

D-01 Training

Programming

Introduction



TABLE OF CONTENT

Day 1 - Sesi 1

01

History of Apps & Web

02

Product Concept

03

Software Development Lifecycle (SDLC)

04

Techstack

1. Sejarah Program dan Web

Melihat bagaimana dulu aplikasi dibuat



Mengapa desain database penting?

Aplikasi

Web

Komputer

**Bahasa
Pemrograman**

Internet

Aplikasi Desktop

Aplikasi Web

Aplikasi Mobile

Tugas Individu

Silahkan riset untuk mencari hal - hal penting apa saja yang terkait dengan kata kunci di sebelah..

Komputer

**Bahasa
Pemrograman**

Internet

Aplikasi Desktop

Aplikasi Web

Aplikasi Mobile

2. Konsep Produk

Penyelesaian masalah yang kita buat menjadi bernilai ekonomi



Konsep yang dipakai dalam membuat “produk terbaik”

Konsep Produk

Konsep Produksi

Filosofi

Konsep Produk

Konsumen menginginkan produk yang berkualitas tinggi dan mempunyai banyak fitur.

Konsep Produksi

Konsumen menginginkan produk yang tersedia di mana - mana dan harganya serendah mungkin.

Cara

Konsep Produk

Secara terus - menerus meningkatkan kualitas produk.

Konsep Produksi

Memperbesar efisiensi produksi dan kanal distribusi.

Tujuan

Konsep Produk

Menurut Kamu apa?

Konsep Produksi

Menurut Kamu apa?

Keunggulan

Konsep Produk

Menurut Kamu apa?

Konsep Produksi

Menurut Kamu apa?

Kelemahan

Konsep Produk

Menurut Kamu apa?

Konsep Produksi

Menurut Kamu apa?

Diskusi: untuk pengembangan perangkat lunak, konsep manakah yang lebih cocok dipakai?

Konsep Produk

**Konsep
Produksi**

Buat Product
Concept
Statement

Evaluasi
Product
Concept
Statement
terhadap
Business
Goals

Lakukan
Concept
Testing

Kembangkan
Produk yang
sesuai dengan
Product
Concept yang
disukai
Pelanggan

Gunakan
Product
Concept ke
Dalam
Marketing
Strategy

Implementasi Konsep Produk

Buat Product
Concept
Statement

Evaluasi
Product
Concept
Statement
terhadap
Business
Goals

Lakukan
Concept
Testing

Kembangkan
Produk yang
sesuai dengan
Product
Concept yang
disukai
Pelanggan

Gunakan
Product
Concept ke
Dalam
Marketing
Strategy

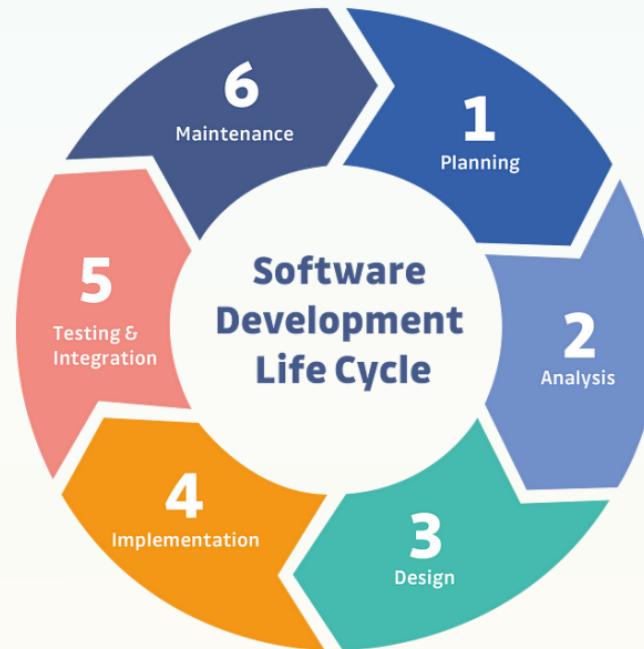
**Diskusi: di bagian manakah kemampuan
programming kamu dapat digunakan?**

3. SDLC

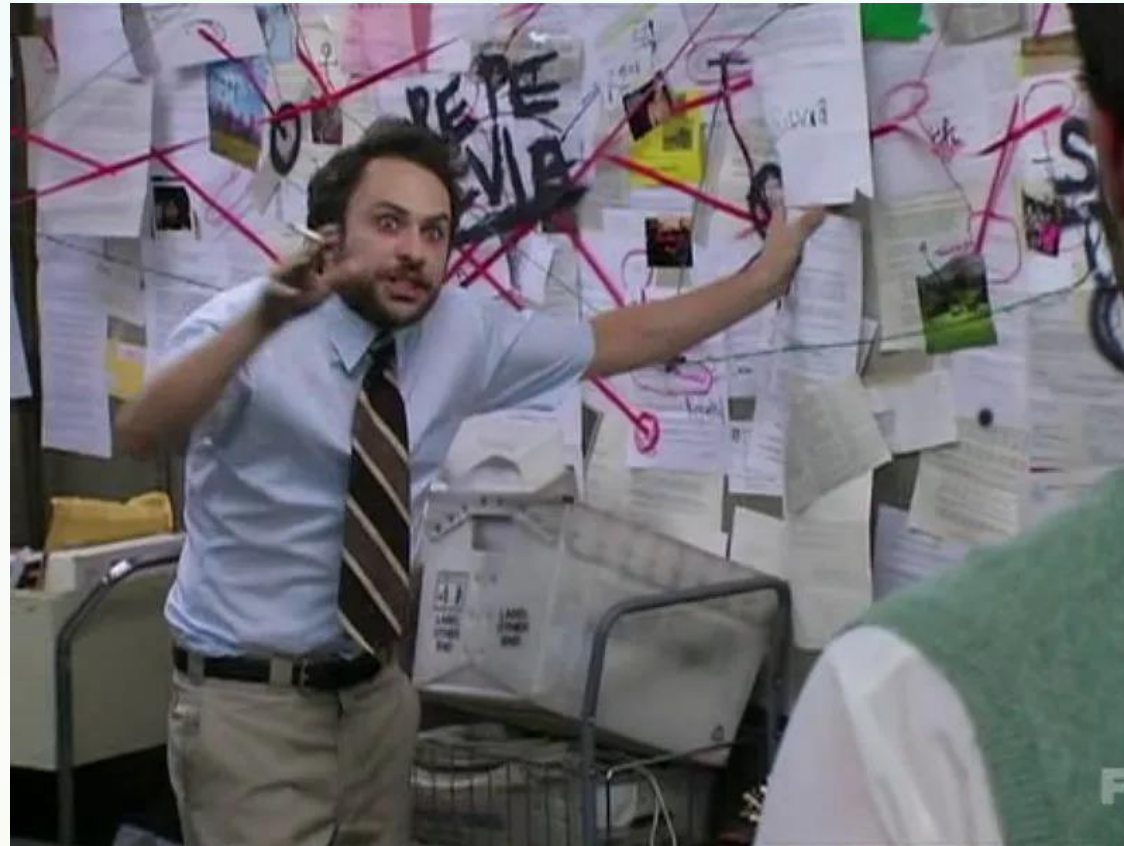
Siklus hidup pengembangan aplikasi



Software Development Life Cycle adalah metodologi yang menggunakan proses - proses tertentu dari awal hingga selesai untuk mengembangkan perangkat lunak.



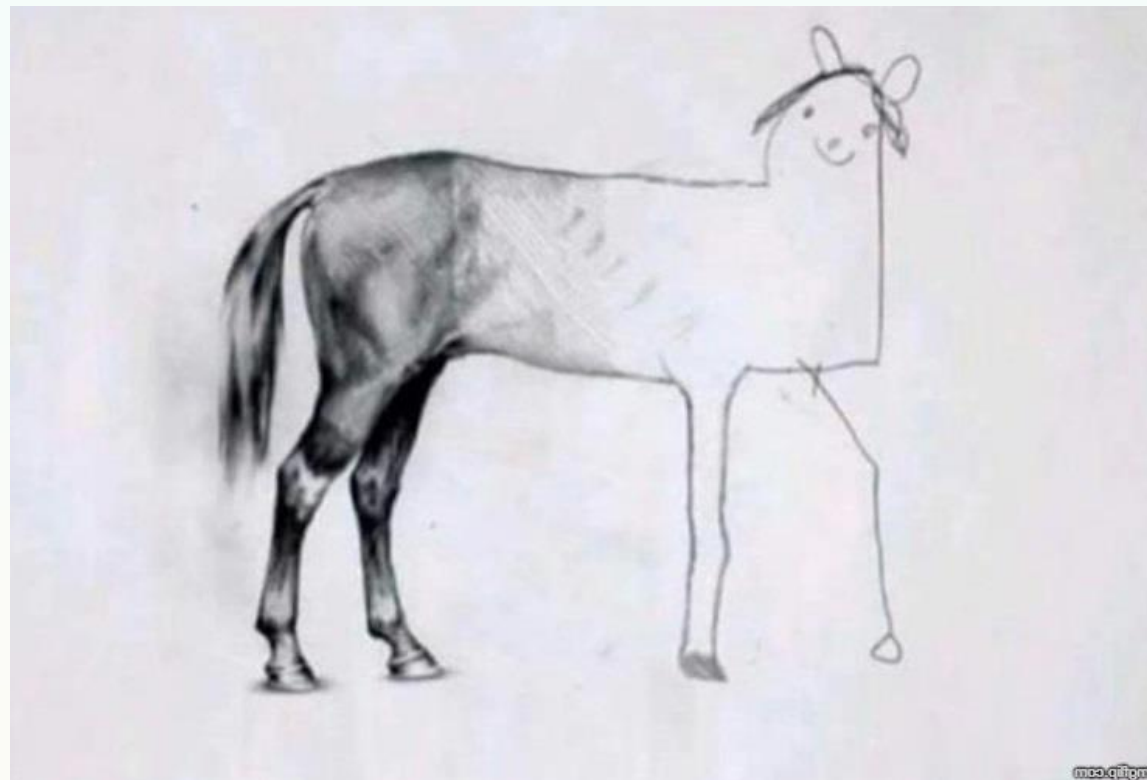
Planning



Analysis



Design



Implementation



Testing



Deployment



Pengelompokan Model SDLC

Adaptive

Membuat perangkat lunak dengan melakukan analisa dan perencanaan secukupnya dan melakukan desain, pengembangan, testing dan implementasi (jika mungkin) atas kebutuhan di atas.

Predictive

Membuat perangkat lunak dengan melakukan analisa kebutuhan dan perencanaan secara menyeluruh terlebih dahulu, lalu melakukan desain, pengembangan, testing dan implementasi sampai **semua kebutuhan** sudah dibuat.

Keunggulan

Adaptive

Lebih mungkin meningkatkan kepuasan pelanggan karena perangkat lunak sudah bisa dipakai dalam waktu singkat dan bisa berubah sesuai kebutuhan.

Predictive

Sebagian besar identifikasi dan manajemen risiko dapat dilakukan di awal.

Kelemahan

Adaptive

Perencanaan lebih minim karena hanya sedikit bagian di produk yang sudah jelas kebutuhannya.

Predictive

Sangat membutuhkan analisa kebutuhan dan perencanaan yang baik.

Contoh Model

Adaptive

Semua model berbasis Agile:
Scrum, Kanban, Extreme Programming,
Feature-Driven Development.

Predictive

Waterfall, Iterative & Incremental,
Spiral, V-Model.

Waktunya Berdiskusi!

- Manakah kelompok model yang lebih baik, adaptive atau predictive?
- Berikan contoh bidang bisnis yang pengembangan perangkat lunaknya memakai masing - masing model adaptive dan predictive.

4. Tech Stack

Teknologi yang digunakan saat membuat aplikasi



Apa itu Tech Stack?

Tech Stack adalah gabungan beberapa framework, tools dan teknologi yang kita “tumpuk” dalam membuat sebuah produk.

Diskusi:

Menurut kamu, kenapa ya kita perlu memakai tech stack tertentu?

Ternyata, tech stack menentukan tipe perangkat lunak yang dapat kita buat, level dari penyesuaian yang dapat dilakukan, dan resource yang diperlukan untuk membuat dan menjalankan perangkat lunak.

Contoh Komponen Tech Stack

| Tech Stack Component | Technologies |
|-----------------------|---|
| Frontend Framework | Vue, Angular, Bootstrap |
| Frontend Library | React |
| Programming Languages | Java, C, PHP, Python, Swift, Javascript |
| App & Web Framework | Spring, Django |
| Database | MySQL, MongoDB, PostgreSQL |
| Event & Messaging | Kafka |
| Cloud Infrastructure | AWS, Azure, Google Cloud |
| Virtualization | Kubernetes, Docker |
| Operating Systems | Windows, Linux, MacOS |
| Environment | Node.js, JRE |

Thank You

Programming

Introduction

