

Klin8-Shoes

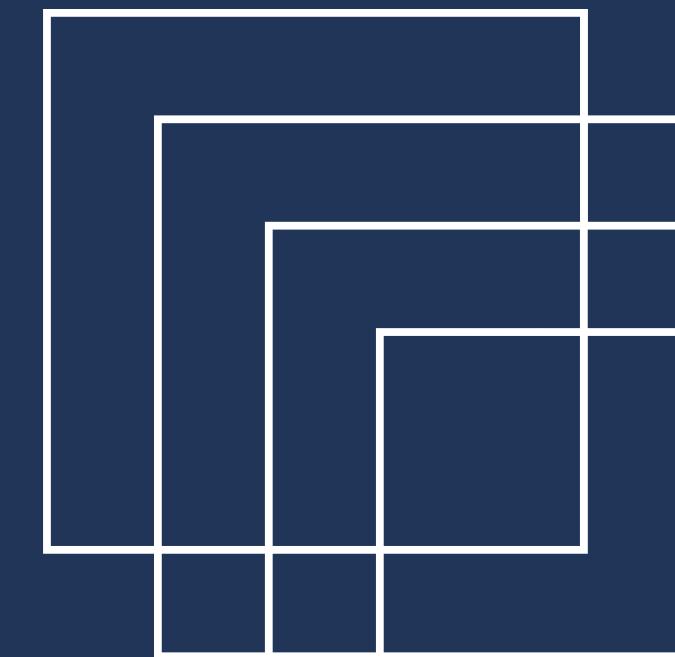
Tugas Bootcamp Harisenin
membuat implementasi
NodeJS dan Database untuk
tugas Backend Final Project.



Presented By : Kelompok 8

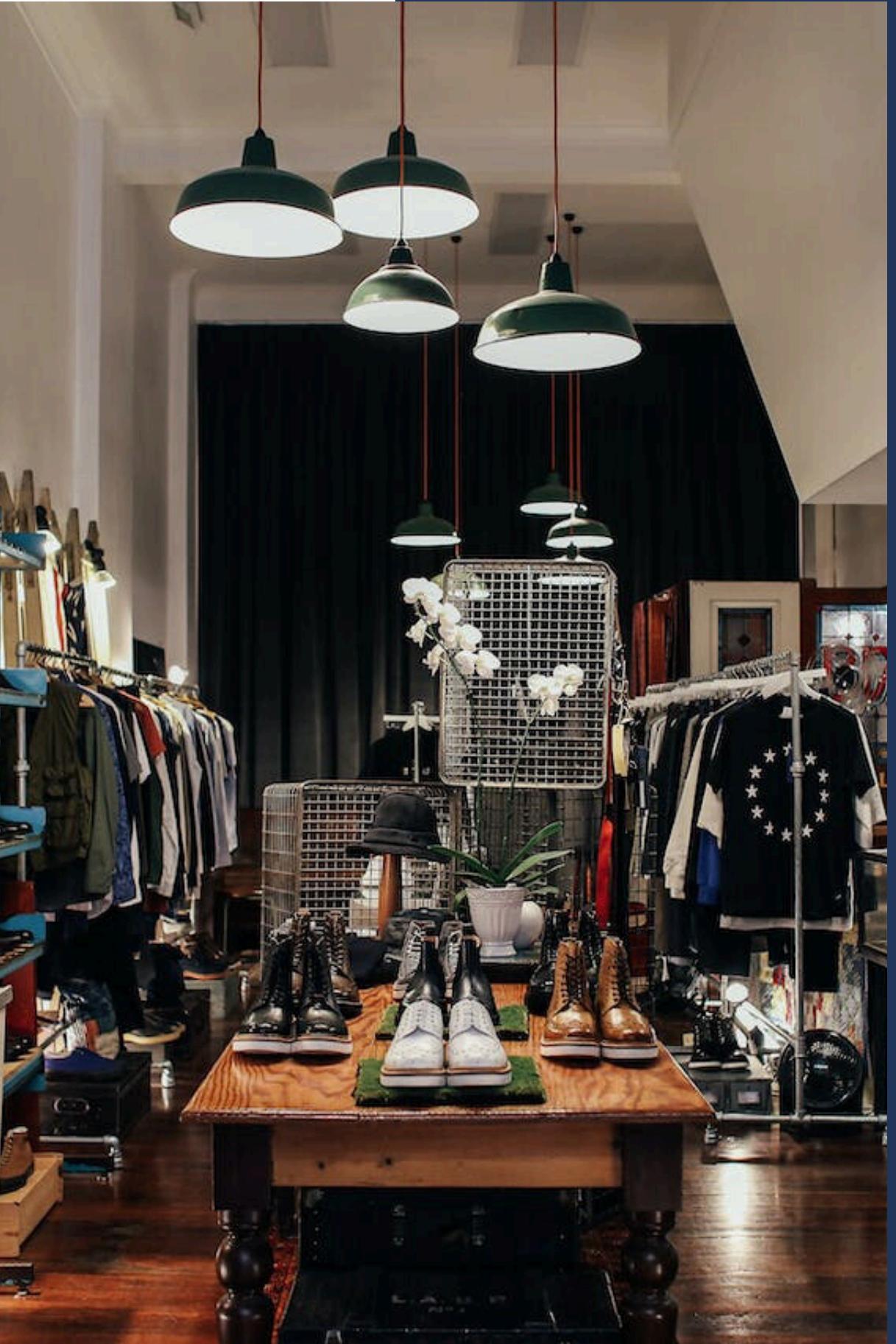
CONTENT

- 01** KELOMPOK 8
- 02** KONTRIBUSI
- 03** ENTITY RELATIONAL MODEL
- 04** PENJELASAN ERD
- 05** ALASAN MENGGUNAKAN FRAMEWORK EXPRESS JS
- 06** ALASAN MENGGUNAKAN PRISMA ORM
- 07** ALASAN MENGGUNAKAN NODEMAILER
- 08** PENJELASAN IMPLEMENTASI AUTENTIKASI JWT
- 09** STATUS CODE RESPONSE HTPP YANG KAMI GUNAKAN
- 10** THANKS FOR WATCHING



Kelompok 8

➤ Ahmad Abuya	06
➤ Ahmad Hanif M	07
➤ Denny Setya H	17
➤ Dwi Prasetyo	18
➤ Hanif Indra K	23
➤ Rico Lim RJ	49
➤ Ridwan Andrea S	50
➤ Surya AS	53





KONTRIBUSI

Ahmad Abuya

- Membuat Auth Router
- Membuat User Router
- Membuat Order Router

Denny Setya H

- Membuat Entity Relational Diagram
- Membuat Code PlantUml
- Membuat Schema Prisma

Dwi Prasetyo

- Membuat Service, Controller, Router, Validation, dan Error Handling
- Membuat Unit Test
- Membuat OpenAPI



DESKRIPSI APLIKASI

Di Klin8, kami berencana untuk mengembangkan aplikasi website untuk layanan laundry sepatu, di mana pelanggan dapat memesan layanan baik secara online maupun offline.

Kami menawarkan beberapa mode layanan, antara lain:

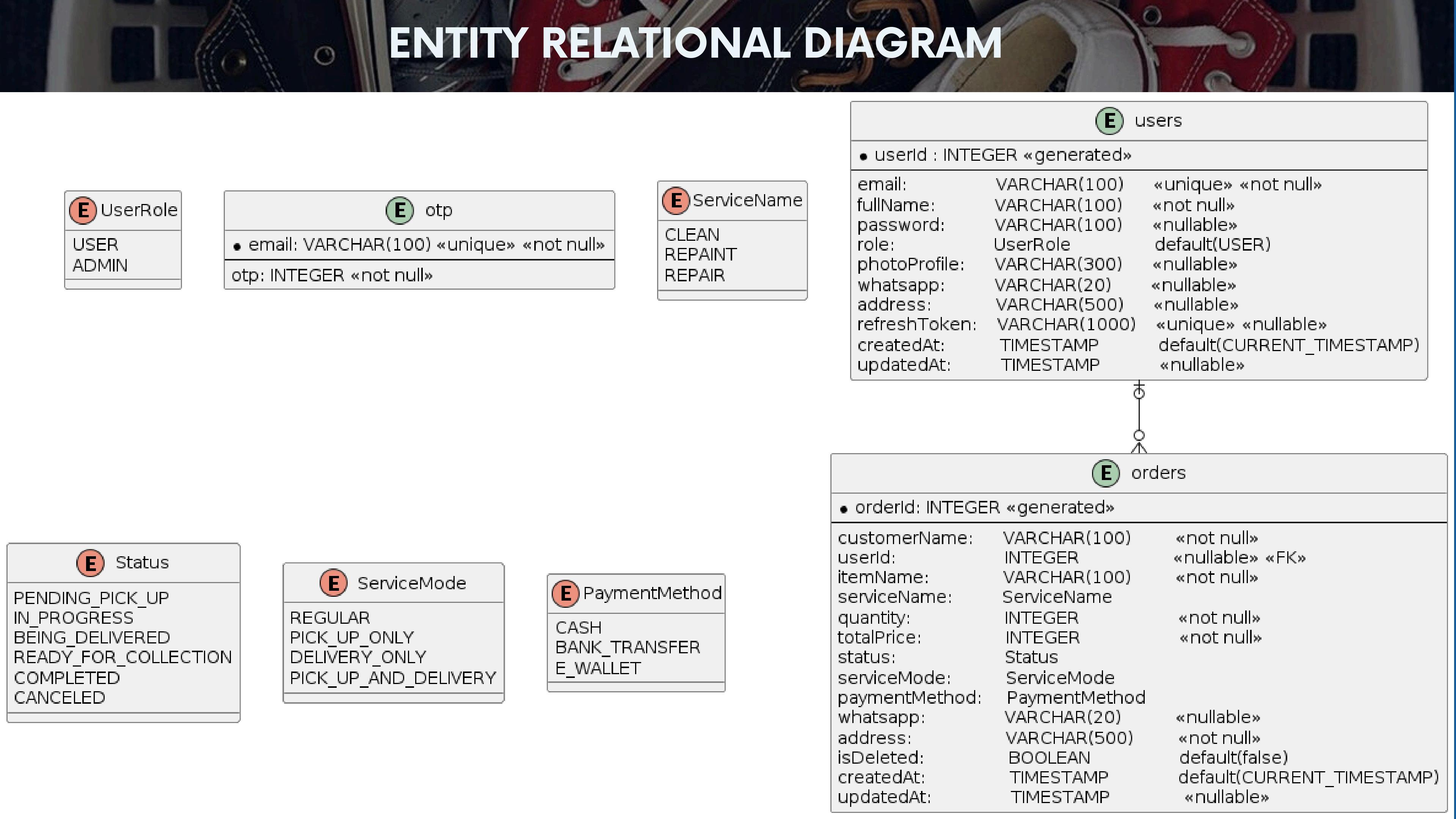
- Regular: Pelanggan mengirim dan mengambil sepatu sendiri di tempat laundry.
- Pick Up Only: Sepatu pelanggan diambil oleh pegawai laundry, tetapi pelanggan yang mengambilnya kembali.
- Delivery Only: Pelanggan mengirim sepatu ke tempat laundry, dan setelah selesai, sepatu akan diantarkan oleh pegawai laundry.
- Pick Up and Delivery: Sepatu pelanggan diambil dan diantarkan oleh pegawai laundry.

Di Klin8, kami berencana untuk mempermudah pelanggan dalam melacak status laundry mereka yang memesan layanan secara offline.

Mereka akan diberikan QR code oleh pegawai laundry saat memesan layanan di tempat laundry.

Kemudian, pelanggan dapat langsung memindai QR code tersebut di halaman utama situs web Klin8 meskipun mereka belum login. Dengan cara ini, pelanggan dapat dengan mudah melihat status laundry mereka.

ENTITY RELATIONAL DIAGRAM



PENJELASAN ERD

Di sini, kami membuat 3 entitas: users, orders, dan otp. Untuk relasinya, entitas users berelasi dengan entitas orders dalam relasi one-to-many, yang berarti users dapat memiliki banyak order.

Simbol |o--o{ digunakan untuk menandakan bahwa user bisa memiliki banyak order, namun tidak wajib memiliki order sama sekali. Dengan kata lain, entitas users tidak memiliki ketergantungan pada entitas orders.

Dan sama seperti entitas users, entitas orders memiliki tidak memiliki ketergantungan pada entitas users, karena di aplikasi ini memiliki 2 cara untuk melakukan pesanan yaitu dengan cara lewat aplikasi (online) dan langsung ke tempat laundry (offline) di mana setiap pelanggan tidak harus memiliki akun di aplikasi kami.

Kami membuat 5 enum untuk memperjelas nilai konstan dalam kode kami, yaitu UserRole, ServiceName, Status, ServiceMode, dan PaymentMethod. Ini bertujuan untuk meningkatkan keterbacaan kode.

ALASAN MENGGUNAKAN FRAMEWORK EXPRESS JS



Alasan

- Kami memilih menggunakan framework ExpressJS karena sintaksnya yang sederhana, membuatnya mudah dipahami dan digunakan.
- Selain itu, ExpressJS juga fleksibel dalam struktur folder, yang memungkinkan kami untuk lebih leluasa dalam menentukan struktur folder nya tanpa harus mengikuti pola modular seperti yang ada di NestJS.
- Karena project kami relatif sederhana, menggunakan ExpressJS lebih cocok karena ukurannya yang lebih ringan.

ALASAN MENGGUNAKAN PRISMA ORM



Prisma

Alasan

- Kami memilih menggunakan Prisma karena sintaksnya yang sederhana dalam pembuatan schema modelnya. Prisma memungkinkan kami untuk berinteraksi dengan database tanpa harus menulis queri SQL secara manual, sehingga mempermudah dalam pengembangan aplikasi.
- Dari segi keamanan, Prisma juga lebih aman karena secara otomatis menggunakan parameterized queries untuk semua operasi database nya, sehingga lebih aman dari SQL injection.
- Meskipun kami tahu bahwa secara kinerja, penggunaan queri manual mungkin lebih cepat, kami tetap memilih menggunakan ORM karena dalam tahap awal production, perbedaan kinerja tersebut belum akan terasa secara signifikan. Kami akan menunggu munculnya masalah kinerja dalam aplikasi kami sebelum memutuskan untuk beralih menggunakan queri manual. Selain itu, Prisma juga menyediakan opsi untuk menggunakan Queri Raw, sehingga kami tidak perlu khawatir jika suatu saat aplikasi kami menjadi besar dan mengalami penurunan performa.

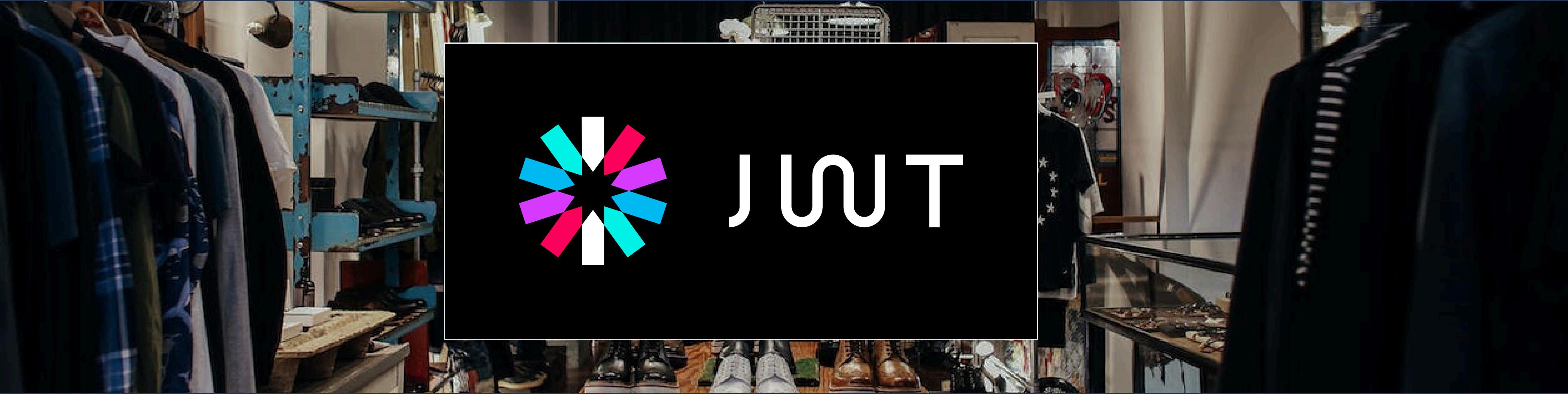
ALASAN MENGGUNAKAN NODEMAILER



Alasan

- Kami menggunakan Nodemailer untuk mengirim OTP karena Nodemailer merupakan library sederhana dan mudah digunakan. Dengan Nodemailer, kami dapat dengan mudah mengintegrasikan fungsi pengiriman otp ke email tanpa harus menulis kode yang rumit.
- Selain itu alasan kami menggunakan Nodemailer untuk mengirim otp, karena Nodemailer adalah library open source yang gratis dan dapat digunakan tanpa biaya tambahan atau biaya lisensi.
- Dan di sini kami membuat implementasi send otp melalui Gmail dengan menggunakan Nodemailer, untuk konfigurasi Nodemailer nya kami menggunakan Oauth2. Kami menggunakan Oauth2 supaya lebih aman karena tidak perlu mengirim Credential seperti password email. ini dapat mengurangi resiko kebocoran Credential. Selain itu, menggunakan OAuth2 memungkinkan kami untuk menggunakan refresh token untuk mendapatkan access token baru tanpa perlu login ulang, sehingga aplikasi kami dapat terus mengirim email setelah access token nya kadaluarsa.

PENJELASAN IMPLEMENTASI AUTENTIKASI JWT



Penjelasan

- Kami menggunakan implementasi Refresh Token dengan menggunakan jsonwebtoken untuk meningkatkan keamanan autentikasi di aplikasi kami. Setelah user berhasil login, kami membuat dua token: Access Token dan Refresh Token. Untuk Payload nya kami bedakan. Access Token berisi data User seperti userId, email, dan role, dengan masa berlaku 30 menit. Sementara, Refresh Token hanya berisi UUID dan role User untuk memastikan keunikan Refresh Token. Masa berlaku Refresh Token adalah 30 hari.
- Kedua token disimpan di cookie http-only agar lebih aman, dan supaya tidak dapat diakses melalui JavaScript. Karena kedua token disimpan di cookie, Jika ada keperluan akses lintas domain, kita bisa menggunakan library cors untuk mengizinkan penggunaan cookie di lintas domain. Refresh Token juga disimpan di database.
- Dengan pemisahan ini, jika Access Token bocor, pencuri hanya memiliki akses selama 30 menit. Namun, jika Refresh Token dicuri, pencuri tidak dapat mengakses aplikasi karena tidak ada data User di dalamnya. Saat User logout, Refresh Token di set menjadi null di database sehingga tidak bisa digunakan lagi. Dengan demikian, ini menghindari menyimpan data User dalam Access Token dengan masa berlaku yang panjang. Refresh Token memungkinkan pembuatan Access Token baru setelah masa berlaku Access Token habis dalam 30 menit, sehingga User tidak perlu login lagi dalam waktu dekat.

STATUS CODE RESPONSE HTTP YANG KAMI GUNAKAN

200

Menunjukkan bahwa request HTTP yang dikirimkan oleh client berhasil dipahami, diterima, dan diproses dengan baik oleh server.

201

Digunakan sebagai respons atas permintaan POST yang berhasil, yang menghasilkan pembuatan data baru.

400

Menunjukkan bahwa server tidak dapat memahami atau memproses request client karena permintaan tersebut terlihat bermasalah atau tidak valid.

401

Menunjukkan bahwa client tidak memiliki otorisasi yang diperlukan untuk mengakses sumber daya yang diminta.

403

Menunjukkan bahwa client tidak memiliki izin yang diperlukan untuk mengakses sumber daya tersebut.

404

Menunjukkan bahwa sumber daya yang diminta oleh client tidak dapat ditemukan pada server.

409

Menunjukkan bahwa server tidak dapat memenuhi permintaan client karena terdapat konflik dengan status saat ini dari sumber daya yang diminta.

422

Digunakan untuk menyatakan bahwa server gagal memproses permintaan karena kesalahan internal yang berkaitan dengan lingkungan atau konfigurasi server itu sendiri, seperti masalah dengan pengaturan environment atau ketergantungan yang tidak terpenuhi.

500

Menunjukkan bahwa terjadi kesalahan internal pada server saat mencoba memproses permintaan client.

Kelompok 8

Thank's For
Watching

