# **Interview Test**

Salam hangat!

Dokumen ini memuat soal interview test untuk Tech Lead, terdapat 2 soal yang masing-masing bisa anda kerjakan dan lengkapi di file ini.

Anda diharapkan submit soal dan jawaban dalam bentuk **Google Docs** yang bisa diakses oleh kami.

Laporkan dan lampirkan link **Google Docs** apabila anda sudah menyelesaikan soal yang diberikan dengan cara reply email dari email yang anda terima saat mendapatkan soal ini.

Selamat mengerjakan!

## **Soal 1**

Deskripsi:

Buat API sederhana dengan fitur backend dan menggunakan teknologi berikut:

* Backend: Golang atau Laravel

Spesifikasi Proyek:

* To-do list untuk pencatatan tugas pribadi
  + Backend:
    - Endpoint untuk CRUD Tugas (Create, Read, Update, Delete) Tugas
    - Gunakan database sebagai penyimpanan data
      * Bisa menggunakan SQL atau No-SQL
    - *Automation Test* terkait API yang dibuat
* Requirements:
  + User bisa mencatat tugas dia hari ini dan kapan ekspektasi tugas tersebut selesai
  + User bisa update status tugas menjadi:
    - Waiting List
    - In Progress
    - Done
  + User bisa hapus task (*soft delete*)
  + User bisa menyimpan deskripsi tugas mereka
  + User bisa melihat daftar tugas mereka berdasarkan waktu pengerjaan dan status

Cara Pengumpulan:

* Sertakan link repository GitHub dengan access *public*
  + Sertakan dokumentasi pengaturan, penggunaan dan instruksi pengujian melalui file **README** di repository GitHub anda
  + (https://github.com/ar1stio/aristio-sagala-test)

## **Soal 2**

Deskripsi:

Buatlah sebuah *system design* yang menjelaskan bagaimana Anda akan mendesain arsitektur sistem untuk aplikasi POS yang *scalable* dan aman.

Product Requirements:

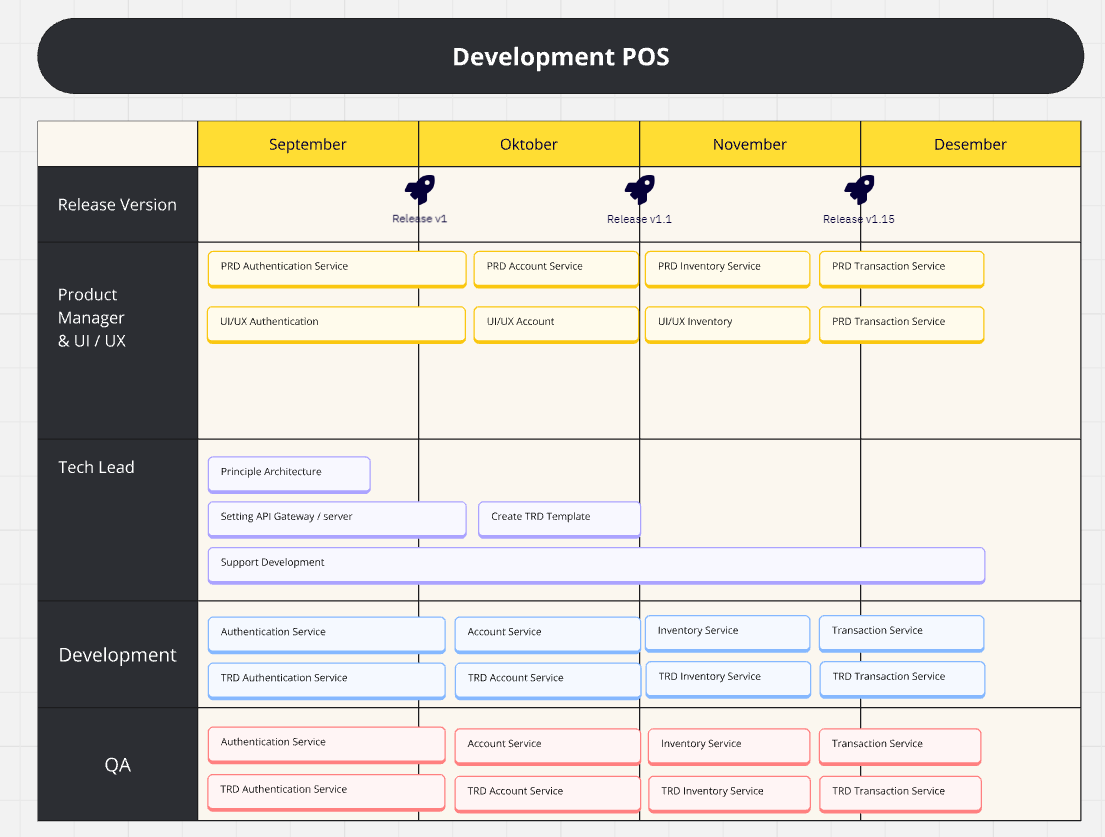
* User dibagi menjadi 2
  + Admin Backoffice Internal
    - Bisa melihat semua transaksi melalui website
    - User bisa mengatur daftar menu
  + Kasir
    - User bisa membuat pesanan
    - User bisa mengubah pesanan
    - User bisa membuat transaksi QRIS sehingga customer bisa melakukan pembayaran
* Kriteria Penilaian
  + Pemahaman tentang arsitektur sistem
  + Desain yang *scalable* dan aman
  + Penjelasan tentang keputusan teknis yang diambil
  + Pemahaman tentang teknologi yang digunakan (bahasa pemrograman, database, caching, load balancing, etc)

Cara Pengumpulan:

* Sertakan jawaban anda pada lembar google docs kosong di bagian bawah setelah soal nomor 2
* Jika anda ingin menyertakan gambar atau link yang cukup besar, anda bisa sertakan dalam bentuk link yang dilampirkan di google docs pada soal ini

(Silahkan tulis jawaban soal 2 di bagian bawah ini)

* Roadmap Development



Keterangan :

1. Development estimeasi waktu sekitar 4 bulan dengan beberapa versi release untuk memanagement secara agile development dibagi menjadi bagian kecil dengan pengerjaan menggunakan beberapa sprint
2. Actor yang berperan ada beberapa yaitu Product Manager, Tech Lead, Developer FE and BE , UI/UX dan QA
3. Untuk mempermudah dalam aturan development perencanaan menggunakan PRD (Procuct Requirement Decoment) sebagai acuan product manager dalam merencanakan sistem yang dibutuhkan oleh User
4. Untuk mempermudah dalam aturan development untuk developer mentranslate menjadi TRD (Technical Requirement Document) biasanya untuk mendokumentasikan seperti API contract, tools yang digunakan, dan daily task yang di copy dari tools dailiy task.
5. Tech Lead bertanggung jawab terhadap architecture yang digunakan, tools dan syarat dan ketentuan dalam development dan membantu support development jika diperlukan.
6. QA memastikan dan mendokumentasikan beberapa case positif maupun negatif yang sekiranya bisa terjadi didalam TRD.

* Teknologi yang dipakai

1. Bahasa Pemgoraman.
2. Backend

Golang: Digunakan untuk membangun API karena performanya yang cepat, efisiensi memori, dan kemampuannya dalam menangani banyak koneksi secara bersamaan. Golang juga mendukung concurrency, yang sangat penting untuk aplikasi yang memerlukan respons cepat.

1. Frontend

React js dengan TypeScript, aplikasi ini memberikan antarmuka interaktif dan responsif untuk mengelola menu dan melihat transaksi.

1. Database

menggunakan database relational yang menjamin integritas data seperti MySQL atau PostgreSQL

1. Caching

Menggunakan redis sebagai caching untuk mempercepat penggunaan data yang sering berada dalam transaksi api yang digunkan.

1. Keamanan
2. Enkripsi Data: Semua data sensitif akan dienkripsi, baik dalam transit (menggunakan TLS) maupun saat disimpan.
3. Autentikasi dan Otorisasi: Menggunakan OAuth2 untuk autentikasi yang aman dan penerapan kontrol akses berbasis peran (RBAC) untuk memastikan hanya pengguna dengan hak akses yang sesuai yang dapat mengakses fitur tertentu.
4. Audit Logging: Semua tindakan pengguna, terutama di antarmuka admin, akan dicatat untuk tujuan audit.
5. Bila perlu menggunakan OTP dalam case tertentu misal update password dan penggunaan autheticate lainnya
6. Teknis

Management Loging : bisa menggunakan tools seperti grafana atau sentry

Menggunakan tools tambahan seperti : messageBroker, elastic search jika diperlukan nantinya untuk menambah performance hit API

Docker: sebagai container yang digunakan dalam handle applikasi yang dibuat. Untuk service maupun tools support yang digunakan.

CI/CD: Menggunakan Jenkins atau GitLab CI/CD untuk pipeline integrasi dan deployment berkelanjutan, memastikan perubahan kode diuji dan di-deploy secara otomatis.