Jurnal Sains Komputer Dan Sistem Informasi
Volume 2 Nomor 1, Juni 2024 ; Page 90-96
E-ISSN : 2987-5684; P-ISSN : 3025-5457

"

Penggunaan Teknologi Mobile Cross-Platform dalam Aplikasi Bergerak: Tinjauan Terhadap Keuntungan dan Kendala

[1]M. Iqbal Hasani, [2]Muhamad Imam Ilham, [3]Muhammad Daffa Firmansyah, [4]Yoga Sahria
[1,2,3,4]Informatika, Universitas Teknologi Yogyakarta
Email: [1]miqbalhasani07@gmail.com, [2]muhamad.imamilham@gmail.com, [3]daffafirmansyah0@gmail.com, [4]yogasahria@amikom.ac.id
Email Penulis Korespondensi: daffafirmansyah0@gmail.com

Article History:
Received Jan 11th, 2024
Revised Jun 10th, 2024
Accepted Jun 21th, 2024

Abstrak

Aplikasi mobile bergerak telah menjadi elemen integral dalam kehidupan sehari-hari, memungkinkan akses mudah ke berbagai layanan dan informasi. Dalam upaya untuk mengoptimalkan pengembangan aplikasi, teknologi cross platform telah muncul sebagai alternatif yang menjanjikan. Penelitian ini memberikan tinjauan mendalam mengenai penerapan teknologi cross platform dalam pengembangan aplikasi mobile bergerak. Ini mencakup analisis keuntungan seperti efisiensi pengembangan, penghematan biaya, dan kemudahan pemeliharaan aplikasi lintas platform, serta kendala seperti keterbatasan dalam akses ke fitur platform tertentu dan performa yang mungkin tidak seoptimal aplikasi native. Dengan demikian, artikel ini berkontribusi pada pemahaman yang lebih baik tentang perkembangan teknologi dalam dunia mobile yang terus berkembang.

Kata Kunci : Cross platform, Aplikasi bergerak, Teknologi Mobile

Abstract

Mobile applications have become an integral element in everyday life, allowing easy access to a wide range of services and information. In an effort to optimize application development, cross platform technology has emerged as a promising alternative. This research provides an in-depth review of the application of cross platform technology in mobile application development. This includes an analysis of benefits such as development efficiency, cost savings, and ease of maintenance of cross-platform applications, as well as constraints such as limitations in access to certain platform features and performance that may not be as optimal as native applications. As such, this article contributes to a better understanding of technological developments in an ever-evolving mobile world.

Keyword : *Cross platform, Mobile apps, Mobile Technology*

1. PENDAHULUAN

Dalam era modern yang didominasi oleh teknologi, aplikasi bergerak telah menjadi bagian penting dari kehidupan manusia sehari hari untuk menyelesaikan berbagai masalah yang ada. Kemajuan dalam teknologi pengembangan aplikasi bergerak telah membuka akses bagi teknologi cross-platform yang memungkinkan pengguna untuk menciptakan aplikasi yang dapat berjalan diberbagai platform seperti iOS dan Android. Perkembangan ini tidak hanya menguntungkan pengguna dengan memberikan akses yang lebih luas ke berbagai aplikasi, tetapi juga memberikan peluang bagi pengembang untuk menciptakan solusi yang lebih inklusif dan efisien. Dengan teknologi *cross-platform*, pengguna dari berbagai sistem operasi dapat dengan mudah mengakses dan berinteraksi dengan aplikasi yang sama, mengurangi hambatan dan meningkatkan keterjangkauan teknologi. Selain itu, pengembang dapat menghemat waktu dan sumber daya dengan mengembangkan satu aplikasi yang dapat dijalankan di beberapa platform sekaligus. Hal ini memungkinkan inovasi teknologi bergerak untuk mencapai lebih banyak orang, menciptakan lingkungan yang lebih terhubung dan inklusif dalam kehidupan sehari-hari kita [1], [2].

Teknologi *cross – platform* merupakan suatu trend dalam pengembangan suatu aplikasi dimana suatu aplikasi dapat berjalan pada beberapa sistem operasi atau *platform*, ini dapat mencakup perangkat *mobile*, komputer dan perangkat

lainya dengan teknologi *Cross-Platform* memungkinkan developer mengembangkan dan mendistribusikan aplikasi yang sama di berbagai platform dengan kode yang sama dengan munculnya beberapa *framework* seperti React Native, flutter atau Xamarin *framework* ini memiliki kekurangan dan kelebihannya tergantung dengan kebutuhan spesifik proyek dengan cross-platform dapat mengurangi waktu dan usaha dalam sebuah *development* sebuah aplikasi, serta membuat pemeliharaan aplikasi jadi lebih mudah, dan membuat aplikasi tersedia untuk lebih banyak pengguna [3], [4], [5], [6].

Meskipun ada banyak keuntungan dalam penggunaan teknologi *cross – platform*, teknologi ini juga memiliki beberapa kekurangan dan Batasan utama seperti tidak dapat mengakses fitur tertentu dari platform tertentu seperti android dan ios yang memiliki antarmuka pengguna yang berbeda, interaksi, dan perilaku sistem yang berbeda. Dalam beberapa situasi tertentu penggunaan teknologi *cross platform* tidak memberikan akses penuh ke fitur tertentu atau memungkinkan membutuhkan lebih banyak pekerja untuk mengintegrasikannya selain itu, kemungkinan kinerjanya akan menurun sedikit dibandingkan native. Akibatnya pengguna harus mempertimbangkan kebutuhan khusus aplikasi sebelum memutuskan untuk menggunakan teknologi lintas platform [7], [8], [9], [10].

Oleh karena itu, tujuan dari artikel ini akan membahas mengenai pentingnya penggunaan teknologi cross-platform dalam pengembangan aplikasi bergerak dan membahas sejauh mana keuntungan dan hambatan dalam teknologi cross-platform dapat mempengaruhi hasil akhir sebuah aplikasi. Pengembangan aplikasi bergerak adalah tahap yang penting dalam menciptakan solusi teknologi yang efisien dan efektif. Seiring dengan meningkatnya kebutuhan akan aplikasi yang berjalan diberbagai platform, penting bagi para pengembang untuk memahami kompleksitas penggunaan teknologi *cross-platform*. Melalui pemahaman yang lebih mendalam tentang pengalaman praktis, pembaca akan memiliki pandangan yang lebih baik tentang kelebihan dan kelemahan yang mungkin dihadapi saat mengembangkan aplikasi bergerak cross-platform. Dengan berbagai studi kasus dan pemahaman yang mendalam, pembaca akan memiliki landasan yang kuat untuk membuat keputusan yang bijak saat memilih pendekatan pengembangan yang paling sesuai dengan proyek yang akan dibuat. Dengan begitu, hasil akhirnya akan memenuhi harapan dan kebutuhan pengguna dengan lebih baik, menciptakan solusi yang efisien dan memuaskan.

2. METODOLOGI PENELITIAN

2.1 Metode

Pada penelitian ini menggunakan metode *Systematic Literature Review* (SLR). *Systematic literature review* merupakan istilah yang digunakan untuk merujuk pada metodologi penelitian atau riset tertentu dan pengembangan yang dilakukan untuk mengumpulkan serta mengevaluasi penelitian yang terkait pada fokus topik tertentu [11]. Penelitian SLR dilakukan untuk berbagai tujuan, diantaranya untuk mengidentifikasi, mengkaji, mengevaluasi, dan menafsirkan semua penelitian yang tersedia dengan bidang topik fenomena yang menarik, dengan pertanyaan penelitian tertentu yang relevan [12], [13], [14]. Penelitian ini dimulai dari pemilihan *topic*, mengumpulkan data, menilai kualitas literatur yang ditemukan, mengorganisir dan menyintesis temuan penelitian.

2.2 Pengumpulan Data

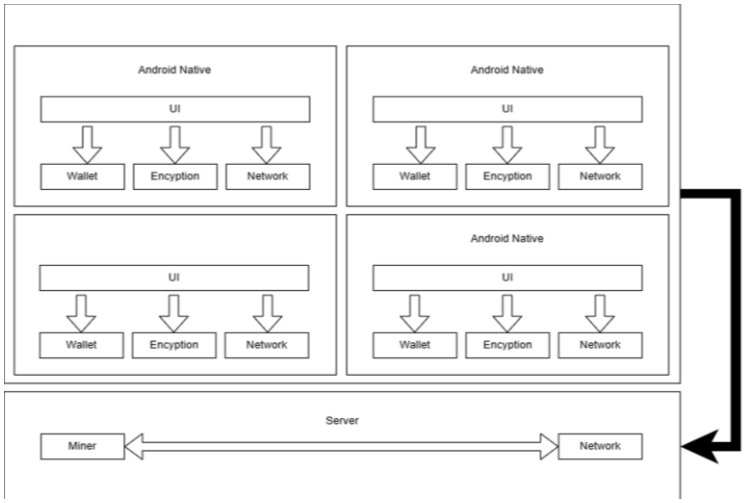
Berisi Metodologi eksperimental diterapkan dalam penelitian ini. Penelitian ini mengembangkan sebuah contoh proyek praktis, sebuah aplikasi Coin Wallet berdasarkan rantai-blok, untuk menjawab pertanyaan penelitian. Aplikasi sampel diimplementasikan dalam Android native dan tiga kerangka kerja lintas platform yang berbeda. Tabel 1 menunjukkan data yang akan dikumpulkan dari setiap aplikasi. Sebagian besar data, seperti Desain UI, fungsionalitas, dan beban kerja dikumpulkan melalui proses pengembangan, sedangkan data kinerja dikumpulkan menggunakan "mobileperf", yang merupakan alat sumber terbuka untuk memantau kinerja, seperti penggunaan CPU dan memori.

2.3 Design

Coin Wallet adalah aplikasi keuangan yang berjalan di ponsel yang memungkinkan pengguna mentransfer uang mereka kepada orang lain. Fitur-fiturnya, seperti UI, komunikasi jaringan, dan berbagai algoritma, dikembangkan dalam proyek sampel. Proyek ini berisi sisi server dan empat sisi klien (lihat Gambar 1). Sisi server adalah layanan daemon yang berjalan di stasiun kerja untuk mengirimkan data blok ke klien. Sisi klien mencakup empat aplikasi yang masing-masing diimplementasikan oleh Android asli, React Native, Flutter, dan Xamarin. Untuk memastikan proyek dapat diuji, sisi server dan sisi klien dikerahkan dalam satu subnet.

Jurnal Sains Komputer Dan Sistem Informasi
Volume 2 Nomor 1, Juni 2024 ; Page 90-96
E-ISSN : 2987-5684; P-ISSN : 3025-5457

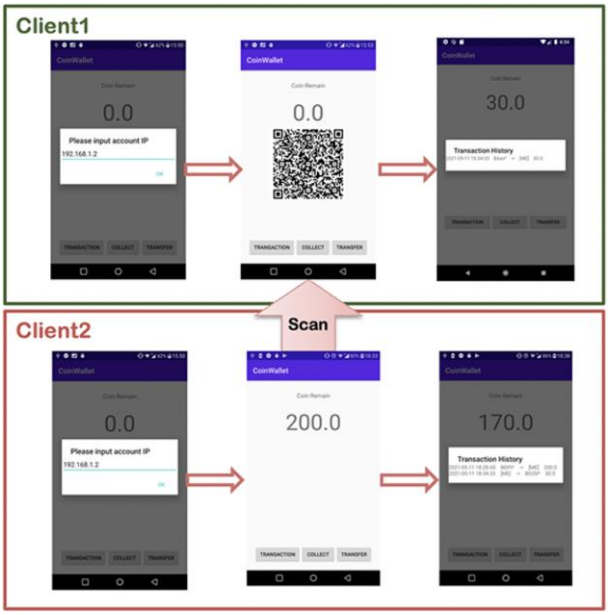
"



Gambar 1. Alur kerja arsitektur Server-Klien

Setiap aplikasi klien terdiri dari dompet, algoritma enkripsi, dan implementasi jaringan. Modul UI bertanggung jawab atas fungsi terkait rendering layar dengan memunculkan notifikasi, menerima teks masukan, dan menyajikan elemen grafis yang kaya pada ponsel. Dompet bertanggung jawab untuk mentransfer atau mengumpulkan uang dari orang lain. Modul enkripsi perlu menghasilkan, menyimpan, dan menghitung kunci di telepon. Modul jaringan merangkum RESTful API untuk memudahkan komunikasi antar klien.

Contoh alur kerja untuk mentransfer uang sebesar $30 dari Klien2 ke Klien1 menggunakan contoh proyek ada pada Gambar 2. Pertama-tama, kedua klien perlu memasukkan alamat IP server untuk memastikan semua aplikasi berkomunikasi melalui server dengan benar. Kedua, penerima (Klien1) harus bersiap dengan mengklik tombol “Kumpulkan” untuk menunjukkan akunnya (kode QR). Ketiga, pengirim (Klien2) siap bertransaksi dengan saldo awalnya (misalnya $200) dengan mengklik tombol “Transfer”; memasukkan jumlah (misalnya $30); dan memindai kode QR pada penerima (Klien1) untuk mentransfer uang. Keempat, penerima (Klien1) menerima uang dan saldo diperbarui pada kedua klien (misalnya, saldo Klien1 adalah $30; saldo Klien2 adala h $170 = $200 - $30). Terakhir, kedua klien mengklik tombol “Transaksi” untuk memeriksa pencatatan transaksi mereka.



Gambar 2. Alur kerja aplikasi

Jurnal Sains Komputer Dan Sistem Informasi
Volume 2 Nomor 1, Juni 2024 ; Page 90-96
E-ISSN : 2987-5684; P-ISSN : 3025-5457

"

3. HASIL DAN PEMBAHASAN

3.1 Beban Kerja

Seperti yang ditunjukkan Tabel 2, Android Native mengambil beban kerja tertinggi dalam baris kode daripada Tiga kerangka kerja lintas platform lainnya. Flutter, menggunakan Dart dalam sintaks paling ringkas, membutuhkan beban kerja terendah yang hanya 47% dari total beban kerja dari Android Native. Meskipun biaya React Native sedekat itu dengan Flutter dalam skala, Xamarin menghasilkan beban kerja yang lebih tinggi dari keduanya.

Beban Kerja	**Android Native**	**React Native**	**Flutter**	**Xamarin**
Java	600	0	0	0
JavaScript	0	445	0	0
C#	0	0	0	524
XML	317	0	0	115
JSON	0	58	0	0
Dart	0	0	432	0
Total	917	503	432	639

Tabel 2. Alur kerja aplikasi

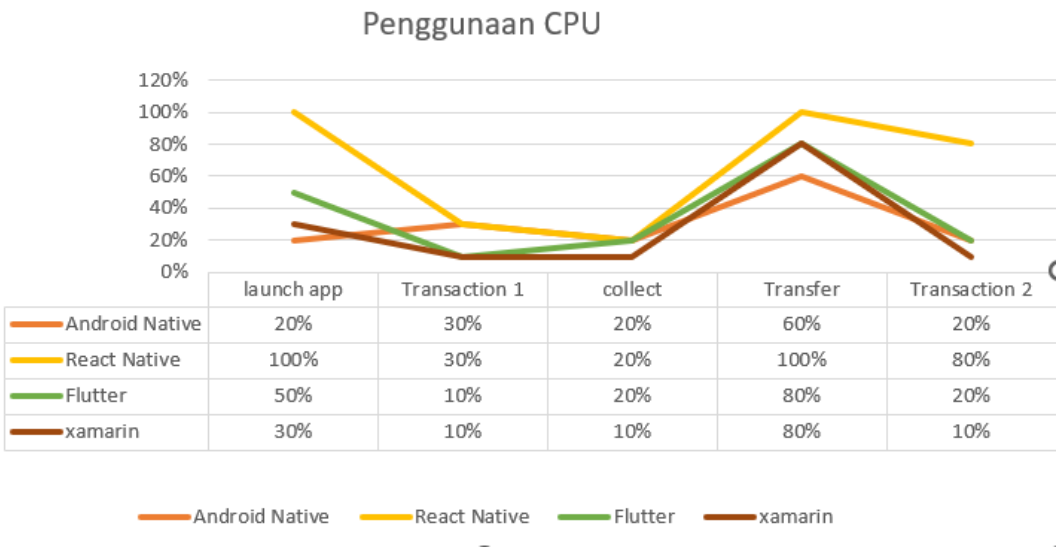
3.2 Performa

Mengenai kinerja menunjukkan pada Tabel 3 dan Gambar 3, penelitian ini mengungkapkan bahwa Android Pengembangan asli membutuhkan penggunaan CPU terendah, penggunaan memori terkecil, dan Paket biner paling kompak. Juga, penelitian ini menemukan bahwa Xamarin berjalan lebih efisien tidak hanya dalam penggunaan CPU, tetapi juga dalam penggunaan memori dengan biner terkecil, sedangkan React Native memiliki kinerja terendah dengan paket biner terbesar.

Tabel 3. Alur kerja aplikasi

Ukuran	Android Native	React Native	Flutter	Xamarin
Penggunaan Memori	82.67MB	177.92MB	167.45MB	99.78MB
Ukuran Aplikasi	3.3MB	30.1MB	21.3MB	12.9MB

Jurnal Sains Komputer Dan Sistem Informasi
Volume 2 Nomor 1, Juni 2024 ; Page 90-96
E-ISSN : 2987-5684; P-ISSN : 3025-5457



	launch app	Transaction 1	collect	Transfer	Transaction 2
Android Native	20%	30%	20%	60%	20%
React Native	100%	30%	20%	100%	80%
Flutter	50%	10%	20%	80%	20%
xamarin	30%	10%	10%	80%	10%

Gambar 3. Chart Pengunaan CPU

Selain itu, penelitian ini mengeksplorasi bahwa prosedur pengembangan yang berbeda digunakan di antara lintas platform ini. Menggunakan Android Studio melalui komponen plug-in, React Native membagi lingkungan pengembangannya menjadi satu alat. Namun, pengguna harus menggunakan baris perintah untuk men-debug dan mendistribusikan aplikasi mereka. Menggunakan Visual Studio, salah satu IDE paling kuat di platform Windows, Flutter dan Xamarin lebih banyak pengguna-ramah daripada React Native.

3.3 Pengembangan UI

Semua kerangka kerja di atas dapat mengimplementasikan fitur aplikasi seluler yang sama pengembangan. Namun, pengguna harus menghadapi pengalaman pengembangan UI yang berbeda Karena setiap framework memiliki karakteristiknya masing-masing. Android Studio adalah lingkungan pengembangan terintegrasi yang mencakup semua siklus hidup pengembangan aplikasi. Ini memberikan pengalaman WYSIWYG dalam penyihir kode, UI merancang, debugging, dan penyebaran. Sebaliknya, React Native membagi ini proses menjadi beberapa bagian yang dikembangkan oleh alat tunggal yang berbeda. Dia perlu menggunakan antarmuka terminal atau baris perintah untuk menginstal lingkungan dan juga Memerlukan ke menerapkan editor yang berbeda untuk mengedit kode tanpa petunjuk tambahan, tulis kode UI di benak pengembang, bukan of Alat WYSIWYG. Lagipula Baik aplikasi debugging dan penerbitannya harus melalui baris perintah. Dalam sebuah kata, penggunaan keseluruhan React Native tidak ramah pengguna seperti Android Studio. Sebaliknya, Flutter lebih ramah pengguna daripada React Native dalam proyek sampel pengalaman pengembangan. Ini menggunakan Android Studio untuk mengedit kode dan debugging, dan Untuk penyebaran proyek. Dia membagikan fitur utama dari Android Studio, seperti petunjuk kode, set breakpoint, jam tangan varian, dll. Meskipun tidak mendukung WYSIWYG, itu memungkinkan pengembang untuk melihat pratinjau perubahan saat proses debug. Xamarin didukung oleh Visual Studio, yang merupakan salah satu IDE paling kuat di Platform Windows dan menyediakan banyak fitur untuk membantu pengembang dalam pengeditan kode mereka dan debugging, dan pengiriman produk. Ini juga menyediakan desain UI WYSIWYG yang sama keol sebagai Android Studio. Selain itu, Visual Studio juga memiliki fitur yang memungkinkan pengembang untuk mendistribusikan aplikasi mereka langsung ke Google Play dan Apple Store. Singkatnya, di antara Tiga lintas platform di atas, Xamarin menawarkan pengguna terbaik's Pengalaman dalam UI pengembangan untuk proyek sampel.

"

4. KESIMPULAN

Studi ini membandingkan tiga kerangka kerja lintas platform modern melalui sampel proyek praktis. Hasilnya menunjukkan bahwa menggunakan kerangka kerja yang tepat dapat secara signifikan mengurangi beban kerja meskipun semua kerangka kerja berusaha untuk mencapai kinerja dalam pengembangan asli. Artinya, teknologi lintas platform yang berbeda sesuai untuk pengembangan yang berbeda. Pertama, kerangka kerja Android Native adalah Lebih disukai jika proyek sensitif terhadap kecepatan berjalan dan penggunaan memori. Kedua React Native agak cocok untuk pengembang web karena JavaScript banyak digunakan dalam pemrograman web. Ketiga, Flutter direkomendasikan bagi mereka yang akrab dengan C ++ atau Java atau yang perlu mem-port proyek saat ini dari kerangka kerja asli ke lintas platform karena menggunakan bahasa pemrograman mirip C, Dart. Dan akhirnya, Xamarin lebih cocok untuk pengembang .NET atau mereka yang khawatir kinerja karena mudah untuk pengembangan lintas platform dan untuk mengimplementasikan aplikasi paling efisien di antara kerangka kerja lintas platform menggunakan C# di Visual Studio. Di Satu kata, mengingat keterbatasan di atas, tidak ada peluru perak sebagai salib terbaik-Kerangka kerja platform di antara mereka, yang cocok untuk semua pengembang di berbagai latar belakang pengembangan. Untuk perbaikan lebih lanjut, karya-karya masa depan berikut akan dipertimbangkan. Pertama, lebih fitur harus dilibatkan dan diuji dalam berbagai API dalam hal kinerja. Kedua, lebih banyak perangkat harus digunakan untuk mengungkapkan apakah ada perbedaan di antara mereka, atau apakah ada batasan perangkat-independen. Sementara perangkat yang berbeda harus berjalan pada versi OS yang berbeda, yang dapat memverifikasi apakah data sama dari mereka. Last but not least, data harus dikumpulkan dan dianalisis Sekali lagi setelah pembaruan baru terjadi dari kerangka kerja, karena setiap pembaruan mungkin termasuk optimasi atau degradasi. Singkatnya, semua lintas platform dapat sangat meningkatkan efisiensi pengembangan, tetapi berbeda Kerangka kerja cocok untuk aplikasi yang berbeda dan pengembang yang berbeda dan berbeda skenario, yang berarti pengambil keputusan perlu membuat pilihan berdasarkan aktual Proyek. Penelitian masa depan akan melacak efek menggunakan berbagai lintas platform untuk peningkatan lebih lanjut.

DAFTAR PUSTAKA

[1] M. Patasik and N. Nirwana, “APLIKASI JADWAL PELAJARAN BERBASIS MOBILE CROSS PLATFORM PADA SMAN 14 MAKASSAR,” 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:229276872.

[2] M. R. Wildan, “Pengembangan Aplikasi Bergerak Berbasis Augmented Reality Untuk Orientasi Mahasiswa Baru Fakultas Teknologi Industri Universitas Islam Indonesia,” *Jurnal Multidisiplin Indonesia*, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:264052569

[3] D. Vaquero-Melchor, A. M. Bernardos, and L. Bergesio, “SARA: A Microservice-Based Architecture for Cross-Platform Collaborative Augmented Reality,” *Applied Sciences*, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:216299938

[4] N. A. Shevtsiv and A. M. Striuk, “Cross platform development vs native development,” 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235409628

[5] J. Ohrt and V. Turau, “Cross-Platform Development Tools for Smartphone Applications,” *Computer (Long Beach Calif)*, vol. 45, pp. 72–79, 2012, [Online]. Available: https://api.semanticscholar.org/CorpusID:18625208

[6] K. Shah, H. Sinha, and P. Mishra, “Analysis of Cross-Platform Mobile App Development Tools,” in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, IEEE, Mar. 2019, pp. 1–7. doi: 10.1109/I2CT45611.2019.9033872.

[7] A. Muzakir, “Prototyping Aplikasi E-Health sebagai Bagian Pengenalan Obat-Obatan Dengan Teknologi Cross-Platform,” *Jurnal Informatika: Jurnal Pengembangan IT*, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:65038565

[8] Y. E. Setyo, “Implementasi Teknologi Cross Platform dalam Pembuatan Modul Pembelian Produk Asuransi Travel Menggunakan Framework Apache Cordova.: Studi Kasus PT. Asuransi Sinar Mas,” 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:217012579

[9] A. H. Mubarok, M. D. Andrian, M. A. Syarif, and N. A. Rakhmawati, “Analisis Pengaruh Penggunaan Platform Zedemy dalam Ketercapaian Pembelajaran Mahasiswa dengan Metode Cross-Sectional (Studi Kasus: Mahasiswa SI ITS),” *INFORMASI (Jurnal Informatika dan Sistem Informasi)*, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:244099342.

[10] R. Robianto, "Pengembangan Media Ajar Berbasis Cross-Platform sebagai Strategi Pembelajaran di Masa Pandemi," *Jurnal KomtekInfo*, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:240523113
[11] Triandini, E., Jayanatha, S., Indrawan, A., Werla Putra, G., & Iswara, B., "Metode Systematic Literature Review untuk Identifikasi Platform dan Metode Pengembangan Sistem Informasi di Indonesia." *Indonesian Journal of Information Systems*, 1(2), 63, 2019..
[12] B. R. Barricelli, F. Cassano, D. Fogli, and A. Piccinno, "End-user development, end-user programming and end-user software engineering: A systematic mapping study," J. Syst. Softw., vol. 149, pp. 101–137, Mar. 2019.
[13] M. Razavian, B. Paech, and A. Tang, "Empirical research for software architecture decision making: An analysis," *J. Syst. Softw.*, vol. 149, pp. 360–381, 2019.
[14] Lusiana and M. Suryani, "Metode SLR untuk Mengidentifikasi Isu-Isu dalam Software Engineering," *SATIN (Sains dan Teknologi Informasi)*, vol. 3, no. 1, 2014.