

Bagian A: Pemahaman Dasar

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

Design Model adalah representasi abstrak dari bagaimana sistem perangkat lunak akan dibangun berdasarkan hasil analisis kebutuhan. Jika analysis model fokus pada apa yang dibutuhkan oleh pengguna, maka design model menjawab bagaimana kebutuhan itu direalisasikan dalam bentuk struktur perangkat lunak, termasuk arsitektur, komponen, antarmuka, dan data.

2. Sebutkan dan jelaskan empat komponen utama design model!

1) Data/Class Design

Menjelaskan struktur data dan class (dalam pendekatan OOP) yang akan digunakan oleh sistem. Ini mencakup atribut, relasi antar class, serta hirarki pewarisan.

2) Architectural Design

Menentukan struktur tinggi dari sistem: bagaimana komponen utama disusun dan berinteraksi. Misalnya arsitektur layered, MVC, atau client-server.

3) Interface Design

Mendeskripsikan bagaimana komponen berinteraksi satu sama lain (interface internal), dan bagaimana sistem berinteraksi dengan pengguna (UI/UX).

4) Component-Level Design

Merinci struktur internal dari tiap komponen, misalnya bagaimana sebuah modul diimplementasikan, fungsi-fungsi yang dimiliki, dan logika kontrol di dalamnya.

3. Mengapa design model sangat bergantung pada hasil dari analysis model?

Design model dibentuk berdasarkan kebutuhan fungsional dan non-fungsional yang ditemukan dalam analysis model. Tanpa pemahaman yang akurat dari analysis model, desain bisa salah arah, membangun fitur yang tidak dibutuhkan atau melewatkan fitur penting. Design model adalah proses transformasi dari kebutuhan menjadi solusi teknis.

Bagian 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 menggambarkan alur interaksi pengguna dengan sistem. Ini membantu desainer:

- 1) Menentukan fungsi-fungsi utama yang harus disediakan pada UI.
- 2) Memetakan skenario ke komponen spesifik yang perlu dibuat.
- 3) Memastikan alur antarmuka logis dan sesuai kebutuhan pengguna.
- 4) Membantu dalam pembuatan test case awal dari skenario yang ada.

2. Apa yang dihasilkan dari class-based model saat diturunkan ke tahap desain?

Class-based model menghasilkan:

- 1) Blueprint dari class yang akan digunakan pada level kode.
- 2) Struktur atribut dan method, termasuk visibility dan hubungan antar class (inheritance, association).
- 3) Pondasi untuk data design dan struktur objek saat runtime.
- 4) Dasar untuk desain database jika ada pemetaan class ke tabel.

3. Bandingkan penggunaan activity diagram dan data flow diagram (DFD) dalam mendukung desain sistem.

Aspek	Activity Diagram (UML)	Data Flow Diagram (DFD)
Pendekatan	Berorientasi objek	Berbasis proses/data
Tujuan	Menunjukkan alur aktivitas dan kontrol	Menggambarkan aliran data antar proses
Detail Kontrol	Memiliki kondisi, loop, branching	Fokus pada alur data, kurang kontrol logika
Relevansi Desain	Cocok untuk sistem berbasis event/state	Cocok untuk sistem pemrosesan data (batch/stream)
Kelebihan	Lebih ekspresif dan modern	Lebih mudah dipahami untuk proses linier

Bagian C: Refleksi Penerapan

1. Berdasarkan sistem yang telah kamu analisis, bagaimana kamu membayangkan struktur desainnya? Struktur desain sistem Absensi Mahasiswa Berbasis QR Code dapat divisualisasikan sebagai berikut:

- 1) UI Design:
 - a) Dosen: Antarmuka berbasis web (dashboard) dengan fitur login, pembuatan QR Code, dan tampilan rekap absensi.
 - b) Mahasiswa: Aplikasi mobile (Android) untuk login dan memindai QR Code sesi aktif.
 - c) Admin: Antarmuka web untuk mengelola data pengguna dan jadwal kuliah.
- 2) Class/Data Design (berbasis OOP):
 - a) Mahasiswa, Dosen, SesiKuliah, dan Absensi merupakan class utama.
 - b) Hubungan antar class mengacu pada ERD, dengan relasi one-to-many seperti:
 - i. Dosen → SesiKuliah
 - ii. SesiKuliah → Absensi
 - iii. Mahasiswa → Absensi
- 3) Architectural Design:
 - a) Client-Server:
 - i. Frontend (mobile & web) berkomunikasi ke backend melalui REST API.
 - b) 3-layer architecture:
 - i. Presentation Layer: UI mobile & web.
 - ii. Application Layer: Backend API (otentikasi, generate QR, validasi, simpan absensi).
 - iii. Data Layer: Database MySQL/PostgreSQL untuk menyimpan data entitas dan transaksi absensi.

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

Beberapa tantangan utama:

- 1) Transisi dari DFD ke Object-Oriented Design
Karena model analisis menggunakan pendekatan proses (DFD), sedangkan desain idealnya menggunakan class dan komponen, perlu usaha untuk mengkonversi proses menjadi entitas dan method yang modular.
- 2) Manajemen waktu aktif QR Code secara real-time
Mendesain sistem yang bisa menangani validasi waktu aktif QR Code secara akurat dan aman cukup menantang, terutama untuk menghindari penyalahgunaan (misalnya QR yang disebar).
- 3) Sinkronisasi antarmuka pengguna dengan backend
Menjamin proses scan, validasi, dan penyimpanan berjalan cepat (< 2 detik) di berbagai perangkat dan kondisi jaringan.
- 4) Keamanan dan autentikasi
Desain harus mengakomodasi enkripsi QR Code, validasi session, serta kontrol akses berbasis peran (mahasiswa, dosen, admin).

3. Bagaimana kamu menilai kelebihan pendekatan berorientasi objek dibandingkan pendekatan berbasis proses?

Pendekatan berorientasi objek lebih unggul dalam konteks sistem ini karena:

Aspek	Berorientasi Objek	Berbasis Proses (DFD)
Modularitas	Setiap entitas (Mahasiswa, Absensi) diwakili class	Proses sulit direpresentasikan sebagai unit independen
Skalabilitas	Mudah dikembangkan dan ditambahkan fitur baru	Harus menambah proses baru dan relasi DFD kompleks
Reusabilitas	Class dan method bisa digunakan ulang di berbagai modul	Kurang mendukung reuse tanpa struktur tambahan
Relevansi terhadap implementasi	Lebih dekat dengan pemrograman nyata (OOP)	Harus diubah dulu ke class/fungsi untuk bisa diprogram
Pengelolaan data	Menggabungkan data dan operasi dalam satu objek	Data dan proses dipisah, rentan inkonsistensi

Namun pendekatan berbasis proses tetap bermanfaat untuk tahap awal analisis karena memudahkan pemangku kepentingan non-teknis memahami alur kerja sistem.