**PERSIAPAN BELAJAR**

Appendiks adalah lampiran detail

Glosarium adalah daftar alfabet suatu istilah

Pseudocode adalah Bahasa mesin yang diterjemahkan ke dalam Bahasa manusia sehingga mudah di pahami

**MEMAHAMI KEBUTUHAN APLIKASI**

Teknologi mempermudah pekerjaan menjadi cepat dan efisien

Penting penentuan kebutuan apikasi dari sisi pengguna

Cara masalah dan temukan solusinya. Seperti manusia menemukan roda

Software Development Life Cycle

1. Planning: Perencanaan
2. Analysis: Analisis
3. Design: Desain
4. Development: Pengembangan
5. Testing: Pengujian
6. Deployment : Penggelaran
7. Maintenance: Pemeliharaan

**User Requirement Specifiations**

User Recuirement Specification / Document (URS / URD). URS itu semacam blueprint agar tidak ada komunikasi antara developer dengan stakeholders.

Dalam pembuatan URS tidak boleh ada jargon teknis

Untuk mendapatkan kebutuhan user, kita perlu melakukan *requirement gathering* yang merupakan proses pengumpulan informasi dari para stakeholders. Dapat melalui interview, user stories, straw man document, dan prototyping.

**Interview** adalah tanya jawab

**User stories** adalah cerita mini seperti: saya adalah … saya menginginkan … sehingga …

**Straw Man Document** salahsatunya adalah flowchart, storyboard, dll. Gunanya untuk menentukan ide mentah agar tidak langsung ngoding.

**Prototyping** itu adalah design interaktif. Bisa menggunakan figma dll

**Dokumen Spesifikasi Teknis Applikasi**

Spesifikasi teknis aplikasi adalah dokumen yang menjelaskan secara detail bagaimana sistem bekerja, termasuk fungsionalitas, proses, batasan, dan kriteria penerimaan fitur agar dapat diimplementasikan dengan jelas oleh developer dan dipahami oleh semua stakeholder.

Dalam pembuatan dokukemen spesifikasi teknis, perlu menyetakan **make it clear, unambiguous, understandable, complete, consistent**

Contigency plan adalah istilah plan B Ketika ide / Solusi tidak berjalan mulus.

Lakukan riset User Requirement Specification dan Dokumen Spesifikasi Teknis Aplikasi dengan benar jangan sampai salah sasaran

**PERENCANAAN MODIFIKASI APLIKASII**

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau Bahasa ingrrisnya adalah Software Requirement Specification (SRS) adalah dokumen penting yang dibuat sebelum proses pengembangan software dimulai. Dokumen ini berfungsi untuk:

1. Menjelaskan sisi fungsionalitas (fitur) dan sisi non-fungsionalitas (keamanan, performa, kendala, dll)
2. Menjadi panduan bagi seluruh stakeholders agar memiliki konsep pemikiran yang sama

Isi dokumen SRS:

1. Pengantar (contoh berisi tujuan aplikasi dan sasaran pengguna)
2. Glosarium
3. Kebutuhan pengguna (contoh berisi deskripsi umum aplikasi dan kegunaan aplikasi bagi pengguna)
   1. Deskripsi umum menjelaskan kenapa fitur itu ada dan bagaimana cara kerjanya
4. Sistem arsitektur
5. Spesifikasi kebutuan system (contoh berisi fungsional, non-fungsional, antarmuka pengguna seperti front-end menggunakan html, css, dan javacscript. Lalu ada antarmuka perangkat keras seperti hp dan laptop yang dapat mengaksesnya)
   1. Penjabaran fungsional adalah terjemajan teknis dari deskripsi umum yang bersifat naratif
   2. Fungsional adalah “apa yang bisa dilakukan, seperti klik bar email dan passowd, klik button logout, dll” dan untuk non-fungsional adalah “bagaimana system melakukannya”
6. Model system (seperti flowchart. Model data: tabel produk, user, transaksi, ulasan, dll)
7. Rencana pengembangan system (rencana kedepannya setelah aplikasi berjalan. Seperti menambahkan fitur live yang sebelumnya tidak ada sebagai contohnya)
8. Apendiks

**Compiler:** Menerjemahkan semua code ke dalam file exe dan nantinya dieksekusi / dijalankan oleh CPU

**Interpreter:** Membaca 1 baris code, menerjemahkan 1 baris code, dan 1 baris code itu dieksekusi oleh CPU. Dan berlanjut ke baris ke dua hingga selesai

**Penyelesaian Masalah Menggunakan Cara Berpikir Komputasional**

1. **Decomposition / Breakdown**

Memecah masalah besar menjadi bagian bagian kecil. Seperti membuat aplikasi marketplace. Ini bisa dipecah menjadi:

 **Fitur Autentikasi** (login, register, logout)

 **Halaman Produk** (tampilkan, filter, cari)

 **Keranjang Belanja**

 **Pembayaran**

 **Profil Pengguna**

 **Dashboard Admin**, dll

1. **Pattern Recognition**

Memahami pattern dari masalah. Seperti menggambar 3 kucing dengan jenis berbeda, tapi kita tau bahwa itu semua adalah kucing hanya fiturnya saja yang beda. Contohnya seperti forloop, yang satu melooping data yang satu melooping angka. Keduanya sama sama loop dengan pattern yang sama tapi hanya fitur nya saja yang beda.

1. **Abstraksi**

Men-generalisasikan code. Kita tidak perlu tau bagaimana code itu bekerja, yang penting kita tau code itu fungsinya apa.

1. **Algorithm**

Bisa menggunakan pseudocode, flowchart, atau langsung di dalam code

1. **Evaluasi Menyeluruh**

Kita sudah deploy aplikasi, tapi ternyata user menemukan kendala

**Flowchart**

1. Start & End (Oval)

Simbol untuk mulai dan akhir

1. Proses (Rectangle)

Simbol untuk proses / langkah / aktivitas

1. Data (Paralellogram)

Simbol untuk menginput data dan output

1. Decision Making (Rhombus)

Yes / No

1. Database (Cylinder)
2. Document

**Dokumentasi Teknis**

Berikut manfaat yang akan didapatkan ketika mengimplementasikan dokumentasi teknis.

1. Pertama, dokumentasi teknis dapat meningkatkan retensi pengguna.
2. Lalu, dokumentasi teknis yang baik dapat menghemat waktu dan tenaga.
3. Dan yang terakhir adalah dokumentasi teknis berpotensi untuk meningkatkan penjualan produk Anda.