

VhiWEB Software Engineer Test Case

SCENARIO:

The business team needs to import a CSV file containing over 100,000 product records. During testing, the import process raised high server load, timeouts, and occasional failures.

Jawaban:

1. Identify the Problem:

- Import dilakukan secara langsung (synchronous), menyebabkan request timeout.
- Proses parsing dan validasi dilakukan seluruhnya di memori (memory heavy).
- Tidak ada batching/pagination (server harus proses sekaligus).
- Index pada database tidak optimal atau tidak ada.
- Duplikasi atau format data tidak divalidasi dengan baik.

2. Propose a Solution:

a. Gunakan Queue dan Worker untuk memproses import secara asynchronous:

- CSV di-upload ke server dan disimpan sementara.
- Worker akan membaca isi file dalam batch (misal 1000 record per job).
- Gunakan queue seperti Bull (Redis), RabbitMQ, atau Laravel Queue (jika pakai Laravel).

b. Lakukan Validasi Sebelum Disimpan:

- Validasi format data (misal price angka, name wajib diisi).
- Skip/Log data yang gagal validasi.

c. Gunakan Upsert jika memungkinkan (untuk menghindari duplikasi).

d. Tambahkan Index di kolom pencarian/kunci unik untuk efisiensi.

e. Gunakan streaming CSV parser (seperti fast-csv, papaparse, atau csv-parser di Node.js) agar tidak membaca seluruh file ke memori sekaligus.

3. Evaluate Efficiency:

- Gunakan time logger (console.time/benchmarking) untuk ukur waktu setiap batch.
- Monitor memory usage & CPU load saat proses import berlangsung.
- Bandingkan waktu import batch vs langsung.

4. Optimize:

- Bandingkan performa beberapa strategi (synchronous, asynchronous, streaming).
- Gunakan bulk insert untuk mengurangi jumlah transaksi DB.
- Gunakan database indexing & partitioning jika dataset besar.

SCENARIO:

Deadline sistem e-Procurement sudah dekat. Beberapa anggota tim mengalami kendala tidak terduga. Tim khawatir tidak bisa menyelesaikan project tepat waktu.

Jawaban:

1. Workload Management:

- Lakukan mapping ulang terhadap task penting dan tentukan prioritas utama.
- Breakdown task besar menjadi bagian lebih kecil dan terukur.
- Lakukan daily stand-up untuk mengevaluasi progress dan kendala.
- Bila perlu, realokasikan resource dari bagian yang tidak mendesak ke task kritis.

2. Team Support:

- Ajak rekan yang mengalami kendala berdiskusi terbuka untuk mencari solusi.
- Lakukan pair programming untuk membantu mereka yang stuck.
- Buat dokumentasi/notes singkat jika ada bagian yang sulit dipahami.
- Jika perlu, eskalasi ke lead agar ada bantuan tambahan atau penyesuaian timeline.

3. Communication with Stakeholders:

- Segera informasikan ke stakeholder jika ada risiko keterlambatan.
- Sampaikan progress objektif, alasan kendala, dan rencana mitigasinya.
- Berikan timeline baru dan pastikan ada komitmen tim untuk menyesuaikan.