

Vanica Zhafira Zahra - 2311523007

### Bagian A : Pemahaman dasar

1. Apa yang dimaksud dengan Design model dalam RPL?

↳ Gambaran detail dari model desain dari perangkat lunak yang akan dibangun, difokuskan pada aspek implementasi dan struktur teknis, bagaimana sistem ini akan diwujudkan.

2. Sebutkan dan jelaskan 4 komponen utama design model.?

a. Data / desain class data : Proses fundamental dalam mengonversi informasi abstrak yang telah diidentifikasi pada fase analisis model menjadi struktur data yang konkret dan dapat diimplementasi. Skema basis data, pengaturan struktur file, serta penentuan representasi data internal.

b. Desain arsitektur : kerangka menyeluruh perangkat lunak, penentuan hubungan antara elemen<sup>2</sup> struktur utama dalam Software.

c. Antarmuka desain : bagaimana sistem akan berinteraksi dengan entitas eksternal, perancangan UI/UX, elemen interaktif dan alur navigasi bagi pengguna.

d. Desain Tingkat komponen : berfokus pada detail internal dari setiap modul atau komponen dalam sistem, struktur data internal, algoritma yg akan diimplementasikan.

3. Mengapa design model sangat bergantung pada hasil analisis model?

Analisis model menjadi dasar yang memuat kebutuhan serta spesifikasi dari sistem. Tanpa pemahaman yang kuat terhadap hasil analisis, desain teknis yg dihasilkan tidak akan relevan atau efektif.

Intinya, design merupakan wujud teknis dari hasil analisis.



Bagian B : Hubungan antara analisis model dan design model.

1. Jelaskan bagaimana scenario based berkontribusi pada design level ?

↳ model berbasis skenario seperti use case & activity diagram, berperan penting dalam tahap design, terutama pada antarmuka & komponen sistem antara lain :

a. Design antarmuka : use case menunjukkan interaksi pengguna dengan sistem yg bisa diterjemahkan ke dalam alur antarmuka, activity diagram membantu merancang alur kerja UI yg jelas dan efisien.

b. Design komponen : Skenario membantu mengidentifikasi fungsi utama sistem yang kemudian dipetakan ke komponen, sekaligus menentukan tanggung jawab dan kolaborasi antar komponen.

2. apa yang dihasilkan dari class-based model saat diturunkan ke tahap design ?

a. Struktur data yang terperinci : kelas² abstrak dianalisis lebih lanjut menjadi kelas desain dg atribut, tipe data, visibilitas, dan metode yang lebih spesifik. kelas antar kelas diterjemahkan ke bentuk teknis.

b. Tanggung jawab dan model : setiap kelas desain diberikan peran jelas dengan metode yang dirinci,

c. Pemetaan ke komponen : kelas² tersebut kemudian dipetakan menjadi komponen perangkat lunak, baik sbg satu kelas per komponen maupun gabungan kelas.

3. Bandingkan penggunaan activity diagram dan DFD dlm mendukung desain sistem.

↳ a. activity diagram : Menekan alur kontrol dan urutan aktivitas, cocok utk merancang navigasi UI, interaksi komponen, dan proses kompleks. kelebihan adalah memodelkan proses paralel, tetapi kurang soal detail data.



b. DFD Fokus pada aliran dan transformasi data. Berguna untuk memahami kebutuhan data, penyimpanan dan pembagian model sistem. Kuat dalam aspek data, tapi lemah dalam logika kontrol & urutan waktu. activity diagram unggul dalam memodelkan alur proses, sedangkan DFD unggul dalam memetakan aliran data.

Bagian C : Refleksi penerapan.

1. Berdasarkan sistem yang telah kamu analisis, bagaimana kamu membayangkan struktur desainnya?

Struktur design SIPOPeD yaitu :

a. UI (antarmuka pengguna) : peserta akan mengakses formulir pendaftaran Seblehmana yg responsif (google form / langsung). Panitia / admin akan dapat dashboard untuk melihat dan mengelola data peserta, serta mengirim link seminar

b. Data class : entitas seperti peserta, formulir\_pendaftaran, data\_terverifikasi, dan Distribusi-Informasi akan menjadi tabel utama. relasi antar table bersifat 1 : N sesuai ERD nya.

c. arsitektur sistem : berbasis web dg arsitektur 3 lapis :

- presentasi : Akses melalui browser
- logika Bisnis : backend dgn PHP
- Akses data : menggunakan MySQL

Sistem juga terintegrasi dg google form dan Google sheet API mengirim email via SMTP.

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

↳ Detail teknis dari model analisis, perlu dijabarkan lebih detail ke komponen teknis seperti tipe data, metode, aturannya.



- Translasi proses ke UI/UX mengubah proses seperti validasi atau distribusi informasi menjadi interaksi UI yang efisien bisa menjadi tantangan.
- Integrasi API mendesain komunikasi data antara sistem dg Google sheets API eksternal agar sinkron dan aman juga perlu perhatian khusus.

5. Bagaimana kamu menilai kelebihan pendekatan berorientasi objek dibanding pendekatan berbasis proses?

- ↳ • dengan oop menggunakan modularitas, dengan menisankan sistem ke dalam objek dg tanggung jawab spesifik sehingga mudah dikembangkan dan dirawat
- kelas dan objek digunakan kembali di bagian sistem lain atau dimasa depan.
- dgn ini lebih realistis atau lebih dekat dengan dunia nyata dimana mudah dipahami karena mencerminkan entitas dan interaksi yang nyata.

Sedangkan berbasis proses (BPD) memang kuat untuk menggambarkan aliran data, tapi kurang dalam hal struktur dan tanggung jawab objek.