**Nama : Adittya Rizqi Fauzi**

**NIM : 23104410079**

**Jawaban dan Analisis**

**Metode yang Paling Sesuai: PATCH**

Metode HTTP yang paling sesuai dan efisien untuk menambahkan satu produk lagi ke dalam daftar *items* pada pesanan yang sudah ada adalah **PATCH**.

**Alasannya:**

1. **Sifat Operasi:** Penambahan satu produk adalah operasi **pembaruan parsial** (sebagian). Kita hanya perlu memodifikasi *array* items tanpa menyentuh bidang data lainnya (seperti customer, shippingAddress, atau orderId).
2. **Efisiensi Bandwidth:** Metode PATCH hanya mengirimkan bagian data yang akan diubah. Dalam kasus ini, *payload* (isi permintaan) hanya perlu berisi instruksi atau data untuk menambahkan item baru, jauh lebih kecil daripada mengirimkan seluruh objek pesanan.

**Perbedaan Mendasar Metode HTTP dalam Konteks Pembaruan Data**

| **Metode HTTP** | **Tujuan Mendasar** | **Konteks Pembaruan Pesanan (Order)** | **Paling Sesuai untuk Tugas?** |
| --- | --- | --- | --- |
| **POST** | **Membuat** sumber daya baru. | Digunakan untuk **Membuat** pesanan baru (/orders). Jika digunakan untuk pembaruan, POST pada *sub-resource* (misal: /orders/ORD-X/items) berarti **menambahkan** item baru sebagai sumber daya terkait. | **Tidak Tepat** untuk pembaruan seluruh pesanan, tetapi **Tepat** untuk membuat *sub-resource* baru (*append* item) jika API dirancang secara hierarkis (/orders/{id}/items). |
| **PUT** | **Mengganti** (replace) sumber daya secara keseluruhan. | Digunakan untuk **Mengganti seluruh** objek pesanan. Anda harus mengirimkan SEMUA bidang data (termasuk orderId, customer, **semua** items yang lama dan baru, dan shippingAddress). Data lama yang tidak dikirim akan dihapus. | **Tidak Tepat** dan **Tidak Efisien.** Risiko data hilang jika ada bidang yang terlewat dalam *payload*. |
| **PATCH** | **Memodifikasi** (modify) sumber daya secara parsial (sebagian). | Digunakan untuk melakukan **pembaruan sebagian** (parsial). *Payload* hanya berisi instruksi perubahan (misalnya, menambahkan satu item baru ke *array* items). | **Paling Tepat** dan **Paling Efisien.** Hanya mengirimkan perubahan yang diperlukan, meminimalkan *traffic* dan risiko. |

**Analisis Mendalam: POST vs. PATCH untuk Penambahan Item**

Dalam skenario *e-commerce*, ada dua cara yang valid untuk menangani penambahan item ke pesanan yang sudah ada, tergantung pada desain arsitektur API:

**1. Menggunakan PATCH (Pembaruan Parsial) - Disarankan**

* **Endpoint:** PATCH /orders/ORD-2025-10-123
* **Payload (Contoh):**

JSON

{

"operation": "add",

"path": "/items",

"value": { "productId": "P008", "quantity": 1 }

}

*(Bergantung pada standar implementasi PATCH, seperti JSON Patch)*

* **Mekanisme:** *Server* akan menerima instruksi ini, menemukan pesanan dengan ID yang relevan, dan hanya menjalankan operasi *append* pada *array* items dalam basis data. Ini adalah cara RESTful yang paling bersih untuk pembaruan parsial.

**2. Menggunakan POST (Membuat *Sub-Resource*)**

* **Endpoint:** POST /orders/ORD-2025-10-123/items
* **Payload (Contoh):**

JSON

{

"productId": "P008",

"quantity": 1

}

* **Mekanisme:** Dalam arsitektur REST yang ketat, items dapat dianggap sebagai *sub-resource* dari orders. Menggunakan POST pada /items berarti Anda sedang **membuat** sebuah objek *item* baru **di bawah** pesanan tersebut. Secara fungsional, ini juga berhasil menambahkan produk, tetapi secara semantik, POST biasanya mengembalikan sumber daya baru yang telah dibuat (dalam hal ini, detail item yang baru ditambahkan).

**Kesimpulan Akhir**

Walaupun kedua PATCH dan POST dapat mencapai hasil fungsional yang diinginkan, **PATCH** ke *endpoint* pesanan utama (/orders/{id}) adalah pilihan yang secara **semantik paling tepat** untuk memodifikasi satu properti (yaitu, *array* items) dari sumber daya yang sudah ada. Ini sesuai dengan prinsip desain API modern yang memisahkan operasi *replace* (PUT) dari operasi *modify* (PATCH).