

2. Berikan tanggapan untuk setiap perulangan dan kapan perulangan tersebut baik untuk digunakan?

1. Loop "Standar": `for` vs `for-in`

Perulangan `for` tradisional (yang pakai `int i = 0; ... i++`), seperti di soal (a), memberi kita kontrol manual. Menurut saya, ini paling pas digunakan hanya jika kita benar-benar butuh angka indeks (`i`)-nya.

Sebaliknya, `for-in` (seperti di soal b). Kita tidak perlu memikirkan `i` atau `i++`. Kita langsung dapat datanya. Ini merupakan pilihan utama untuk mengunjungi semua data di `List`.

2. Loop "Manual": `while` dan `do-while`

Loop `while` (soal c) dan `do-while` (soal d) terasa mirip dengan `for` tradisional karena kita harus mengurus counter (`i`) sendiri. ini agak rawan error kalau kita lupa menulis `i++`, programnya bisa terjadi infinite loop .

sedangkan `do-while` berguna jika kita ingin kode berjalan minimal satu kali, tapi untuk data `List` yang jumlahnya sudah pasti.

3. Metode Fungsional: `.forEach`, `.where`, dan `.map`

- **`.forEach`** (soal e) ini mirip `for-in`, tapi lebih ringkas. Ini dapat dipakai kalau kita cuma mau melakukan sesuatu (kayak `print laporan`) untuk setiap data.
- **`.where`** (soal g & h) ini sangat membantu sebagai alat saring. Menggunakan ini jauh lebih ringkas daripada membuat loop `for` tradisional yang di dalamnya ada `if` untuk menyaring data.
- **`.map`** (soal f & g) sebagai alat ubah. Ini sangat praktis saat kita ingin mengubah satu `List` menjadi `List` baru dengan bentuk data yang berbeda, misalnya dari `List karyawan lengkap` menjadi `List` yang isinya angka proyek saja.