

NIM : 2311521012
Nama : Aldi
Mata Kuliah : Rancangan Perangkat Lunak
Kelas : B

A. Pemahaman Dasar

- 1) Apa yang dimaksud dengan Design Model dalam rekayasa perangkat lunak?

Design model adalah representasi visual atau konseptual dari sistem yang akan dikembangkan, yang dihasilkan berdasar analisis kebutuhan. Tujuannya untuk menjembatani kebutuhan fungsional (apa yang harus dikerjakan sistem) menjadi solusi teknis (bagaimana sistem mengerjakannya).

- 2) Sebutkan dan jelaskan empat komponen utama design model!

a. Data/Class Design

Menjelaskan struktur data yang digunakan dan bagaimana objek atau class saling terhubung.

b. Architectural Design

Menentukan struktur sistem tingkat tinggi, termasuk pembagian subsistem, modul, dan interaksi antar komponen.

c. Interface Design

Mencakup antarmuka pengguna (UI/UX) dan antarmuka antar modul atau sistem lain (API). Fokusnya pada interaksi.

d. Component-Level Design

Mendeskripsikan secara detail bagaimana setiap modul atau komponen akan diimplementasikan, termasuk logika internalnya.

- 3) Mengapa design model sangat bergantung pada hasil dari analysis model?

Karena analysis model mendefinisikan apa yang dibutuhkan oleh pengguna

dan sistem, sedangkan design model menjawab bagaimana kebutuhan itu akan diwujudkan secara teknis. Tanpa analisis yang benar, desain bisa salah arah atau tidak relevan terhadap kebutuhan pengguna. Analysis model adalah pondasi dari desain.

B. Hubungan antara Analysis Model dan Design Model

- 1) Jelaskan bagaimana scenario-based model berkontribusi pada design level (misalnya interface/component design)!

Scenario-based model (seperti use case dan sequence diagram) membantu perancang memahami alur interaksi antara pengguna dan sistem. Informasi ini digunakan untuk mendesain antarmuka pengguna (UI) agar sesuai dengan alur dan kebutuhan pengguna, dan untuk menentukan komponen dan metode yang harus ada agar dapat memenuhi alur skenario.

- 2) Apa yang dihasilkan dari class-based model saat diturunkan ke tahap desain?
 - a. Desain struktur class, termasuk atribut, metode, relasi (asosiasi, pewarisan, agregasi).
 - b. Class diagram yang lebih rinci (implementatif).
 - c. Struktur data dan object design yang akan diimplementasikan di kode program.
- 3) Bandingkan penggunaan activity diagram dan data flow diagram (DFD) dalam mendukung desain sistem.
 - a. Activity diagram berfokus ke menggambarkan urutan aktivitas dalam sistem. Sementara DFD lebih fokus pada bagaimana data bergerak antar proses dan tempat penyimpanan data.
 - b. Activity diagram memakai pendekatan berorientasi objek atau sistem secara keseluruhan, sedangkan DFD menggunakan pendekatan yang berorientasi pada proses, yaitu menekankan bagaimana data diproses

dalam sistem.

- c. Activity diagram menampilkan alur aktivitas, mirip seperti flowchart. DFD menampilkan bagaimana data masuk dan keluar dari sebuah proses, serta hubungan antar elemen data dalam sistem.
- d. Kelebihan activity diagram adalah kemampuannya menggambarkan alur logika dengan jelas. Sementara itu, DFD lebih unggul untuk memahami bagaimana struktur aliran data bekerja di dalam sistem.

C. Hubungan antara Analysis Model dan Design Model

- 1) Berdasarkan sistem yang telah kamu analisis (dalam tugas sebelumnya), bagaimana kamu membayangkan struktur desainnya (misal: bagaimana UI, data class, atau arsitektur disusun)?

- a. UI Design

Terdapat antarmuka untuk mahasiswa (scan QR), dosen/admin (validasi kehadiran, melihat data), dan login sistem.

- b. Data/Class Design

Class: Mahasiswa, Dosen, Absensi, Kelas, QRGenerator, UserSession.

Relasi: Mahasiswa melakukan Absensi untuk Kelas. QRGenerator menghasilkan kode unik untuk Absensi.

- c. Architectural Design

Menggunakan arsitektur MVC (Model-View-Controller):

- Model: Logika data dan database.
- View: Tampilan pengguna (web atau mobile).
- Controller: Menangani permintaan dari pengguna dan menghubungkannya ke model.

- 2) Apa tantangan paling mungkin jika kamu harus menerjemahkan model analisis ke desain?

- a. Menentukan tingkat detail yang tepat dalam desain agar tidak terlalu umum atau terlalu kompleks.

- b. Menghadapi ketidaksesuaian antara kebutuhan pengguna dan batasan teknis.
 - c. Menjaga konsistensi antara model analisis (konseptual) dan model desain (teknis).
 - d. Kompleksitas UI dan logika QR code seperti waktu kadaluwarsa, validasi lokasi, dll.
- 3) Bagaimana kamu menilai kelebihan pendekatan berorientasi objek dibandingkan pendekatan berbasis proses?
- a. Modularitas
Pendekatan berorientasi objek memungkinkan sistem dibangun dalam bentuk modul-modul kecil berupa objek dan class, yang masing-masing memiliki tanggung jawab sendiri. Hal ini membuat sistem lebih mudah dipelihara dan diperluas karena setiap bagian dapat dikelola secara terpisah.
 - b. Reusability
Dengan pendekatan ini, komponen atau class yang sudah dibuat dapat digunakan kembali di sistem lain tanpa perlu membangun dari awal. Ini sangat membantu dalam menghemat waktu dan usaha dalam pengembangan perangkat lunak.
 - c. Representasi Dunia Nyata
Pendekatan berorientasi objek sangat cocok untuk memodelkan sistem yang melibatkan entitas nyata, seperti mahasiswa, dosen, atau kelas. Objek-objek dalam sistem mencerminkan entitas nyata dengan atribut dan perilakunya, sehingga lebih mudah dipahami.
 - d. Fleksibilitas
Sistem berorientasi objek memudahkan pengembangan proyek besar karena tugas-tugas dapat dibagi berdasarkan objek atau class. Ini memungkinkan tim bekerja secara paralel dan sistem menjadi lebih fleksibel untuk diubah atau dikembangkan di masa depan.