

Tugas Besar

IF4031 - Arsitektur Aplikasi Terdistribusi

Dipersiapkan oleh
Asisten IF4031
(Iya, asistennya hanya ada 1)



Waktu Mulai

Senin, 8 Desember 2025, 22:00 WIB

Waktu Akhir

Jum'at, 2 Januari 2026, 23.59 WIB

Changelog

1. Aturan terkait kontribusi anggota kelompok

I. Latar Belakang

II. Teknis Pengerjaan

Berikut merupakan teknis pengerjaan tugas besar:

1. Tugas besar dikerjakan secara berkelompok, dengan masing-masing kelompok terdiri atas 3 anggota. Kelompok yang terdiri atas 4 anggota dapat dibentuk hanya apabila jumlah anggota kelas tidak habis dibagi 3, dan seluruh kelompok lain di sheets sudah memiliki 3 anggota.
 - a. Isi kelompok pada [sheets pemilihan kelompok](#). Pastikan kelompok sudah tertulis sebelum 11 Desember 2025, pukul 23.59.
 - b. Peserta kuliah yang tidak tercantum pada *sheets* pada akhir masa pengisian kelompok akan dikelompokkan satu sama lain, atau disebarkan ke kelompok-kelompok lain yang sudah terdaftar.
 - c. Setiap anggota kelompok wajib berkontribusi ke tugas besar.
2. Tuliskan seluruh pertanyaan terkait tugas besar pada [sheets QnA](#). Apabila pertanyaan belum dijawab setelah 24 jam, silahkan hubungi asisten melalui akun-akun berikut:
 - a. (LINE/Discord) pkmntrainerfrost. atau
 - b. (Teams) 23525044@mahasiswa.itb.ac.id
3. Diperbolehkan untuk menggunakan LLM untuk membantu menuliskan dokumen rancangan, dengan syarat penggunaannya dideklarasikan di laporan. Diperbolehkan juga untuk menggunakan LLM untuk melakukan implementasi, selama kode yang dihasilkan dipahami, dan penggunaan LLM diakui dan dituliskan pada kode.
 - a. **Catatan penting:** Jika diketahui terdapat bagian laporan atau kode yang sama dengan hasil pekerjaan lain maupun *repository* di internet, maka akan dianggap melakukan kecurangan. Alasan menggunakan bantuan LLM akan diabaikan.
4. Segala kecurangan baik sengaja dan tidak disengaja akan ditindaklanjuti, yang dapat berakibat sanksi akademik ke setiap pihak yang terlibat

III. Spesifikasi Tugas

Tujuan pada tugas besar ini adalah merancang sebuah arsitektur aplikasi terdistribusi, serta mengimplementasikan *proof-of-concept* sederhana. Implementasi *proof-of-concept* dilakukan terhadap satu fungsionalitas atau kualitas saja yang dianggap paling penting.

3.0. Background: Aplikasi Pelaporan Warga

Aplikasi yang akan kalian rancang dan implementasi adalah aplikasi yang memungkinkan warga suatu kota (dengan jumlah penduduk sekitar 2,5 juta) untuk **melaporkan permasalahan-permasalahan di lingkungannya** kepada pihak yang berwenang. Laporan-laporan ini dapat terkait dengan berbagai jenis masalah, seperti kriminalitas, kebersihan, kesehatan, perawatan fasilitas, dan seterusnya. Masalah-masalah yang dilaporkan kemudian akan diteruskan (hanya) ke pihak yang terkait; contohnya, masalah terkait kerusakan fasilitas umum hanya akan diteruskan kepada lembaga sarana prasarana, **dan tidak diteruskan kepada pihak lainnya**.

Mengingat bahwa aplikasi seperti ini perlu menangani berbagai jenis masalah - serta daerah dan jumlah pengguna yang cukup besar - terdapat suatu “keharusan” untuk menggunakan arsitektur bersifat terdistribusi (misalnya *microservice*) dibandingkan arsitektur bersifat monolitik. Terdapat beberapa tantangan yang perlu dijawab dalam proses perancangan. Tantangan-tantangan ini dapat berkaitan dengan fungsionalitas secara langsung, atau terkait dengan kualitas aplikasi secara keseluruhan. Pada tugas ini, tantangan-tantangan yang perlu kalian selesaikan sudah diberikan pada bagian selanjutnya dalam bentuk pernyataan-pernyataan klien.

3.1. Kebutuhan Aplikasi

Berikut adalah kebutuhan-kebutuhan fungsional yang perlu dipenuhi oleh aplikasi yang akan kalian rancang:

1. Aplikasi dapat digunakan oleh warga untuk melaporkan masalah-masalah di lingkungan sekitarnya dalam bentuk laporan tertulis; laporan mungkin dilengkapi oleh lokasi dan multimedia; laporan boleh bersifat publik (dapat dilihat oleh seluruh warga), privat (hanya dapat dilihat pelapor dan penerima), atau anonim (seperti privat, tetapi penerima tidak dapat mengetahui identitas pelapor; digunakan untuk *whistleblowing*).
2. Aplikasi dapat digunakan oleh warga untuk mengelola laporan-laporan yang telah ia buat serta melihat status penyelesaiannya.
3. Aplikasi dapat digunakan oleh warga untuk melihat laporan-laporan lain yang bersifat publik serta memberikan *upvote* ke laporan-laporan yang ingin didukung.
4. Aplikasi dapat memberikan notifikasi kepada pelapor ketika terdapat kemajuan pada penyelesaian laporan-laporannya.

5. Aplikasi dapat digunakan oleh pihak berwenang untuk memantau dan membantu merespon masalah-masalah yang dilaporkan oleh warga, dan terkait dengan kewenangannya, secara *real-time*.
6. Aplikasi dapat digunakan oleh pihak berwenang untuk melakukan analisis data terkait masalah-masalah yang dilaporkan.
7. Aplikasi dapat digunakan oleh pihak berwenang untuk meneruskan laporan-laporan yang terkait dengan kewenangannya ke aplikasi-aplikasi atau sistem-sistem eksternal yang sudah ada.
8. Aplikasi dapat mengeskalasi laporan-laporan ke pihak dengan kewenangan lebih tinggi apabila tidak berhasil ditangani dalam waktu tertentu.
9. Aplikasi dapat digunakan oleh kewenangan tertinggi untuk memantau kinerja bawahan-bawahannya dalam menanggapi masalah-masalah yang dilaporkan oleh warga.

Sementara itu, berikut adalah kebutuhan-kebutuhan non-fungsional atau **kualitas-kualitas** yang wajib dipenuhi oleh sistem:

1. Keamanan (*security*)

- a. Aplikasi harus mampu memisahkan masalah-masalah yang dilaporkan oleh warga berdasarkan jenisnya. Pihak penerima laporan hanya dapat melihat masalah yang berada di bawah wewenangnya; sebagai contoh, departemen kebersihan hanya dapat melihat masalah kebersihan dan tidak dapat melihat masalah departemen lainnya.
- b. Aplikasi harus mampu menyediakan kemampuan untuk memberikan laporan secara anonim jika diinginkan pelapor, sehingga pelapor tidak dapat dilacak identitasnya.
- c. Aplikasi harus mampu memvalidasi identitas pelapor dan memastikan identitas tersebut tidak dipalsukan.
- d. Aplikasi harus mampu menghilangkan (menyembunyikan) seluruh data pribadi pembuat sebuah laporan anonim.
- e. Aplikasi harus mampu menjaga keamanan data pengguna serta data laporan baik saat dikirimkan (*in-transit*) maupun saat disimpan (*at-rest*).

2. Keandalan (*reliability*)

- a. Aplikasi harus mampu memastikan bahwa kegagalan suatu (*instance*) fungsionalitas atau komponen tidak menyebabkan kegagalan (total) keseluruhan aplikasi.

3. Skalabilitas (*scalability*)

- a. Aplikasi harus mampu menangani lonjakan beban pengguna yang bersifat mendadak dan tidak terduga.
- b. Aplikasi harus mampu mengurangi jumlah sumber daya yang digunakan ketika beban tidak terlalu besar.

4. Kinerja (*performance*)

- a. Aplikasi harus mampu memberikan *response time* yang memuaskan.

5. Observabilitas (*observability*)

- a. Aplikasi harus mampu memungkinkan tim infrastruktur untuk memantau dan menganalisis kinerja sistem secara keseluruhan serta melacak pergerakan lalu lintas antar komponen
- b. Aplikasi harus mampu memungkinkan tim keamanan untuk memantau lalu lintas yang diterima sistem dengan tetap mempertahankan privasi pengguna

Good Luck, Have Fun ;)

IV. Pengumpulan dan Deliverables

Berikut adalah *deliverables-deliverables* yang perlu kalian kumpulkan:

1. Dokumen perancangan sistem, yang setidaknya berisi:
 - a. Halaman sampul (*cover*) yang setidaknya berisi nama kelompok, nama dan NIM setiap anggota kelompok, nama aplikasi, serta gambar bebas.
 - b. Tabel kontribusi setiap anggota kelompok.
 - c. Deskripsi umum sistem yang dirancang.
 - d. Diagram arsitektur sistem serta penjelasannya
 - e. Daftar teknologi terpilih untuk setiap komponen sistem, beserta dengan **alasan pemilihan masing-masing teknologi**.
 - f. Pemilihan fungsionalitas dan kualitas yang akan diimplementasi pada *proof-of-concept*, beserta dengan **alasan pemilihannya**. Silahkan pilih fungsionalitas (atau fungsionalitas-fungsionalitas) beserta kualitas (atau kualitas-kualitas) yang **dianggap paling kritis** serta **mampu menggambarkan sistem secara keseluruhan**.
 - g. Penjelasan hasil implementasi *proof-of-concept* yang dilengkapi dengan tautan kepada sumber kode beserta gambar-gambar yang dokumentasi.
 - h. Seluruh asumsi (bila ada) yang diambil dalam membuat rancangan.
2. (Tautan) sumber kode implementasi *proof-of-concept*.
 - a. Cukup lakukan implementasi terhadap fungsionalitas dan kualitas yang telah dipilih sebelumnya.
 - b. Pastikan terdapat **instruksi untuk menjalankan aplikasi**.
3. (Tautan) video demonstrasi aplikasi yang diunggah ke YouTube atau Google Drive.
 - a. Seluruh anggota kelompok wajib hadir pada video demonstrasi, yang menggantikan demonstrasi secara sinkron.
 - b. Video demonstrasi harus menunjukkan fungsionalitas-fungsionalitas dan kualitas-kualitas yang diimplementasi, beserta dengan penjelasannya.
 - c. Video demonstrasi maksimal berdurasi 10 menit.

Seluruh *deliverables* dikumpulkan melalui forms pengumpulan sebelum *deadline*.

V. Penilaian dan Bonus

Penilaian akan dilakukan menggunakan **submisi terakhir pada *forms***, dengan rubrik sebagai berikut:

1. (60) Dokumen perancangan sistem

- a. (5) Penulisan dan kelengkapan dokumen
- b. (5) Deskripsi umum
- c. (15) Diagram arsitektur
- d. (15) Pemilihan teknologi
- e. (15) Pemilihan fungsionalitas dan kualitas
- f. (5) Penjelasan dan dokumentasi hasil implementasi

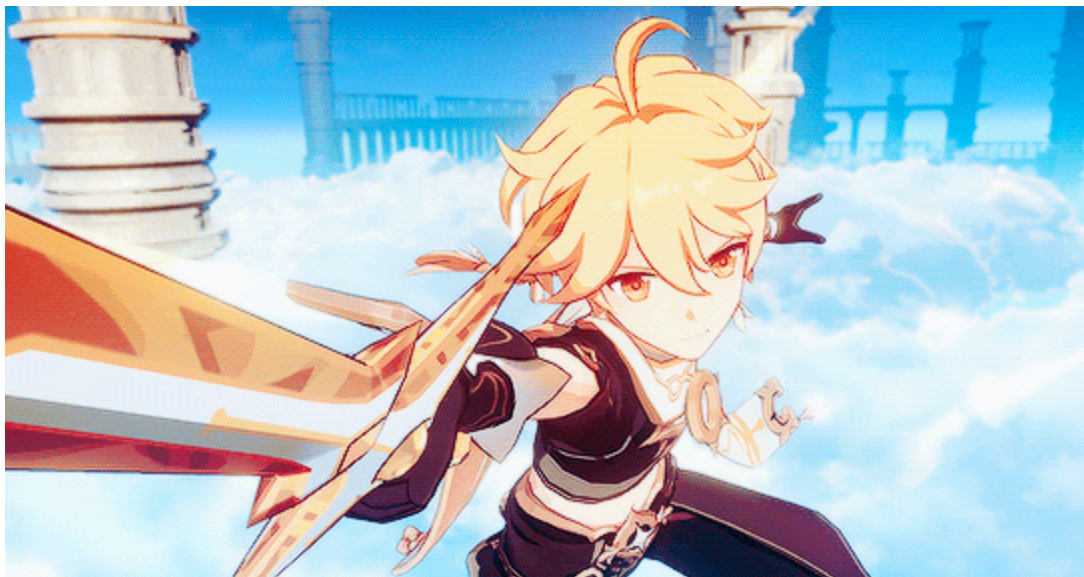
2. (30) Implementasi *proof-of-concept*

3. (10) Video demonstrasi

Selain itu, berikut merupakan tugas-tugas tambahan yang dapat dilakukan untuk memberikan nilai bonus:

1. (5) Melakukan pemilihan alternatif teknologi yang didukung dengan data, penelitian, dan atau proses pemilihan yang lengkap dan runut.
2. (10) Melakukan *deployment* terhadap aplikasi

Perlu dicatat bahwa nilai maksimum tugas adalah 100, dan poin bonus hanya dapat membantu “menambal” nilai yang tidak sempurna.



~

夕暮れに頬が染まって

色付いた町が心煽る

特別な夜に言葉を紡ぐ

~

Duke