yang diperlukan untuk mengakses fitur-fitur aplikasi seperti pengelolaan *event* dan update profil. Sistem *login* ini memastikan hanya pengguna yang terdaftar dan terautentikasi yang dapat menggunakan aplikasi, sehingga menjaga keamanan data pribadi pengguna.

c. Fitur Register

Gbr. 4 Register

Halaman *register* memungkinkan pengguna baru untuk mendaftar dan membuat akun pada aplikasi My Agenda. Pengguna diharuskan untuk mengisi informasi pribadi mereka seperti nama, email, dan kata sandi untuk membuat akun. Setelah mengisi formulir registrasi, data pengguna akan disimpan di *database MySQL* dan dikirimkan ke

backend menggunakan *API* yang dibangun dengan *Node.js* dan *Express*. Proses pendaftaran juga dilengkapi dengan validasi *input* untuk memastikan bahwa data yang dimasukkan benar dan terformat dengan baik.

d. Update-profile

Gbr. 5 Update Profil

Fitur update profil memungkinkan pengguna mengubah informasi akun mereka, termasuk *username*, email, dan *password*. Perubahan ini dapat dilakukan melalui form khusus dan dikirim ke *backend* melalui *endpoint* yang telah disediakan. Sistem melakukan validasi pada *input*, terutama untuk *password* yang harus dienkripsi sebelum disimpan di *database* guna menjaga keamanan. Dengan adanya fitur ini, pengguna dapat mengelola data pribadi mereka secara mandiri, meningkatkan kenyamanan dan keamanan penggunaan aplikasi.

2. Pengujian Fungsional

Setelah proses implementasi selesai, dilakukan pengujian fungsional untuk memastikan bahwa setiap fitur dalam aplikasi My Agenda berjalan sesuai dengan yang dirancang. Pengujian ini bertujuan untuk mengevaluasi apakah sistem dapat merespons *input* pengguna dengan benar dan memberikan *output* yang sesuai. Pengujian dilakukan pada sisi *frontend* menggunakan *emulator Android*, serta pada sisi *backend* menggunakan Postman untuk menguji berbagai *endpoint API*. Hasil pengujian menunjukkan bahwa fitur-fitur utama telah berfungsi secara optimal, seperti dijelaskan berikut ini:

- Fitur *login* berhasil mengautentikasi pengguna dengan mengembalikan *token JWT* yang valid.

- b. *CRUD* pada form *event* berjalan sesuai ekspektasi, dimana pengguna dapat membuat, melihat, mengupdate, dan menghapus data *event* tanpa kendala.
- c. *Update profile* diuji dengan mengubah *username*, email, dan *password*, dan perubahan tersebut berhasil tersimpan dengan aman di *database*.

3.2. PEMBAHASAN

Setelah proses implementasi dan pengujian selesai, dilakukan pembahasan untuk mengevaluasi efektivitas penggunaan teknologi serta kualitas sistem yang dibangun. Pembahasan ini mencakup aspek performa, keamanan, kemudahan pengembangan, dan kesesuaian aplikasi terhadap kebutuhan pengguna. Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa pemilihan arsitektur dan teknologi yang digunakan memberikan dampak positif terhadap kinerja dan fungsionalitas sistem. Berikut adalah beberapa poin penting dari hasil pembahasan:

- a. Penggunaan *React Native* dalam pengembangan aplikasi memberikan kemudahan dalam membangun aplikasi lintas platform karena memungkinkan penggunaan satu basis kode untuk berbagai perangkat.
- b. *Backend* dikembangkan menggunakan *Express (Node.js)* karena kerangka kerja ini mendukung pembuatan API yang modular dan mudah dikembangkan.
- c. *JSON Web Token (JWT)* diimplementasikan untuk keperluan autentikasi pengguna sebagai bagian dari pengamanan akses terhadap data pribadi dalam aplikasi.
- d. Secara keseluruhan, aplikasi *My Agenda* mampu memenuhi kebutuhan dasar pencatatan agenda pribadi serta pengelolaan data pengguna secara aman dan efisien.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil pengembangan aplikasi *My Agenda*, sistem ini berhasil memenuhi kebutuhan pengguna dalam mencatat dan mengelola agenda pribadi secara efektif melalui fitur *CRUD* pada form *event* yang memungkinkan pembuatan, pembacaan, pembaruan, dan penghapusan data acara dengan mudah. Sistem autentikasi menggunakan *token JWT* berjalan dengan baik, memberikan keamanan akses yang memadai, serta fitur *update profil* yang mencakup perubahan *username*, email, dan *password* telah diimplementasikan dengan validasi dan enkripsi yang menjaga keamanan data pribadi pengguna. Penggunaan *React Native*, *Node.js Express*, dan *MySQL* terbukti efektif dalam membangun aplikasi mobile yang responsif dan handal. Namun, untuk pengembangan selanjutnya, disarankan menambahkan fitur notifikasi pengingat agenda, meningkatkan tampilan *UI/UX* agar lebih menarik dan intuitif, serta menambahkan fitur *backup* data untuk menjaga keamanan dan ketersediaan data pengguna. Selain itu, penerapan autentikasi dua faktor dan pengujian performa aplikasi secara berkala juga perlu dilakukan untuk meningkatkan keamanan dan memastikan aplikasi tetap optimal saat digunakan dalam skala yang lebih besar.

REFERENSI

- [1] S. Suhartono, "Aplikasi Pengaturan Agenda Perkuliahan Pribadi Mahasiswa Menggunakan Smartphone Berbasis Android," *J. Teknol. Elektroika*, vol. 13, no. 2, p. 164, 2016, doi: 10.31963/elektroika.v13i2.984.
- [2] N. Praditta and A. Voutama, "Rancang Bangun Sistem To-Do List Berbasis *Website* dalam Efisiensi Pengelolaan Tugas," vol. 13, no. 1, 2025.
- [3] M. Mutiawati, R. Syaban, E. Kurniasari, and O. Oktaviani, "Aplikasi E-Learning Berbasis Android sebagai Media Pembelajaran Bahasa Pemrograman," *J. Kridatama Sains Dan Teknol.*, vol. 5, no. 01, pp. 161–175, 2023, doi: 10.53863/kst.v5i01.720.
- [4] Griffith, J., Monkman, H., Penner, S., Karoli, K., & Stockdale, C. (2024). An iterative approach to usability evaluation: A workplace wellness mobile application example. *Digital Health and Informatics Innovations for Sustainable Health Care Systems*, 449–455.
- [5] Prajapati, V. Y., & Bhattacharjee, S. (2025). React Native: A comprehensive analysis of cross-platform mobile application development. *International Journal for Research Trends and Innovation*, 10(3), 359–362. <http://www.ijrti.org/papers/IJRTI2503151.pdf>
- [6] Kuniawan, A. A., & Yulhendri. (2023). Pemanfaatan framework React Native dalam perancangan aplikasi penjualan merchandise. *Nucleus: Journal of Technology and Information Systems*, 4(2), 84–87. <https://doi.org/10.37010/nuc.v4i2.1369>
- [7] Weichbroth, P. (2024). Usability testing of mobile applications: A methodological framework. *Applied Sciences*, 14(5), 1792. <https://doi.org/10.3390/app14051792>
- [8] Islam, M. N., Karim, M. M., Inan, T. T., & Islam, A. K. M. N. (2020). Investigating usability of mobile health applications in Bangladesh. *BMC Medical Informatics and Decision Making*, 20, Article 19. <https://doi.org/10.1186/s12911-020-1033-3>
- [9] Yasykur, M. W., & Efrilianda, D. A. (2024). Evaluation of user interface and user experience on Solo Destination app using system usability scale and human-centered design methods. *Journal of Advances in Information Systems and Technology*, 6(2), 183–199. <http://creativecommons.org/licenses/by-sa/4.0/>
- [10] Hussain, A., Mkpjoiguo, E. O. C., & Kamal, F. M. (2016). A systematic review on usability evaluation methods for M-Commerce apps. *Journal of Telecommunication, Electronic and Computer Engineering*, 8(10), 29–33.