**Dokumentasi REST API**

**Dibuat oleh**

**Sena Fadjar**

**Muhammad Raffi**

**Affredo alin Kambu**

**1. Pendahuluan** REST API (Representational State Transfer Application Programming Interface) adalah arsitektur yang digunakan untuk membangun layanan web. API ini memungkinkan komunikasi antara klien dan server dengan menggunakan protokol HTTP. Dalam proyek ini, kami telah mengembangkan REST API menggunakan Node.js dan Express, yang menyediakan fungsionalitas untuk mengelola pengguna, produk, pesanan, dan item pesanan. Dengan menggunakan pendekatan RESTful, API ini dirancang untuk menjadi efisien, mudah digunakan, dan dapat diintegrasikan dengan sistem lain. Dalam era digital saat ini, kebutuhan akan layanan web yang responsif dan dapat diandalkan semakin meningkat. REST API menawarkan solusi yang fleksibel dan skalabel untuk memenuhi kebutuhan tersebut.

**2. Desain dan Arsitektur** Desain API ini mengikuti prinsip RESTful, yang berarti bahwa setiap endpoint dirancang untuk melakukan operasi tertentu pada sumber daya. Kami menggunakan metode HTTP standar seperti GET, POST, PUT, dan DELETE untuk berinteraksi dengan data. Struktur database dirancang dengan empat tabel utama: Users, Products, Orders, dan Order\_Items, yang saling berhubungan untuk memastikan integritas data. Setiap tabel memiliki atribut yang relevan dan relasi yang jelas, memungkinkan pengambilan dan manipulasi data yang efisien. Dengan menggunakan Node.js, kami dapat memanfaatkan event-driven architecture yang memungkinkan penanganan permintaan secara asinkron, meningkatkan performa dan responsivitas API.

**3. Tabel dan Relasi** Tabel Users menyimpan informasi pengguna, termasuk ID, username, password, dan email. Tabel Products menyimpan informasi tentang produk yang tersedia, termasuk ID, nama, deskripsi, dan harga. Tabel Orders mencatat pesanan yang dibuat oleh pengguna, sementara tabel Order\_Items menyimpan detail tentang produk yang termasuk dalam setiap pesanan. Relasi antar tabel ini memastikan bahwa data dapat diakses dan dikelola dengan efisien. Misalnya, setiap pesanan terkait dengan pengguna tertentu, dan setiap item pesanan terkait dengan produk tertentu. Dengan struktur ini, kami dapat dengan mudah melakukan query untuk mendapatkan informasi yang dibutuhkan.

**4. Endpoint API** API ini menyediakan berbagai endpoint untuk mengelola data. Endpoint untuk pengguna memungkinkan penambahan, pengambilan, pembaruan, dan penghapusan pengguna. Endpoint untuk produk dan pesanan memiliki fungsi serupa, memungkinkan pengguna untuk mengelola inventaris dan pesanan mereka. Setiap endpoint dirancang untuk memberikan respons yang jelas dan informatif, termasuk status operasi dan data yang relevan. Misalnya, saat pengguna berhasil menambahkan produk baru, API akan mengembalikan informasi produk yang baru ditambahkan beserta status sukses. Dengan cara ini, pengguna dapat dengan mudah memahami hasil dari setiap operasi yang dilakukan.

**5. Pengujian dan Validasi** Pengujian dilakukan menggunakan alat seperti Postman untuk memastikan bahwa semua endpoint berfungsi dengan baik. Setiap endpoint diuji untuk memastikan bahwa mereka mengembalikan respons yang diharapkan dan menangani kesalahan dengan benar. Validasi juga diterapkan untuk memastikan bahwa data yang diterima sesuai dengan format yang diharapkan, mengurangi kemungkinan kesalahan saat memproses permintaan. Kami juga melakukan pengujian beban untuk memastikan bahwa API dapat menangani sejumlah besar permintaan secara bersamaan tanpa penurunan performa. Dengan pengujian yang menyeluruh, kami dapat memastikan bahwa API ini siap digunakan dalam lingkungan produksi.

**6. Keamanan** Keamanan adalah aspek penting dari setiap API. Dalam proyek ini, kami menggunakan JSON Web Tokens (JWT) untuk mengautentikasi pengguna dan melindungi endpoint yang sensitif. Dengan menggunakan JWT, kami dapat memastikan bahwa hanya pengguna yang terautentikasi yang dapat mengakses data tertentu, meningkatkan keamanan keseluruhan API. Selain itu, kami juga menerapkan praktik terbaik dalam pengelolaan kata sandi, seperti hashing dan salting, untuk melindungi informasi pengguna. Dengan langkah-langkah keamanan ini, kami berusaha untuk melindungi data pengguna dan mencegah akses yang tidak sah.

**7. Kesimpulan** REST API yang telah dibangun menyediakan fungsionalitas dasar untuk mengelola pengguna, produk, pesanan, dan item pesanan. Dengan desain yang mengikuti prinsip RESTful, API ini dapat dengan mudah diperluas dan diintegrasikan dengan sistem lain. Proyek ini menunjukkan bagaimana teknologi modern dapat digunakan untuk membangun layanan web yang efisien dan aman. Kami percaya bahwa API ini akan memberikan nilai tambah bagi pengguna dan membantu mereka dalam mengelola data dengan lebih baik.

**8. Rencana Pengembangan Selanjutnya** Ke depan, kami berencana untuk menambahkan fitur tambahan, seperti integrasi dengan API publik, pengelolaan inventaris yang lebih baik, dan antarmuka pengguna yang lebih ramah. Dengan terus mengembangkan API ini, kami berharap dapat memberikan solusi yang lebih komprehensif untuk kebutuhan manajemen data. Kami juga akan terus memantau umpan balik pengguna untuk meningkatkan fungsionalitas dan pengalaman pengguna secara keseluruhan.