Tes Magang Backend Developer: Node.js, Drizzle ORM, PostgreSQL

Tujuan Tes:

Menguji kemampuan calon magang dalam mengembangkan aplikasi backend sederhana menggunakan Node.js, Drizzle ORM untuk berinteraksi dengan PostgreSQL, serta dokumentasi API dengan Swagger. Aplikasi ini harus dapat melakukan operasi CRUD (Create, Read, Update, Delete) pada dua tabel: products dan orders.

Ketentuan Umum:

1. Bahasa Pemrograman: Node.js
2. Database: PostgreSQL
3. ORM: Drizzle ORM
4. Framework: Express.js (TypeScript nilai Plus)
5. Dokumentasi API: Swagger (menggunakan swagger-jsdoc dan swagger-ui-express)
6. Fitur yang Diharapkan:
- API untuk mengelola data produk (products) dan pesanan (orders).
- Setiap operasi CRUD harus diimplementasikan sebagai endpoint terpisah di API.
- API harus dapat berinteraksi dengan database PostgreSQL menggunakan Drizzle ORM.
- API harus disertai dengan dokumentasi menggunakan Swagger.
7. Output yang Diharapkan: Kandidat harus membuat aplikasi yang bisa diakses melalui URL lokal, misalnya http://localhost:3000, dan dokumentasi API di http://localhost:3000/api-docs.

Deskripsi Tugas:

1. Buat Database PostgreSQL:
- Buat database dengan nama ecommerce_db.
- Buat tabel products dan orders sesuai dengan struktur berikut:

```
CREATE TABLE products (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    description TEXT,
    price DECIMAL(10, 2)
);

CREATE TABLE orders (
    id SERIAL PRIMARY KEY,
```

```
    product_id INT REFERENCES products(id) ON DELETE CASCADE,
    quantity INT,
    total_price DECIMAL(10, 2),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

2. Implementasi API CRUD:
- Create: Implementasikan endpoint POST /products untuk menambahkan produk baru ke database. Data yang diterima dalam body request adalah name, description, dan price.
- Read: Implementasikan endpoint GET /products untuk mengambil daftar semua produk. Implementasikan juga endpoint GET /orders untuk mengambil daftar semua pesanan.
- Update: Implementasikan endpoint PUT /products/:id untuk memperbarui data produk berdasarkan ID.
- Delete: Implementasikan endpoint DELETE /products/:id untuk menghapus produk berdasarkan ID.
- Buat juga endpoint POST /orders untuk membuat pesanan baru dengan produk tertentu dan jumlah yang diminta.

3. Dokumentasi API dengan Swagger:
- Gunakan swagger-jsdoc dan swagger-ui-express untuk mendokumentasikan API yang telah dibuat.
- Dokumentasi Swagger harus bisa diakses di http://localhost:3000/api-docs.
- Dokumentasi API harus mencakup:
  - Deskripsi setiap endpoint beserta metode HTTP-nya (GET, POST, PUT, DELETE).
  - Deskripsi parameter input (body, path) dan format response.
  - Kode status HTTP yang sesuai dengan setiap respons.

4. Pengujian:
- Pastikan semua operasi CRUD dapat berjalan dengan baik.
- Lakukan pengujian untuk setiap endpoint (dapat menggunakan Postman atau CURL).
- Pastikan saat menghapus produk yang sudah ada, pesanan yang berhubungan juga terhapus.

Penilaian:

1. Fungsionalitas (60%):
- Apakah API berjalan dengan baik dan sesuai dengan spesifikasi?
- Apakah setiap operasi CRUD berfungsi seperti yang diharapkan?
- Apakah API dapat menangani request dan response dengan benar?
- Apakah data disimpan dan diperbarui dengan benar dalam database PostgreSQL menggunakan Drizzle ORM?

2. Dokumentasi (20%):
- Apakah dokumentasi API lengkap dan mudah dipahami?
- Apakah semua endpoint terdaftar dengan jelas di Swagger?
- Apakah API request dan response dijelaskan dengan baik dalam dokumentasi?

3. Struktur Kode dan Keamanan (20%):
- Apakah kode mudah dibaca dan terstruktur dengan baik?
- Apakah ada penggunaan prinsip-prinsip keamanan yang tepat? (misalnya, password sebaiknya disimpan dalam bentuk terenkripsi jika relevan)

Contoh Endpoint yang Diharapkan:

1. POST /products

Body:

```
{
 "name": "Smartphone",
 "description": "Latest model smartphone",
 "price": 599.99
}
```

Response (201 Created):

```
{
 "id": 1,
 "name": "Smartphone",
 "description": "Latest model smartphone",
 "price": 599.99
}
```

2. GET /products

Response (200 OK):

```
[
 {
  "id": 1,
  "name": "Smartphone",
  "description": "Latest model smartphone",
  "price": 599.99
 }
]
```

3. POST /orders

Body:

```
{
 "product_id": 1,
```

```
 "quantity": 2
}

Response (201 Created):
{
 "id": 1,
 "product_id": 1,
 "quantity": 2,
 "total_price": 1199.98,
 "created_at": "2025-04-15T12:00:00Z"
}
```

4. PUT /products/:id

```
Body:
{
 "name": "Smartphone Pro",
 "description": "Updated model smartphone",
 "price": 699.99
}

Response (200 OK):
{
 "id": 1,
 "name": "Smartphone Pro",
 "description": "Updated model smartphone",
 "price": 699.99
}
```

5. DELETE /products/:id

Response (204 No Content): Tidak ada konten yang dikembalikan, menandakan bahwa produk telah berhasil dihapus.

Pengiriman:

1. Source Code:
- Kandidat harus mengunggah source code aplikasi (termasuk file .env, package.json, dan kode backend) ke GitHub atau platform serupa.
- Sertakan langkah-langkah yang jelas untuk menginstal dan menjalankan aplikasi di file README.md.

2. Dokumentasi API:
- Sertakan screenshot atau dokumentasi terkait API Swagger di file README.md atau link langsung ke dokumentasi (misalnya http://localhost:3000/api-docs).

3. Deadline:
- Tes ini harus diserahkan dalam waktu 3 hari setelah penerimaan tugas.