

VERSION 4.0
JANUARI, 2023



MANAJEMEN PROYEK PERANGKAT LUNAK

MODUL 1 – Software Development Life Cycle & Business Case

TIM PENYUSUN :
- ILYAS NURYASIN, S.KOM., M.KOM.
- DINDA ARINAWATI WIYONO
- MUHAMMAD NUR ICHSAN

PRESENTED BY : LAB. TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

MANAJEMEN PROYEK PERANGKAT LUNAK

CAPAIAN PEMBELAJARAN MATA KULIAH

1. Memahami metode SDLC (Software Development Life Cycle)
2. Memahami dan dapat membuat Business Case

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

1. Memahami tahapan/proses pengembangan perangkat lunak menggunakan metode SDLC
2. Memahami dalam analisa Business Case dari sebuah sistem informasi

KEBUTUHAN HARDWARE & SOFTWARE

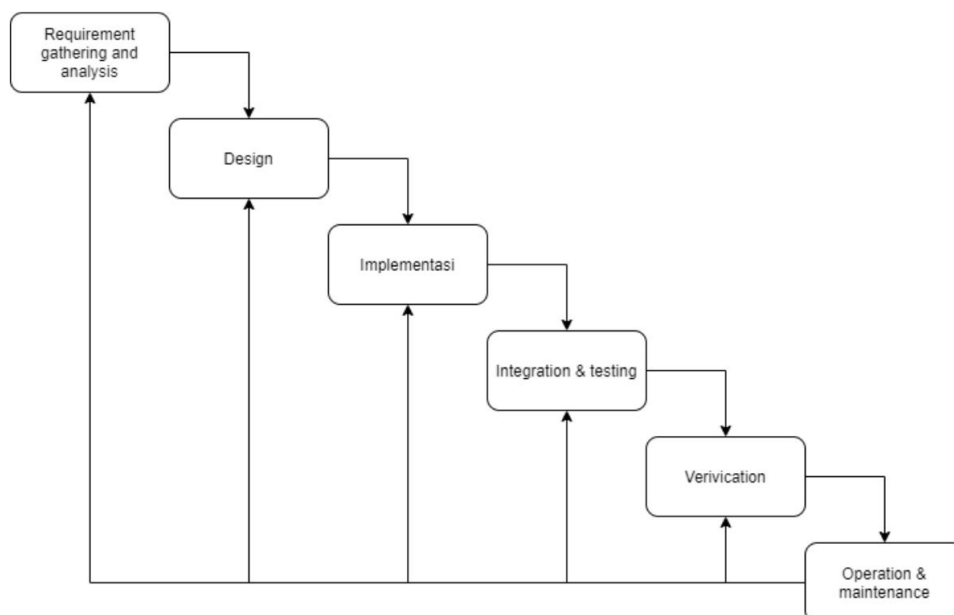
- Platform Presentasi (Power Point, dll.)
- Platform Penulisan (Word, dll.)

MATERI POKOK

Software Development Life Cycle

Metode SDLC (*Software Development Life Cycle*) adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem rekayasa perangkat lunak. Metode SDLC hadir untuk membantu dalam pengembangan produk. Metode ini memiliki banyak jenisnya, berikut adalah 4 contoh metode SDLC dalam pengembangan software.

1. Waterfall



Metode waterfall adalah metode kerja yang menekankan fase-fase yang berurutan dan sistematis. Disebut waterfall karena proses mengalir satu arah “ke bawah” seperti air terjun. Metode waterfall ini harus dilakukan secara berurutan sesuai dengan tahap yang ada. Berikut adalah tahap-tahap pengembangan dalam metode waterfall.

- **Requirement gathering and analysis**

Mengumpulkan kebutuhan secara lengkap untuk dianalisis dan mendefinisikan kebutuhan apa saja yang harus dicapai oleh program. Informasi dapat diperoleh melalui wawancara, diskusi, atau survey.

- **Design**

Melakukan perancangan desain perangkat lunak sebagai perkiraan sebelum dibuatnya kode. Desain sistem dapat dibuat menggunakan *Flowchart*, *Mind Map*, atau *Entity Relationship Diagram* (ERD).

- **Implementasi**

Implementasi ini adalah tahap dimana seluruh desain yang sebelumnya sudah dibuat diubah menjadi kode-kode program. Kode yang dihasilkan masih berbentuk modul-modul yang harus digabungkan di tahap selanjutnya.

- **Integration & testing**

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat sebelumnya dan melakukan pengujian untuk mengetahui apakah perangkat lunak yang dibuat telah sesuai dengan desain dan fungsinya atau tidak.

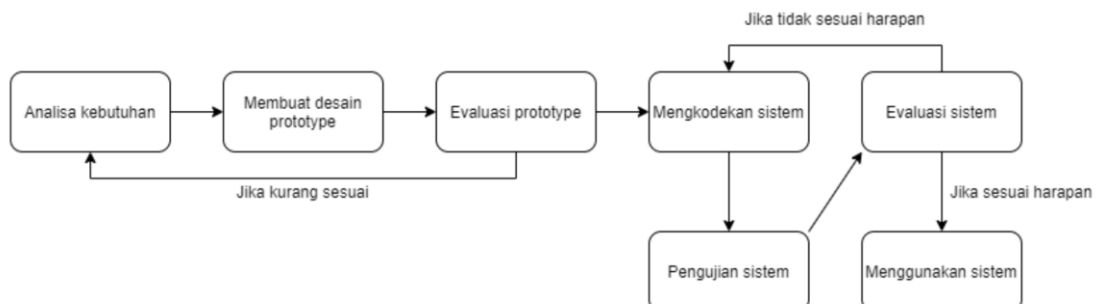
- **Verification**

Di tahap ini, pengguna atau klien yang langsung melakukan pengujian pada sistem, apakah sistem telah sesuai dengan yang disetujui atau belum sesuai.

- **Operation & maintenance**

Tahap ini merupakan tahap terakhir dari model waterfall. Sistem yang sudah selesai dijalankan serta dilakukan pemeliharaan. Pemeliharaan berupa memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

2. Prototype



Metode prototype adalah metode yang memungkinkan pengguna atau *user* memiliki gambaran awal tentang perangkat lunak yang akan dikembangkan, serta pengguna dapat melakukan pengujian di awal sebelum perangkat lunak dirilis.

Metode ini bertujuan untuk mengembangkan model menjadi perangkat lunak yang final. Artinya sistem akan dikembangkan lebih cepat dan biaya yang dikeluarkan lebih rendah. Metode prototype ini memiliki tahap-tahap yang harus dilakukan dalam pengembangan perangkat lunak. Berikut adalah tahap-tahap pengembangan perangkat lunak menggunakan metode prototype.

- **Analisa kebutuhan**

Pada tahap ini pengembang melakukan identifikasi perangkat lunak dan semua kebutuhan sistem yang akan dibuat.

- **Membuat prototype**

Membuat rancangan sementara yang berfokus pada alur program kepada pengguna.

- **Evaluasi prototype**

Evaluasi dilakukan untuk mengetahui apakah model prototype sudah sesuai dengan harapan.

- **Pengkodean sistem**

Jika prototype disetujui maka akan diterjemahkan ke dalam bahasa pemrograman yang sesuai.

- **Pengujian sistem**

Setelah perangkat lunak sudah siap, perangkat lunak harus melewati pengujian. Pengujian ini biasanya dilakukan dengan White Box Testing, Black Box Testing, dan lain-lain.

- **Evaluasi sistem**

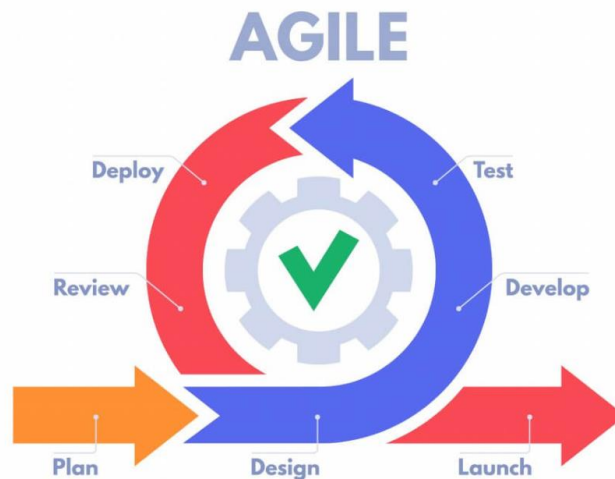
Pengguna melakukan evaluasi apakah perangkat lunak sudah sesuai dengan apa yang diharapkan atau tidak. Jika ya, lakukan tahap selanjutnya. Jika tidak, ulangi tahap mengkodekan sistem dan pengujian sistem.

- **Penggunaan sistem**

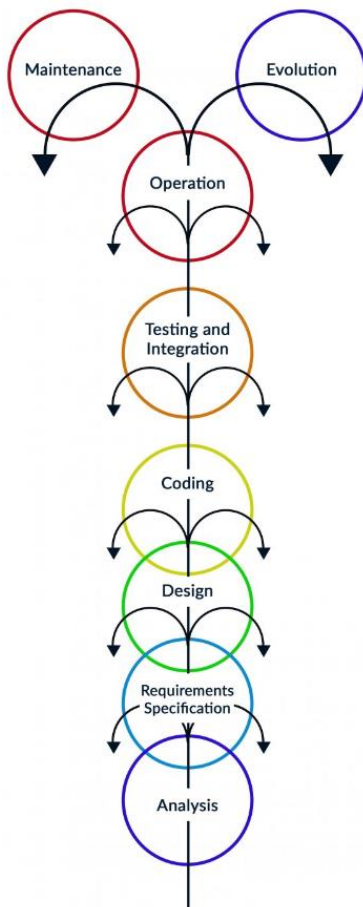
Perangkat lunak yang telah diuji dan disetujui siap untuk digunakan.

3. Agile

Metode agile adalah metode yang fleksibel di mana pengembangan dilakukan dalam jangka pendek. Namun diperlukan adaptasi yang cepat dari developer terhadap perubahan dalam bentuk apa pun. Berikut adalah alur tahapan dari agile software development.



4. Fountain



Metode fountain adalah perbaikan dari metode waterfall, di mana jenis tahapan masih sama. Namun beberapa jenis tahapan boleh didahulukan atau dilewati, tetapi ada tahapan yang tidak bisa dilewati, contohnya seperti memerlukan design sebelum melakukan implementasi, jika hal tersebut dilewati maka akan ada tumpang tindih. Berikut adalah tahap-tahap pengembangan perangkat lunak menggunakan metode fountain.

- **User requirement specification**

Mencari tahu apa saja yang dibutuhkan oleh pengguna dalam perangkat lunak yang sedang dikembangkan.

- **Software requirement specification**

Penyesuaian perangkat lunak dari sisi pengguna.

- **System design**

Pembuatan desain sistem yang akan dibuat sebelum diimplementasikan.

- **Program design**

Pembuatan desain yang lebih sempurna dan hampir mendekati hasil akhir dari perangkat lunak.

- **Implementation**

Di tahap ini dilakukan implementasi sesuai dengan desain yang sudah dibuat di tahap sebelumnya.

- **Program testing: unit**

Dalam tahap ini dilakukan uji coba terhadap unit-unit yang dibutuhkan dalam perangkat lunak yang dikembangkan.

- **Program testing: system**

Di tahap ini dilakukan uji coba terhadap sistem dari perangkat lunak seutuhnya sebelum perangkat lunak digunakan.

- **Program use**

Dalam tahap ini dilakukan pengajaran kepada pengguna untuk menggunakan perangkat lunak yang telah dibuat.

- **Software maintenance**

Biasanya dalam tahap ini dilakukan perawatan terhadap perangkat lunak yang sudah dibuat, perawatan dapat berupa update sistem atau perbaikan kesalahan atau *bugs* yang ada.

Business Case

Business case merupakan gambaran umum suatu proyek sebagai konsumsi dari high level manajemen untuk menentukan disetujui tidaknya proyek tersebut. Dokumen ini memuat organizational value, feasibility, biaya, manfaat, dan resiko dari proyek dan bukan merupakan anggaran atau perencanaan proyek.

Pada umumnya organizational value untuk suatu proyek berfokus pada meningkatkan efektivitas (atau produktivitas) dan mencapai efisiensi (atau menghemat biaya). Berikut merupakan contoh dari business case pada kasus pengembangan mobile application “Digital Parking UMM”.

1. Pernyataan Masalah

Tidak sedikit mahasiswa, dosen, beserta staf UMM yang menggunakan kendaraan pribadi terutama sepeda motor sebagai alat transportasi untuk ke kampus. Banyaknya jumlah pengendara motor ini membuat diperlukannya lahan parkir yang luas. Namun faktanya di lapangan adalah walau lahan parkir yang disediakan sudah cukup luas, tak jarang pengguna sepeda motor (terutama mahasiswa) yang merasa kesusahan untuk mendapat akses masuk dengan berbagai sebab diantaranya kebingungan mencari di mana kira2 tempat yang masih bisa menampung dikarenakan juga tidak diketahui kapasitas maksimal yang pasti.

Ketidak-efektifitasan ini menyebabkan waktu lebih banyak terbuang untuk mencari cari parkir. Oleh karena itu Digital Parking UMM diciptakan dengan tujuan mempermudah mahasiswa untuk mengakses informasi terkait area parkir yang masih tersedia.

2. Analisa Situasi Keadaan

Dengan aplikasi Digital Parking UMM dapat melihat ketersediaan tempat parkir yang ada di UMM. Dengan menggunakan aplikasi digital parking UMM mahasiswa, karyawan, dan dosen menjadi lebih mudah mendapatkan informasi terkait dengan ketersediaan tempat parkir.

3. Kebutuhan Bisnis

- Aplikasi akan menampilkan jumlah slot motor yang tersedia
- Aplikasi akan menampilkan area parkir mana yang buka dan tutup
- Aplikasi akan menampilkan prioritas tempat parkir untuk Mahasiswa, Karyawan, dan Dosen
- Mahasiswa, Karyawan, dan Dosen dapat melakukan scan barcode saat masuk ke area parkir

4. Solusi Masalah

Menyediakan informasi terkait dengan area parkir UMM seperti buka atau tidaknya area parkir, jumlah slot kendaraan yang tersedia, dan prioritas tempat parkir untuk mahasiswa, dosen, karyawan yang berguna untuk mempermudah mahasiswa untuk menentukan lokasi parkir kendaraan.

5. Konsistensi dengan misi organisasi

- Menampilkan jumlah slot motor yang tersedia secara akurat dan realtime
- Menampilkan area parkir mana yang buka dan tutup secara realtime
- Memberikan kenyamanan kepada orang-orang yang hendak menggunakan lahan parkir

6. Manfaat yang diharapkan

- Mempermudah mahasiswa, karyawan, dan dosen untuk menentukan lokasi parkir
- Dapat memberikan informasi terkait dengan ketersediaan parkir kepada mahasiswa

Note : tidak ada template pasti dalam pembuatan Business Case, dapat kalian kembangkan atau tambahkan sesuai dengan referensi yang kalian temukan.

REFERENSI

<https://www.dicoding.com/blog/metode-sdlc/>

<https://business.adobe.com/blog/basics/business-case>

LEMBAR KERJA

PRE-PRAKTIKUM

Buatlah kelompok beranggotakan 5 orang dengan role Manager, Initiator, Planner, Executor, dan Closing. Kelompok praktikum harus sama dengan kelompok yang ada di kelas mata kuliah.

Kegiatan 1

Pilih salah satu metode SDLC yang paling menarik dan dirasa paling cocok untuk digunakan dalam pengembangan software menurut satu kelompok. Jelaskan mengenai metode tersebut mulai dari tahapan hingga kelebihan dan kekurangan dari metode tersebut. Materi dapat ditambahkan dari luar modul dan referensi modul agar lebih lengkap dan jelas. Presentasikan kepada asisten secara bergantian oleh seluruh anggota kelompok.

Kegiatan 2

Tentukan ide untuk manajemen proyek perangkat lunak yang akan dibuat dan buat Business Case dari ide yang telah ditentukan oleh satu kelompok. Contoh : Sistem Informasi Kepegawaian.

Note : kelompok akan berlaku hingga Modul 6

RUBRIK PENILAIAN

Kegiatan 1 : 50%

Kegiatan 2 : 50%

****Penilaian juga berdasarkan individual, nilai bergantung dari penyampaian dan tanya jawab dengan asisten (nilai satu kelompok bisa ada yang berbeda).**