

Laporan Pemrograman Mobile

Nama : Sebastian Abe Santoso

NIM : 2410817210002

Understanding

Deskripsi

22/02/2026 : Saya selesai menonton modul 2 bagian 7-9, di sini saya berfokus untuk membuat fondasi untuk pengetahuan umum. Disini saya belum mencerna seluruhnya.

Insight

Di sini saya dapat sedikit pemahaman mengenai project yang akan dikerjakan selama modul 2, 3, dan 4, serta C4 model yang diajarkan pada chapter 9.

Impact

Saya dari dulu sering dengar bahwa waterfall model itu jarang dipakai dan juga disarankan untuk tidak memakainya, tetapi ironisnya, pada chapter 7 saya baru dapat memahami kenapa. ~~Waterfall model tidak dipakai karena mencoba melakukan semua proses dalam 1 iterasi, sehingga penemuan atau jika ada hal yang bermasalah pada tahap implementasi tidak bisa dimanfaatkan.~~ Setelah saya baca artikel dari penemunya di <https://beza1e1.tuxen.de/waterfall.html>, Royce mempresentasikan waterfall model linear dengan satu iterasi sebagai contoh problematik dan memberi sugesti untuk mengiterasi, tetapi industri saat itu lebih memilih untuk mengadopsi versi linearnya dikarenakan lebih mudah untuk dipantau, yang menghasilkan legacy software dengan dokumentasi yang memiliki ratusan halaman.

Sebenarnya waterfall model asli yang dijabarkan oleh Royce masih bisa digunakan, dan sekarang versi modifikasinya seperti V-Model merupakan standar untuk proyek yang kegagalan dibayar dengan harga mahal, misalnya software untuk pesawat atau medis, karena tidak mungkin untuk testing software harus meraup nyawa, jika menggunakan Agile yang bermotto "Fail fast, learn faster" pasti berakhir malapetaka. Saya baca lebih lanjut dan modern developer tidak menyukai waterfall model karena menyebabkan "Frozen Requirements" di mana jika requirements sudah ditetapkan maka tidak bisa diubah, jadi harus di paksakan pada tahap tahap selanjutnya walau tidak diinginkan, disebut "Change is expensive", membuatnya sangat tidak cocok untuk proyek proyek era modern yang memiliki requirement yang cepat berganti.

Terakhir, agile dan waterfall sebenarnya tidak berlawanan, hanya beda filosofi untuk masalah yang berbeda. Saya baca dari Reddit, "Waterfall: You think you know what you want you're unreasonably confident about how you'll get there. Agile: you have a vague idea what you want and you are uncertain how to get there." Dalam management triangle, "In waterfall you have requirements and you plan for cost and time. In agile you have capacity and time and figure out requirement / scope."

Deskripsi

23/02/2026 : Mencatat bagian 7-11, dan menonton ulang yang belum paham, kemudian mengerjakan tugas pada bagian 10.

- tugas di GitHub:
https://github.com/SebastianAbeSantoso/Me/blob/main/LAPORAN_PEMROGRAMAN_MOBILE/week%202/Exercise_1.jpeg atau di Lucidchart: https://lucid.app/lucidchart/35fbd26-5cb0-40ff-8ff7-28d3f14dbaa3/edit?invitationId=inv_31ac1de4-b752-4504-bc2e-571fb39c0726&page=0_0#
- catatan:
https://github.com/SebastianAbeSantoso/Me/blob/main/LAPORAN_PEMROGRAMAN_MOBILE/week%202/CATATAN_MODUL2_EN.md.

Insight

Di sini saya dapat insight berupa model yang digunakan disini bisa disebut dengan "Iterative Development" di mana understanding, design, development semua dilakukan berkali-kali untuk membuat produk yang sesuai dengan pasar sekarang, ini juga termasuk dari Agile Model. Tetapi tidak selalu harus berurutan, sesuai dengan keperluan. C4 Model juga dijelaskan pada chapter 9, yang disebut oleh mentornya dengan "A Google Maps For Your Code", yaitu metode diagram yang bertujuan untuk memvisualisasi sebuah arsitektur dengan 4 opsi detail yang berbeda (Context Diagram > Container Diagram > Component Diagram > Code Diagram). Saya juga belajar untuk menggambar sebuah container diagram dari C4 Model disini.

Impact

Saya pertama kali mendengar mentor nya menjelaskan metode iterasi di sini sangatlah menarik, karena kode-kode yang saya buat sebelumnya itu kurang lebih menggunakan waterfall model yang linear, karena itu saya akan memikirkan penggunaan model Agile lebih sering di masa depan. Model C4 juga sangat menarik bagi saya, penjelasannya sebagai google maps itu relatable, tetapi saya pikir agak sulit menggunakannya untuk proyek personal, lebih cocok untuk hal yang membutuhkan dokumentasi dan pemikiran yang lebih matang.

Deskripsi

24/02/2026 : Saya menyelesaikan menonton sisa modul 2, yaitu bagian 12, 13, dan 16, tetapi belum sepenuhnya mencerna materi.

Insight

Di sini saya pertama kali mendengar dan belajar tentang Architectural Drivers, Requirements, dan Decisions. Mereka saling berhubungan dan saling berkoordinasi untuk membuat sebuah arsitektur. Architectural Driver bisa disebut dengan hal yang dapat memengaruhi arsitektur, dan Architectural Requirement juga dapat dibuat berdasarkan ini. Architectural Requirement adalah seperti checklist yang perlu diselesaikan agar arsitekturnya sukses, ini yang akan membuat decisions nanti. Architectural Decisions bisa dijelaskan sebagai keputusan-keputusan yang berada di sisi arsitektural pada spektrum design dan arsitektur.

Impact

Saat pertama kali mendengar mengenai ketiga konsep ini, saya berpikir lebih baik untuk membuat satu per satu fondasi (Decisions), dan menentukan hal lainnya berdasarkan fondasi yang telah dibuat, daripada membuat semuanya sekaligus dan mendapatkan analysis paralysis, akan saya perhatikan lebih lanjut

mengenai ketiga ini saat saya membuat sebuah software. Saat mata kuliah APS saya baru mendengar tentang FR dan NFR, tetapi disini mendalami lebih lanjut, saya baru terpikir saat mentor menyebut bahwa quality attribute bisa juga disamakan dengan NFR.

Deskripsi

25/02/2026 : mencatat bagian 12-16 dan menonton ulang yang belum saya pahami, kemudian mengerjakan tugas pada bagian 14, yang bisa dilihat di

https://github.com/SebastianAbeSantoso/Me/blob/main/LAPORAN_PEMROGRAMAN_MOBILE/week%202/Exercise_2.md

Saya juga menyelesaikan laporan ini dalam hari ini.

Insight

Saya memahami tentang ketiga arsitektur (Architectural Drivers, Requirements, Decisions) lebih lanjut dengan mencatat bagian 12-16, jika disamakan dengan game development, misalnya Architectural Drivers, ini adalah hal utama yang mendorong arsitektur, dapat disamakan dengan "Game Pillars", yaitu hal yang menentukan pengalaman yang ingin ditekankan. Ini nantinya memengaruhi Architectural Requirements, yang bisa disamakan dengan sebuah core feature atau daftar fitur-fitur yang diimplementasi yang menentukan kesuksesan game dibuat, dapat dipikirkan juga sebagai checklist, ini yang akan membuat decisions. Terakhir, Architectural Decisions adalah fondasi yang paling dasar, penting dan ditentukan di awal proyek, dalam gamedev ini bisa disebut sebagai "North Star", sulit diganti, penting, dan keputusan lainnya terpengaruh oleh ini.

Impact

Saat memikirkan ketiga arsitektur itu sebagai hal yang didalam game dev, saya jadi lebih paham menggunakannya diluar frontend architecture, bisa juga diaplikasikan di lain. Karena pada intinya, Architectural drivers mendorong arsitektur, dan memengaruhi Requirements. Requirements adalah kriteria sukses sebuah arsitektur, yang nantinya akan membuat Decisions. Decisions adalah keputusan awal yang penting dan sulit diubah, dan bisa memengaruhi drivers. Ketiga ini bisa dibuat dengan loop, bukan linear, karena mereka bisa memengaruhi satu sama lain.

Terakhir, mereka tidak hanya aturan-aturan, tetapi juga dapat dirakit seperti pertanyaan:

1. Architectural Driver: "Kenapa ini ada?" atau "Mengapa sistem ini dibangun, nilai apa yang harus diberikan, dan faktor apa yang paling memengaruhi bentuk arsitekturnya?"
 2. Architectural Requirements: "Apa yang harus dicapai?" atau "Berdasarkan tujuan, nilai, dan faktor pendorong tersebut, kriteria keberhasilan apa yang harus dipenuhi sistem ini?"
 3. Architectural Decisions: "Bagaimana kita membuatnya?" atau "Dengan tujuan, nilai, dan kriteria keberhasilan yang telah ditetapkan, pendekatan arsitektural apa yang paling tepat, dan kompromi apa yang secara sadar kita terima?"
- ◦ Setelah ketiga pertanyaan itu, maka pertanyaan terakhir adalah "Apa konsekuensinya?" Karena kita tidak hanya menentukan pilihan, tetapi juga merekam apa efek atau kegunaannya.
-

Laporan ini berada di:

https://github.com/SebastianAbeSantoso/Me/edit/main/LAPORAN_PEMROGRAMAN_MOBILE/week%202/LAPORAN_MODUL2_PEMROGRAMAN_MOBILE.md