

# Struktur Aplikasi Express.js

# Pemisahan Konsep (Separation of Concerns)

- Pemisahan Konsep (SoC) adalah prinsip desain perangkat lunak yang mendorong pemisahan berbagai tanggung jawab dan perhatian dalam kode.
- Hal ini membantu meningkatkan kerapihan dan pemeliharaan perangkat lunak.

# Apa Itu Pemisahan Konsep?

- Pemisahan Konsep adalah pendekatan yang memecah kode menjadi bagian-bagian terpisah yang menangani tugas-tugas tertentu.
- Setiap bagian (konsep) bertanggung jawab hanya atas satu hal.
- Prinsip ini membantu menghindari percampuran logika yang berbeda.

# Manfaat Pemisahan Konsep

- Meningkatkan Kerapihan: Kode lebih mudah dipahami dan dikelola.
- Kemudahan Pemeliharaan: Memperbaiki atau menambah fitur lebih mudah.
- Pengujian yang Lebih Baik: Bagian kode dapat diuji secara terpisah.
- Kolaborasi yang Lebih Baik: Memungkinkan tim untuk bekerja secara efisien.

# Pemisahan Model

- Direktori `app/models` berisi definisi model data.
- Contoh: `author.js` adalah model untuk entitas `Author`.
- Ini memisahkan bagian aplikasi yang bertanggung jawab atas interaksi dengan basis data.

# Pemisahan Kontrol

- Direktori `app/routes` dan `app/router.js` berisi definisi rute dan logika kontrol.
- `authors.js` berisi rute terkait Author.
- Ini memisahkan bagian aplikasi yang bertanggung jawab atas logika bisnis dan kontrol aliran.

# Konfigurasi Terpusat

- Direktori `config` berisi berkas `database.js`, yang mungkin berisi konfigurasi basis data.
- Ini memisahkan pengaturan dan konfigurasi dalam satu lokasi terpusat.

# Pemisahan Database

- Direktori `database` berisi `migrations` dan `seeders`.
- `Migrations` digunakan untuk mengelola versi basis data.
- `Seeders` digunakan untuk mengisi data awal.



# Pencapaian Pemisahan Konsep dengan Menggunakan Modul Node.js

- Pemisahan Konsep (SoC) dalam aplikasi Node.js dapat dicapai dengan menggunakan modul Node.js.
- Modul memungkinkan pemisahan peran dan tanggung jawab dalam kode.

# Mengapa Menggunakan Modul?

- Modul adalah bagian terpisah dari kode yang dapat digunakan kembali.
- Membantu dalam memisahkan tanggung jawab dan konsep dalam aplikasi.
- Mempermudah pemeliharaan dan pengujian.

# Membuat Modul

- Membuat modul sangat mudah dalam Node.js.
- Anda dapat membuat modul dengan membuat berkas JavaScript terpisah.
- Contoh: Membuat modul "database.js" untuk mengelola koneksi basis data.