

Equilibrium: Sistem Terdistribusi untuk Aplikasi Diskusi Khusus Topik mengenai Gender Menggunakan Google Cloud Platform dan Vercel

Abdullah Azzam¹, Dima Thariq Susanto², Vina Patricia³, Yandiyan⁴, Zuhad Harkasy, Al-Hikam⁵
^{1,2,3,4,5} Teknik Informatika, Universitas Islam Negeri Sunan Gunung Djati

Keywords

Kesetaraan gender 1
Google Cloud Platform 2
Docker 3
Arsitektur terdistribusi 4
Microservices 5

ABSTRAK

Dalam era digital saat ini, kebutuhan akan platform diskusi yang aman dan terfokus semakin meningkat, terutama dalam topik sensitif seperti gender. Equilibrium hadir sebagai solusi, menawarkan ruang diskusi khusus yang berfokus pada isu-isu gender seperti kesetaraan gender, identitas gender, dan lainnya. Dengan memanfaatkan kekuatan Google Cloud Platform dan Vercel, aplikasi ini dirancang sebagai sistem terdistribusi yang handal dan skalabel. Backend aplikasi menggunakan Cloud Run dengan kontainerisasi yang dibungkus dalam Docker, sementara frontend di deploy melalui Vercel. Pendekatan ini tidak hanya memastikan performa yang optimal tetapi juga kemudahan dalam pengelolaan dan skalabilitas. Paper ini mengeksplorasi arsitektur teknis, implementasi, dan manfaat dari pendekatan ini dalam pengembangan aplikasi diskusi terfokus seperti Equilibrium.

Corresponding Author:

Insan Al-Amin
Teknik Informatika
Universitas Islam Negeri Sunan Gunung Djati
Jalan A.H Nasution No. 105, Cipadung, Cibiru, Kota Bandung, Jawa Barat 40614
Email: insanalamin@uinsgd.ac.id

1. PENDAHULUAN

Dalam beberapa dekade terakhir, isu-isu seputar gender telah menjadi topik diskusi yang semakin penting di masyarakat global khususnya di Indonesia.[1] Hal yang menjadi sorotan ialah salah satunya ialah banyaknya kasus terkait kekerasan gender khususnya yang terjadi kepada perempuan. Ini semakin diperparah dengan kondisi bahwa lebih banyak kasus yang tidak dilaporkan.[2]. Sehingga akhirnya dibutuhkan suatu forum untuk bisa saling bercerita terkait permasalahan gender sehingga para korban yang tidak melaporkan punya tempat untuk bertukar cerita atas permasalahan mereka

Quora menjadi aplikasi yang banyak digunakan karena informasi yang tersebar sangat beragam dan luas. Hal ini memungkinkan untuk pengguna bercerita tentang kejadian yang mereka alami terlepas opini yang nyata di masyarakat karena dalam Quora kondisi sosialnya merupakan maya sehingga akan menjadi leluasa untuk bercerita khususnya permasalahan yang tabu Ketika dibicarakan dalam social yang nyata seperti ketidakadilan gender seperti kekerasan dalam rumah tangga. Selain itu informasi yang banyak tersebar merupakan informasi yang dinilai baik oleh mekanisme penilaian quora sehingga menjamin mutu informasi.[3]

Oleh karena itu terinspirasi dari apa yang dilakukan oleh quora. Kami berusaha menciptakan aplikasi untuk menunjang berbagi informasi khususnya yang berkaitan dengan kesetaraan gender. Aplikasi ini akan menjadi tempat berbagi informasi dengan mutu yang baik dan tekanan social yang rendah sehingga para pengguna dapat bercerita tanpa tekanan sedikit apapun dan dapat membaca informasi dengan mutu yang baik.

Paper ini akan membahas secara mendetail arsitektur dan implementasi teknis dari Equilibrium, serta keuntungan yang diperoleh dari penggunaan teknologi-teknologi tersebut dalam menciptakan platform diskusi yang fokus dan efektif untuk isu-isu gender.

2. METODOLOGI

Metodologi pengembangan dan implementasi aplikasi Equilibrium terdiri dari beberapa tahapan utama. Setiap tahapan dirancang untuk memastikan bahwa aplikasi memenuhi kebutuhan pengguna dan berfungsi dengan optimal dalam lingkungan terdistribusi. Proses pengembangan perangkat lunak ini terdiri dari berbagai tahapan yang saling terkait. Setiap tahapan memiliki kerangka waktu spesifik di mana hasilnya harus disampaikan, dan bobot setiap tahapan tergantung pada proyek yang dikerjakan. Berikut adalah rincian metodologi yang digunakan:

A. Penelitian (Research):

Melakukan analisis kebutuhan dan memahami masalah yang perlu diselesaikan.

Mengumpulkan informasi dan data yang relevan untuk mendukung tahap perencanaan dan desain.

B. Perencanaan (Planning):

Menyusun rencana proyek yang mencakup tujuan, cakupan, jadwal, dan sumber daya yang diperlukan.

Membuat timeline proyek dengan penetapan kerangka waktu untuk setiap tahapan.

C. Desain (Design):

Membuat desain arsitektur aplikasi dan antarmuka pengguna.

Mengidentifikasi teknologi dan alat yang akan digunakan dalam pengembangan aplikasi.

D. Pengembangan (Development):

Menulis kode dan mengimplementasikan fitur-fitur yang telah dirancang.

Melakukan integrasi antara berbagai komponen sistem.

E. Pengujian (Testing):

Melakukan pengujian untuk memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang telah ditetapkan.

Mengidentifikasi dan memperbaiki bug atau masalah yang ditemukan selama pengujian.

F. Deployment:

Mengimplementasikan aplikasi ke dalam lingkungan produksi.

Memastikan bahwa aplikasi siap digunakan oleh pengguna akhir dan berfungsi dengan baik dalam kondisi operasional nyata.

Dengan mengadopsi metodologi ini, setiap tahapan dalam pengembangan aplikasi Equilibrium diharapkan dapat memberikan hasil yang optimal dan sesuai dengan kebutuhan pengguna, sekaligus menjaga kualitas dan efisiensi dalam proses pengembangan.[4]

3. HASIL DAN PEMBAHASAN

3.1. *Arsitektur Sistem*

3.2.1. **Arsitektur Backend**

Equilibrium menggunakan pendekatan microservices, di mana setiap layanan backend diisolasi ke dalam kontainer menggunakan Docker. Pendekatan ini memudahkan pengembangan, pengujian, dan deployment masing-masing layanan secara independen. Cloud Run memungkinkan layanan kontainer untuk dijalankan di cloud tanpa harus mengelola infrastruktur server. Layanan ini dapat diskalakan secara otomatis berdasarkan permintaan, mengoptimalkan penggunaan sumber daya dan memastikan performa yang konsisten.

Cloud Run memungkinkan layanan kontainer untuk dijalankan di cloud tanpa harus mengelola infrastruktur server. Layanan ini dapat diskalakan secara otomatis berdasarkan permintaan, mengoptimalkan penggunaan sumber daya dan memastikan performa yang konsisten. Docker memastikan bahwa lingkungan pengembangan dan produksi konsisten, mengurangi "works on my machine" issues. Docker images yang telah di-build dapat dijalankan di mana saja, mempermudah deployment dan migrasi.

3.2.2. **Arsitektur Frontend**

Vercel menyediakan platform deployment yang cepat dan efisien untuk aplikasi frontend. Dengan fitur seperti automatic scaling, caching, dan optimasi performa, Vercel memastikan aplikasi frontend berjalan dengan cepat dan responsif. Frontend berkomunikasi dengan backend melalui API yang telah di-definisikan dengan baik. Penggunaan API memastikan bahwa data dapat diakses dengan cepat dan aman. Arsitektur Jamstack (JavaScript, APIs, Markup) memisahkan frontend dari backend, memungkinkan pengembangan yang lebih cepat dan deployment yang lebih aman dan skalabel.

Setiap layanan backend, seperti otentikasi, manajemen pengguna, dan diskusi, di-deploy sebagai kontainer terpisah menggunakan Docker. Kontainer ini kemudian dijalankan di Cloud Run, yang mengelola penskalaan dan penyeimbangan beban secara otomatis.

3.2. *Implementasi Backend*

Setiap layanan backend, seperti otentikasi, manajemen pengguna, dan diskusi, di-deploy sebagai kontainer terpisah menggunakan Docker. Kontainer ini kemudian dijalankan di Cloud Run, yang mengelola penskalaan dan penyeimbangan beban secara otomatis. Cloud Run menawarkan keamanan tingkat tinggi dengan menjalankan kontainer dalam lingkungan yang terisolasi. Selain itu, penskalaan otomatis memungkinkan aplikasi untuk menangani lonjakan lalu lintas tanpa downtime.

3.3. *Implementasi Frontend*

Frontend dikembangkan menggunakan framework modern seperti Next.js, yang memungkinkan pembuatan aplikasi yang interaktif dan SEO-friendly. Vercel mendukung Next.js secara native, memudahkan deployment dan optimasi. Antarmuka pengguna dirancang agar intuitif dan responsif, memastikan pengalaman pengguna yang optimal di berbagai perangkat. Feedback dari pengguna diintegrasikan dalam iterasi desain untuk terus meningkatkan pengalaman pengguna.

3.4. *Integrasi dan Pengujian*

Frontend menggunakan API untuk mengakses data dari backend. Ini memastikan bahwa data selalu up-to-date dan dapat diakses dengan cepat. Manajemen state di frontend dilakukan menggunakan tools modern seperti Redux atau Context API. Pengujian unit dan integrasi dilakukan untuk setiap layanan backend. Pengujian performa dilakukan untuk memastikan sistem dapat menangani beban tinggi. Uji coba beta dengan pengguna nyata memberikan feedback yang berharga untuk perbaikan lebih lanjut.

3.5. Integrasi dan Pengujian

Pipeline CI/CD diterapkan untuk backend dan frontend, memastikan bahwa setiap perubahan kode yang diterapkan dapat di-deploy secara otomatis dan aman. Monitoring dan logging dilakukan menggunakan tools dari GCP untuk memastikan aplikasi berjalan dengan optimal. Pemeliharaan berkelanjutan meliputi penanganan bug, pembaruan fitur, dan peningkatan performa berdasarkan feedback pengguna. Skalabilitas aplikasi disesuaikan dengan beban kerja yang dinamis, memastikan bahwa aplikasi selalu responsif dan dapat diandalkan.

4. PENUTUP

Implementasi sistem terdistribusi untuk aplikasi Equilibrium menunjukkan bahwa penggunaan Google Cloud Platform dan Vercel dapat memberikan performa yang tinggi, skalabilitas yang baik, dan manajemen yang lebih sederhana. Pendekatan microservices dengan kontainerisasi memastikan bahwa setiap layanan dapat dikembangkan, di-deploy, dan diskalakan secara independen. Vercel menawarkan platform deployment yang cepat dan efisien untuk frontend, sementara Cloud Run menyediakan solusi hosting yang aman dan skalabel untuk backend.

Rekomendasi untuk pengembangan lebih lanjut meliputi peningkatan fitur interaktif di frontend, peningkatan integrasi dengan layanan pihak ketiga, dan pengembangan analitik yang lebih mendalam untuk memahami kebutuhan dan perilaku pengguna. Dengan terus melakukan iterasi dan perbaikan berdasarkan feedback pengguna, Equilibrium dapat menjadi platform diskusi yang unggul dan relevan untuk isu-isu gender.

UCAPAN TERIMA KASIH

Penelitian ini didukung oleh UIN Sunan Gunung Djati Bandung.

REFERENSI

- [1] “Keadilan_Dan_Kesetaraan_Gender_Untuk_Par”.
- [2] N. Rinukti, H. E. R. Siahaan, and A. S. Putri, “Kesetaraan dan Keadilan Gender dalam Bingkai Teologi Hospitalitas Pentakostal,” *DUNAMIS: Jurnal Teologi dan Pendidikan Kristiani*, vol. 6, no. 2, pp. 782–796, Apr. 2022, doi: 10.30648/dun.v6i2.711.
- [3] S. A. Paul, H. ^ Lichan, and E. H. Chi, “WHO IS AUTHORITATIVE? UNDERSTANDING REPUTATION MECHANISMS IN QUORA.” [Online]. Available: <http://www.quora.com/about/tos>
- [4] S. Saeed, N. Z. Jhanjhi, M. Naqvi, and M. Humayun, “Analysis of software development methodologies,” *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 445–460, 2019, doi: 10.12785/ijcds/080502.