

Pocket Hospital: Digitalisasi Administrasi Pelayanan Rumah Sakit Pada Aplikasi Mobile

1st Muhammad Harits Rahman
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

haritsrahman@student.telkomuniversity.ac.id

2nd Rita Purnamasari
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

ritapurnamasari@telkomuniversity.ac.id

3rd Rustam
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

rustamtelu@telkomuniversity.ac.id

Abstrak — Digitalisasi merupakan proses transisi dari media cetak menjadi digital, hal ini dilakukan dengan fokus untuk meningkatkan efisiensi dan memberikan kenyamanan kepada pengguna. Dalam era modern, digitalisasi sudah merupakan hal yang normal tak terkecuali dalam bidang medis seperti rumah sakit. Banyak rumah sakit yang sudah melakukan digitalisasi karena pentingnya penyimpanan dan pengolahan data. Media digital dapat menyimpan data lebih lama dan dapat membuat data cadangan jikalau terjadi sesuatu kepada data utama, tidak seperti berkas fisik yang dapat hilang dan hancur. Media digital juga mempermudah pengolahan data dan meningkatkan efektifitas karena dapat mempermudah proses penyimpanan dan pencarian data jika dibutuhkan. Namun, sebagian besar rumah sakit baru menerapkan digitalisasi pada sistem staf administrasi rumah sakit dan masih sedikit rumah sakit besar yang menerapkan digitalisasi pada konsumen seperti pendaftaran online. Padahal digitalisasi dapat menawarkan banyak kenyamanan kepada pasien rumah sakit sebagai konsumen. Pocket Hospital merupakan sebuah aplikasi mobile yang diciptakan untuk meningkatkan kenyamanan pasien rumah sakit dalam berobat. Penggunaan aplikasi mobile ditujukan untuk mempermudah akses pasien rumah sakit sehingga dapat melakukan pendaftaran dan memantau nomor antrian melalui ponsel mereka.

Kata kunci— Digitalisasi, Administrasi, Aplikasi Mobile

I. PENDAHULUAN

Pada era modern ini, digitalisasi merupakan sebuah standar baru dalam berbagai bidang industri, tak terkecuali pada bidang medis. Digitalisasi membawa kemudahan dalam penyimpanan dan pengolahan data yang merupakan hal penting dalam administrasi rumah sakit, hal ini akan meningkatkan tingkat efisiensi kerja mengingat banyaknya jumlah pasien yang harus dikelola datanya. Di Indonesia, sudah banyak rumah sakit yang menerapkan digitalisasi kepada sistem administrasi rumah sakit demi meningkatkan efisiensi dan efektifitas kerja staf administrasi. Salah satu dari digitalisasi yang banyak dilakukan adalah pengimplementasian SIMRS (Sistem Informasi Manajemen Rumah Sakit), sistem ini mengintegrasikan bagian-bagian rumah sakit termasuk proses administrasi [1]. Proses digitalisasi ini tidak terbatas pada sistem internal staf rumah sakit, karena digitalisasi juga dapat memberikan kemudahan kepada pasien sebagai konsumen. Beberapa implementasi

telah dilakukan untuk meningkatkan pengalaman berobat pasien di rumah sakit. Seperti implementasi pendaftaran online yang memungkinkan pasien untuk melakukan pendaftaran terlebih dahulu, hal ini akan memudahkan pasien berobat agar tidak perlu mengantri lagi saat berkunjung ke rumah sakit [1], [2].

Kualitas pelayanan memerlukan perhatian khusus agar dapat meningkatkan pengalaman berobat pasien. Implementasi pendaftaran online merupakan langkah yang bagus untuk meningkatkan kualitas layanan. Penggunaan aplikasi *mobile* sebagai *platform* juga dapat dipertimbangkan untuk mempermudah akses. Berbeda dengan dengan penggunaan *website* yang mengharuskan pasien untuk mencari alamat URL (*Uniform Resource Locator*) dari *website* tersebut, aplikasi *mobile* tidak mengharuskan pasien untuk mencari alamat URL setiap kali ingin menggunakan layanan. Aplikasi *mobile* akan ter-*install* di dalam ponsel pasien sehingga akan mempermudah akses dan memberikan performa yang lebih stabil dibandingkan *website* yang sangat bergantung dengan kondisi server dan internet pengguna. Penggunaan aplikasi *mobile* ini diharapkan dapat menghubungkan pasien dan pihak rumah sakit dan menguntungkan kedua belah pihak [3]. Karena kualitas pelayanan perlu ditingkatkan karena berpengaruh terhadap kepuasan pasien dan memiliki pengaruh yang cukup signifikan terhadap loyalitas pasien [4].

II. KAJIAN TEORI

A. Aplikasi *Mobile*

Aplikasi *mobile* adalah program aplikasi yang dirancang untuk perangkat *mobile* yang dapat dibawa kemana-mana seperti ponsel dan tablet. Aplikasi *mobile* memiliki kepopuleran yang terus meningkat dikarenakan besarnya jumlah pengguna ponsel, pada tahun 2020 diperkirakan ada sekitar 5.65 miliar pengguna ponsel [5].

B. Flutter

Flutter merupakan *framework open-source* yang dikembangkan oleh Google untuk membuat aplikasi *multi platform*. Flutter memungkinkan *developer* untuk mengembangkan aplikasi untuk berbagai *platform* menggunakan satu basis kode, yang berarti *developer* hanya perlu menulis satu kode dan dapat *men-deploy* kode tersebut ke berbagai *platform* [6].

C. Firebase

Firebase merupakan layanan *Backend-as-a-Service* (BaaS) yang dikembangkan oleh Google. BaaS adalah jenis model layanan *cloud* yang menyediakan fungsionalitas *backend* tanpa perlu mengelola infrastruktur, sehingga *developer* dapat fokus kepada pengembangan *frontend*. Firebase menyimpan data dalam format JSON (JavaScript Object Notation) [7].

D. Google Cloud Platform

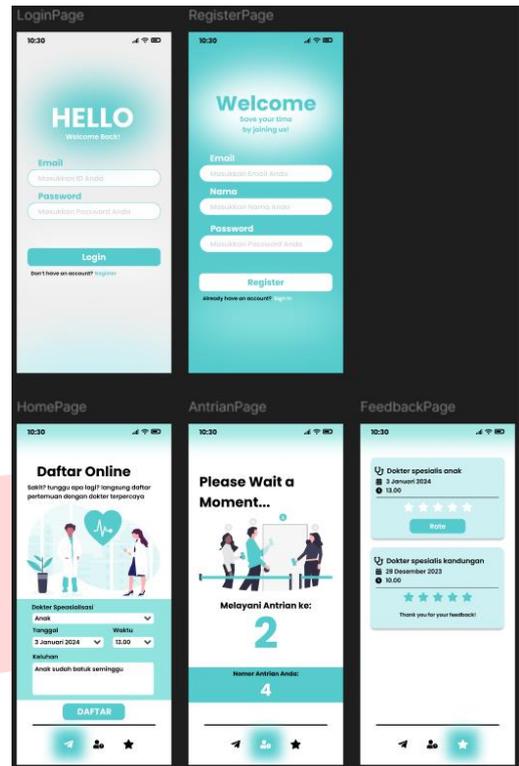
Google Cloud Platform (GCP) merupakan layanan *Platform-as-a-Service* (PaaS) yang dikembangkan oleh Google. PaaS memungkinkan *developer* untuk lebih leluasa dalam memilih layanan dan mengatur performa server sesuai dengan kebutuhan yang diperlukan [8].

III. METODE

Metode yang digunakan dalam pembuatan aplikasi Pocket Hospital memiliki beberapa langkah yang terdiri dari desain aplikasi, *frontend development*, dan *backend development*.

A. Desain Aplikasi

Langkah pertama yang harus dilakukan adalah membuat gambaran besar tentang fungsionalitas aplikasi dan membuat rincian detail dari gambaran fungsionalitas aplikasi. Hal ini penting untuk menentukan tata letak UI (*User Interface*), tema desain, dan skema warna yang akan dipakai. Gambaran besar dari fungsionalitas aplikasi Pocket Hospital adalah autentikasi pengguna, pendaftaran online, memantau antrian, dan dapat menampung ulasan. Dari daftar fungsionalitas aplikasi tersebut akan diuraikan secara detail tentang apa saja yang akan dilakukan fungsi tersebut dan apa saja yang dibutuhkan fungsi tersebut agar dapat berjalan dengan baik, desain UI/UX (*User Interface / User Experience*) harus dapat mencukupi kebutuhan tersebut.

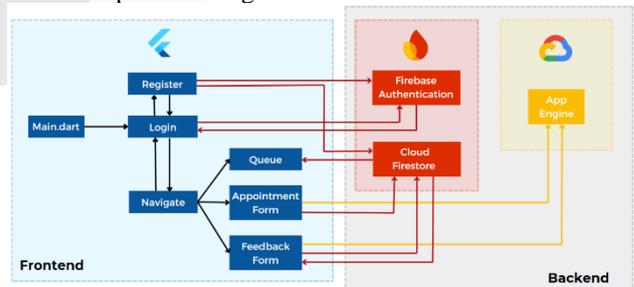


GAMBAR 1.
Desain UI/UX

Desain UI/UX pada gambar 1 merupakan desain awal aplikasi yang dibuat menggunakan *platform* Figma. Tata letak UI disesuaikan dengan kebutuhan masing-masing fitur seperti kolom *input*, tombol *submit*, dan ikon navigasi. Desain aplikasi memakai tema medis untuk menyesuaikan dengan fungsionalitas dan tujuan aplikasi, dengan skema warna hijau laut, toska, dan putih untuk melambangkan kehidupan dan kesehatan.

B. Frontend Development

Dalam pembuatan aplikasi *mobile* sangat diperlukan untuk membuat arsitektur aplikasi terlebih dahulu agar mendapat gambaran yang jelas mengenai aplikasi yang akan dibuat, arsitektur aplikasi juga akan memperjelas alur pergerakan data. Sehingga proses *frontend development* atau pengembangan *frontend* aplikasi dapat berjalan dengan lebih efisien dan terarah dengan menggunakan desain UI/UX dan arsitektur aplikasi sebagai acuan.



GAMBAR 2.
Arsitektur Aplikasi

Arsitektur *client-server* pada gambar 2 menggunakan Flutter sebagai *platform* untuk mengembangkan *frontend* dengan Firebase sebagai BaaS dan Google Cloud Platform sebagai PaaS untuk *backend*. Aplikasi akan diawali dengan

file main.dart sebagai kode utama, pengguna akan diarahkan menuju *login*, dimana sistem aplikasi akan meminta dan mengirimkan data *email* dan *password* pengguna ke Firebase Authentication untuk mengautentikasi akun pengguna. Pengguna yang belum memiliki akun akan dialihkan menuju *register* untuk membuat akun, sistem akan mengirimkan data *email* dan *password* yang sudah diberikan pengguna ke Firebase Authentication untuk mendaftarkan akun pengguna dan mengirimkan data *nama* pengguna ke dalam *database* di Cloud Firestore.

Jika pengguna berhasil *login* maka pengguna dapat memakai fungsi *navigate* pada *navigation bar* untuk berpindah halaman. Pada halaman *home* terdapat sistem *appointment form* yang akan mengirim data formulir yang sudah diisi oleh pengguna ke dalam *database* Cloud Firestore dan API (*Application Programming Interface*) App Engine untuk mengirimkan isi data formulir melalui *email* ke rumah sakit. Sehingga memungkinkan pengguna untuk melakukan pendaftaran online melalui aplikasi.

Pada halaman antrian terdapat terdapat sistem *queue*, dimana sistem akan mengambil nomor antrian yang sedang dilayani di rumah sakit dari *database* Cloud Firestore dan menampilkan nomor antrian kepada pengguna. Sistem juga akan mengambil nomor antrian pengguna dari *database* Cloud Firestore dan menampilkannya kepada pengguna, jika pengguna tidak memiliki nomor antrian maka sistem akan menampilkan simbol *dash* (-). Sistem ini membuat pengguna dapat memantau antrian dari mana saja melalui aplikasi.

Melalui sistem *feedback form* pada halaman *review*, pengguna dapat memberikan penilaian dan masukan bagi pihak rumah sakit berdasarkan pengalaman mereka. Sistem *feedback form* akan mengambil data riwayat berobat pasien dari *database* Cloud Firestore dan menampilkannya pada aplikasi. Lalu sistem akan mengirim data yang sudah dikirim oleh pengguna menuju *database* Cloud Firestore dan API App Engine untuk mengirim penilaian dan saran pengguna ke *email* rumah sakit.

C. Backend Development

Tahapan *backend development* sangat penting karena mencakup autentikasi akun, penyimpanan data, dan pengelolaan server. Arsitektur aplikasi pada gambar 2 memiliki peran yang sangat penting pada tahap *backend development*, karena menggambarkan alur pergerakan data dengan jelas. Sehingga akan mempermudah proses pengembangan karena sudah mengetahui layanan apa saja yang dibutuhkan oleh sistem aplikasi. Seperti yang sudah di sebutkan dalam penjelasan arsitektur aplikasi pada gambar 2, *backend* akan menggunakan Firebase dan Google Cloud Platform sebagai *platform backend*.

Firebase akan digunakan untuk melakukan autentikasi akun dan penyimpanan *database*. Layanan Firebase yang akan digunakan adalah Firebase Authentication dan Cloud Firestore. Layanan Firebase Authentication akan digunakan untuk melakukan autentikasi akun pengguna dalam aplikasi Pocket Hospital, autentikasi akan menggunakan *email* dan *password*. Sedangkan Cloud Firestore digunakan untuk menyimpan *database* NoSQL, *database* akan dibagi menjadi beberapa *collection* untuk menampung dan mengelompokan data-data yang dikirim dari *frontend*. *Collection* terbagi menjadi *appointments*, *feedback*, *queue*, dan *users*.

Collection appointment dan *feedback* menyimpan data-data formulir pendaftaran online yang dikirim oleh sistem *appointment form*, data disimpan ke dua *collection* yang berbeda karena *collection appointment* di pakai sebagai arsip data pesanan pasien sedangkan *collection feedback* akan digunakan oleh sistem *feedback form*, data dari *collection feedback* akan dihapus jika pengguna telah mengirim ulasan.

Collection queue berisi data nomor antrian yang memiliki tanda untuk setiap akun, *database* ini akan dipanggil oleh sistem *queue*. Sistem akan memanggil data nomor antrian terkecil dari seluruh *database* untuk menampilkan nomor antrian yang sedang dilayani dan nomor antrian terkecil dari akun yang memanggil sebagai nomor antrian pengguna.

Collection users menyimpan data *nama* dari setiap akun yang dikirim oleh sistem *register* pada saat pengguna melakukan pembuatan akun baru. Selain sistem *register*, *database* ini akan digunakan oleh sistem *navigate*. Sistem akan memanggil data *nama* yang sesuai dengan akun yang memanggil dan menampilkannya pada *appbar*.

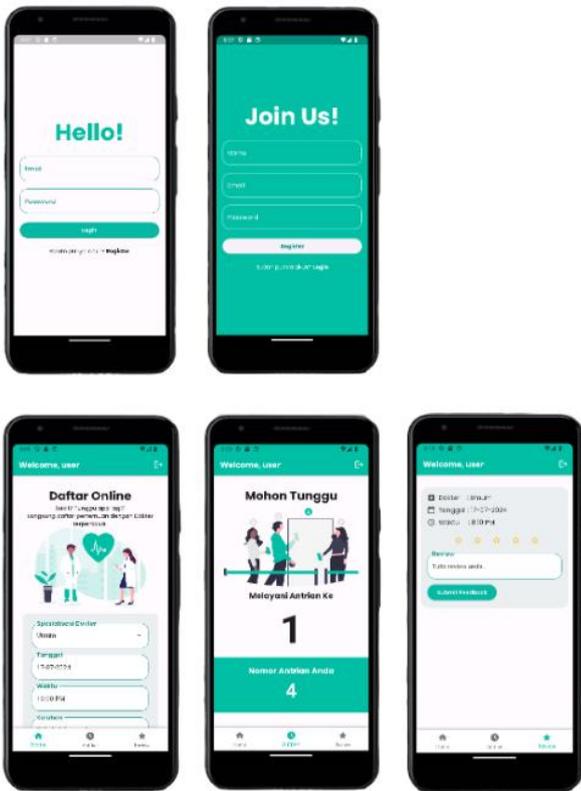
Google Cloud Platform (GCP) digunakan sebagai layanan *hosting* untuk *file index.js*, menggunakan layanan App Engine. *File* ini berisi fungsi untuk mengirim data ke *email*. Hal ini dilakukan karena fitur Cloud Functions pada Firebase merupakan layanan berbayar, sehingga App Engine dari GCP akan digunakan sebagai alternatif untuk menekan penggunaan biaya. Penggunaan layanan App Engine dari GCP dikarenakan App Engine sebagai memiliki kebebasan lebih dalam pengaturan server yang akan dipakai untuk menekan biaya.

IV. HASIL DAN PEMBAHASAN

Pengujian aplikasi *mobile* dilakukan dalam dua tahapan, yaitu *alpha testing* dan *beta testing*. *Alpha testing* dilakukan oleh tim *developer* sebelum melakukan *beta testing* untuk memastikan tidak ada *bug* dan aplikasi dapat berfungsi dengan baik dan optimal. Jika aplikasi sudah optimal dan layak pakai, maka barulah *beta testing* akan dilakukan. *Beta testing* dilakukan dengan cara membagikan aplikasi ke berbagai kelompok pengguna yang mewakili pengguna sesungguhnya (pasien) untuk mendapatkan *feedback* dari pengguna sesungguhnya dan mendapatkan data yang lebih bervariasi.

A. Alpha Testing

Alpha testing dilakukan tim *developer* dengan cara menjalankan pengujian menyeluruh terhadap semua fitur aplikasi, termasuk pendaftaran online, melihat nomor antrian secara *real-time*, dan pemberian *review* tentang pengalaman berobat. Hasil dari *alpha testing* menunjukkan bahwa tidak ditemukan *bug* dalam sistem dan semua fitur berfungsi sesuai dengan yang diharapkan. Namun, aplikasi masih terasa kasar dan kaku sehingga belum dapat memberikan pengalaman yang baik bagi pengguna. Untuk mengatasi masalah ini, penulis menambahkan animasi transisi pada aplikasi, mengganti *font* aplikasi, melakukan perubahan pada beberapa UI, dan memperbaiki skema warna agar pengguna lebih nyaman.



GAMBAR 3. Hasil Perbaikan Aplikasi

Pada gambar 3 menunjukkan hasil perubahan aplikasi setelah menjalani *alpha testing*. Hasil perbaikan aplikasi yang sudah dilakukan setelah melalui tahap *alpha testing* adalah aplikasi yang akan dibagikan dan dipakai untuk melakukan *beta testing* agar mendapat lebih banyak data yang beragam.

B. Beta Testing

Pengguna *beta testing* diberikan kuisisioner *feedback* untuk mengumpulkan data tentang pengalaman mereka menggunakan aplikasi, pengguna harus merupakan representatif dari pengguna asli yaitu pasien rumah sakit. Untuk dapat mengisi kuisisioner *feedback* aplikasi, pengguna diharuskan melihat *user manual* dan mengikuti intruksi yang telah dibagikan.

1. *Download* dan *install* Pocket Hospital
2. Buat akun di halaman *register*
3. Masuk menggunakan akun tersebut
4. Cek antrian terlebih dahulu sebelum daftar online
5. Pastikan nomor antrian anda tidak ada (-)
6. Kembali ke *home* dan lakukan daftar online (ingat data yang telah dimasukkan)
7. Cek antrian kembali, pastikan nomor antrian anda muncul
8. Pergi ke halaman *review* dan cek apakah data yang ditampilkan sudah sesuai dengan data yang anda isi saat daftar online
9. kirim *review*
10. *Logout* pada *icon* disebelah atas kanan
11. Isi kuesioner yang tercantum di bawah ini berdasarkan pengalaman Anda dalam menggunakan aplikasi ini

Berikut pada tabel 1 merupakan beberapa pertanyaan yang dilampirkan untuk mendapatkan *feedback* dari pengguna.

TABEL 1. Variabel Pertanyaan

Variabel	Pertanyaan
A	Seberapa mudah anda memahami dan menggunakan aplikasi?
B	Seberapa mudah anda menavigasi dan menemukan fitur-fitur dalam aplikasi?
C	Seberapa nyaman anda dengan skema warna aplikasi?
D	Seberapa nyaman anda melihat <i>font</i> (huruf) pada aplikasi?
E	Seberapa cepat proses autentikasi (<i>Login</i> dan <i>Register</i>)
F	Seberapa cepat proses pengiriman formulir daftar online
G	Apakah sistem antrian berfungsi dengan baik?
H	Apakah sistem <i>review</i> berfungsi dengan baik?

Menggunakan variabel pertanyaan pada tabel 1, berikut pada tabel 2 merupakan hasil kuisisioner yang didapat dari 30 responden.

TABEL 2. Hasil Pengujian Aplikasi

Variabel pertanyaan	Skala penilaian				
	Sangat Buruk	Buruk	Normal	Bagus	Sangat bagus
A	0	0	1	15	14
B	0	0	1	12	17
C	0	0	3	15	12
D	0	0	2	15	13
E	0	0	4	13	13
F	0	0	2	15	13
G	0	0	1	12	17
H	0	0	0	13	17

Hasil *beta testing* menunjukkan bahwa tidak terdapat masalah pada fitur-fitur utama, semua fitur berfungsi dengan baik dan tidak ditemukan *bug* yang menghambat penggunaan aplikasi. Fitur antrian, *review*, dan kemudahan navigasi mendapatkan nilai sangat baik. Namun, beberapa pengguna melaporkan ketidaknyamanan terkait halaman pendaftaran online. Ketidaknyamanan ini disebabkan oleh tidak adanya *loading screen* atau perubahan halaman yang jelas setelah formulir pendaftaran dikirimkan. Hal ini membuat pengguna ragu apakah formulir sudah terkirim atau terjadi kendala jaringan. Untuk mengatasi masalah ini, penulis menambahkan pesan *pop up* untuk memberitahu pengguna jika pendaftaran telah sukses.

Hasil yang didapat dari *beta testing* juga menandakan keberhasilan perbaikan aplikasi yang dilakukan pada masalah yang didapat saat *alpha testing*. Hal ini terbukti dengan tingginya penilaian yang didapat pada pertanyaan terkait UI/UX, pertanyaan tersebut terdapat pada variabel pertanyaan A, B, C, dan D.

V. KESIMPULAN

Aplikasi *mobile* Pocket Hospital berhasil melakukan digitalisasi administrasi pelayanan rumah sakit. Hal ini dibuktikan dengan hasil pengujian aplikasi yang berhasil mendapatkan nilai *sangat bagus* sebesar 50% dan *bagus*

sebesar 44.17% pada penilaian fungsionalitas. Sedangkan untuk penilaian desain UI/UX mendapatkan nilai *sangat bagus* sebesar 46.67% dan *bagus* sebesar 47.5%. Total nilai keseluruhan aplikasi adalah *sangat bagus* sebesar 48.33% dan *bagus* sebesar 45.83%. Aplikasi *mobile Pocket Hospital* berhasil mendapatkan nilai terbaik (*sangat bagus*) secara keseluruhan kecuali pada penilaian desain UI/UX yang masih mendapat nilai *baik*, namun selisih jumlah penilaian yang dimiliki dengan *sangat baik* hanya sekitar 1%. Hal ini dikarenakan kendala yang didapat beberapa pengguna pada saat awal *beta testing*, namun hal ini menjadi masukan yang sangat bagus dan penulis sudah mengatasi masalah ini dengan menambahkan pesan *pop up* pada sistem sehingga dapat meningkatkan pengalaman pengguna.

REFERENSI

- [1] C. B. Kevin, "TRANSFORMASI DIGITAL SEBAGAI BAGIAN DARI STRATEGI PEMASARAN DI RUMAH SAKIT SILOAM PALANGKA RAYA TAHUN 2020," *URNAL ADMINISTRASI RUMAH SAKIT INDONESIA*, vol. 8, no. 2, pp. 66–73, Jun. 2022.
- [2] A. H. Muhammad Arif, "SISTEM PENDAFTARAN ONLINE PASIEN RAWAT JALAN BERBASIS ANDROID KLINIK PRATAMA ARY FARMA," *Prosiding Seminar Informasi Kesehatan Nasional (SIKESNAS) 2023*, 2023.
- [3] A. Ponmalar, S. Maansi, S. Mahalakshmi, M. Shalini, and R. Madhavan, "Mobile application for hospital management system," in *Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 1434–1437. doi: 10.1109/ICICCS51141.2021.9432286.
- [4] M. Jurnal Politik dan Sosial Kemasyarakatan, W. Agustin Setyawati, M. Rifa, and C. Sasmito, "Pengaruh Kualitas Pelayanan, Fasilitas, Harga dan Citra Institusi Terhadap Kepuasan Pasien".
- [5] W. Ali, O. Riaz, S. Mumtaz, A. R. Khan, T. Saba, and S. A. Bahaj, "Mobile Application Usability Evaluation: A Study Based on Demography," *IEEE Access*, vol. 10, pp. 41512–41524, 2022, doi: 10.1109/ACCESS.2022.3166893.
- [6] Awel Eshetu Fentaw, "Cross platform mobile application development: a comparison study of React Native Vs Flutter."
- [7] C. Khawas and P. Shah, "Application of Firebase in Android App Development-A Study," *Int J Comput Appl*, vol. 179, no. 46, pp. 49–53, Jun. 2018, doi: 10.5120/ijca2018917200.
- [8] A. Gupta, "Deploying an Application using Google Cloud Platform," *2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020): conference proceedings: 5-7 March, 2020*. IEEE, 2020.