

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian mengenai implementasi dan pengembangan REST API untuk aplikasi media sosial Stargram menggunakan Node.js, dapat disimpulkan beberapa hal. Pertama, implementasi REST API pada aplikasi web developer dapat dilakukan dengan merancang endpoint yang sesuai kebutuhan sistem, serta mengintegrasikan metode HTTP (GET, POST, PUT, DELETE) untuk mendukung komunikasi data secara efektif. Kedua, pembangunan REST API menggunakan Node.js terbukti mampu memberikan fleksibilitas dan kecepatan dalam proses pengembangan berkat dukungan ekosistem npm yang kaya serta performa asinkron dari event-driven architecture. Ketiga, teknik pengembangan yang digunakan untuk meningkatkan kinerja dan skalabilitas mencakup penggunaan konsep modularisasi kode, caching, load balancing, serta optimasi query database agar sistem tetap responsif meskipun menangani beban pengguna yang besar. Terakhir, aspek keamanan REST API dapat dijaga melalui penerapan autentikasi dan otorisasi (misalnya JWT atau OAuth), enkripsi data, validasi input, serta perlindungan dari ancaman umum seperti SQL injection dan cross-site scripting (XSS).

Dengan demikian, penelitian ini menunjukkan bahwa penerapan Node.js dalam pengembangan REST API pada aplikasi media sosial Stargram tidak hanya memungkinkan terciptanya layanan yang efisien dan skalabel, tetapi juga tetap memperhatikan aspek keamanan demi menjamin keandalan sistem.

5.2 Saran

Berdasarkan dari pengkajian hasil pembuatan Rest API menggunakan Node JS ini penulis bermaksud untuk memberikan saran yang mudah-mudahan dapat bermanfaat bagi peneliti yang selanjutnya, yaitu sebagai berikut :

1. Pengembangan Lebih Lanjut

- Eksplorasi Fitur Tambahan: Menambahkan fitur tambahan pada aplikasi media social stargram untuk meningkatkan pengalaman pengguna. Misalnya, integrasi sistem rekomendasi, sistem notifikasi real-time, atau implementasi metode pembayaran yang lebih beragam.
- Penyempurnaan Antarmuka Pengguna (UI/UX): Melakukan peninjauan dan peningkatan pada antarmuka pengguna untuk memastikan tampilan yang lebih menarik, navigasi yang intuitif, dan pengalaman pengguna yang lebih baik.

2. Pengoptimalan Kinerja

- Pengoptimalan Server: Mengevaluasi dan mengoptimalkan kinerja server menggunakan teknik-teknik terbaru seperti load balancing,

caching, dan tuning parameter server untuk meningkatkan responsivitas aplikasi.

- **Pemantauan Kinerja Aplikasi:** Mengimplementasikan sistem pemantauan kinerja aplikasi untuk melacak dan menganalisis performa aplikasi secara real-time. Ini dapat membantu mengidentifikasi dan menanggulangi potensi masalah kinerja.

3. Keamanan Aplikasi

- **Peningkatan Lapisan Keamanan:** Meningkatkan lapisan keamanan aplikasi dengan mempertimbangkan praktik keamanan yang lebih lanjut, seperti enkripsi data tambahan, validasi input yang ketat, dan pemantauan keamanan secara proaktif.
- **Audit Keamanan Periodik:** Melakukan audit keamanan secara berkala untuk menilai potensi risiko keamanan baru dan memastikan bahwa aplikasi tetap aman dari ancaman keamanan yang berkembang.

4. Pengembangan Lanjutan pada Teknologi Tertentu

- **Penelitian Node.js Terbaru:** Tetap mengikuti perkembangan terbaru dalam ekosistem Node.js, memantau pembaruan versi, dan mengeksplorasi pustaka-pustaka baru yang dapat meningkatkan fungsionalitas aplikasi.
- **Eksplorasi Teknologi Terkait:** Mengeksplorasi teknologi terkait yang dapat meningkatkan performa, keamanan, dan fleksibilitas aplikasi, seperti pustaka-pustaka baru dalam ekosistem Node.js atau teknologi terkait basis data.

5. Penerapan Metodologi Pengembangan

- **Penggunaan Metodologi Pengembangan Terkini:** Mempertimbangkan penggunaan metodologi pengembangan terkini seperti metode agile atau DevOps untuk meningkatkan efisiensi dan responsivitas tim pengembangan.

- Implementasi CI/CD: Mengevaluasi potensi implementasi Continuous Integration (CI) dan Continuous Deployment (CD) untuk mempercepat siklus pengembangan dan pengiriman perubahan.

