

VERSION 1.0  
JANUARI, 2024



# MANAJEMEN PROYEK PERANGKAT LUNAK

MODUL 2 – SDLC, COCOMO & Estimasi Bottom-Up

TIM PENYUSUN:  
ILYAS NURYASIN, S.KOM., M.KOM.  
NUR EVINA MAKNUN  
GILANG DWI DARMAWAN

PRESENTED BY : LABORATORIUM INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## MANAJEMEN PROYEK PERANGKAT LUNAK

---

### CAPAIAN PEMBELAJARAN MATA KULIAH

1. Memahami metode SDLC (Software Development Life Cycle)
  2. Memahami penghitungan COCOMO
  3. Memahami penghitungan estimasi Bottom-Up
- 

### SUB CAPAIAN PEMBELAJARAN MATA KULIAH

1. Memahami tahapan/proses pengembangan perangkat lunak menggunakan metode SDLC
  2. Dapat melakukan penghitungan dengan COCOMO dan estimasi Bottom-Up
- 

### KEBUTUHAN HARDWARE & SOFTWARE

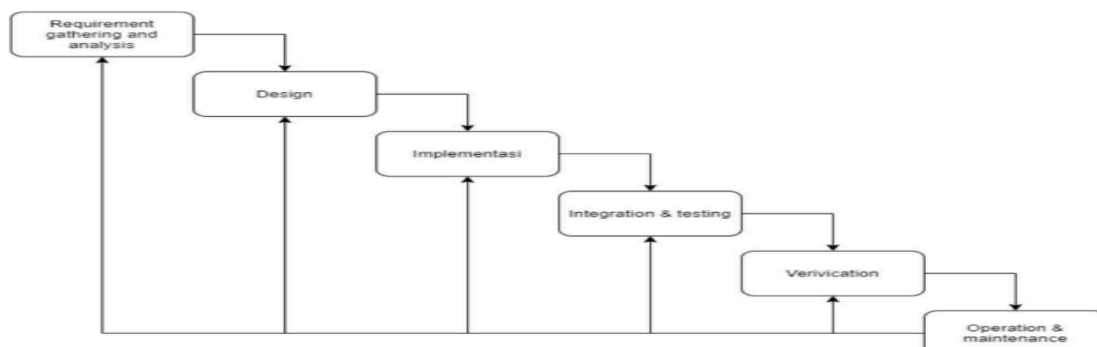
- Platform Presentasi (Power Point, dll.)
  - Platform Penulisan (Word, dll.)
- 

### MATERI POKOK

#### Software Development Life Cycle

Metode SDLC (Software Development Life Cycle) adalah proses pembuatan dan pengubahan 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.

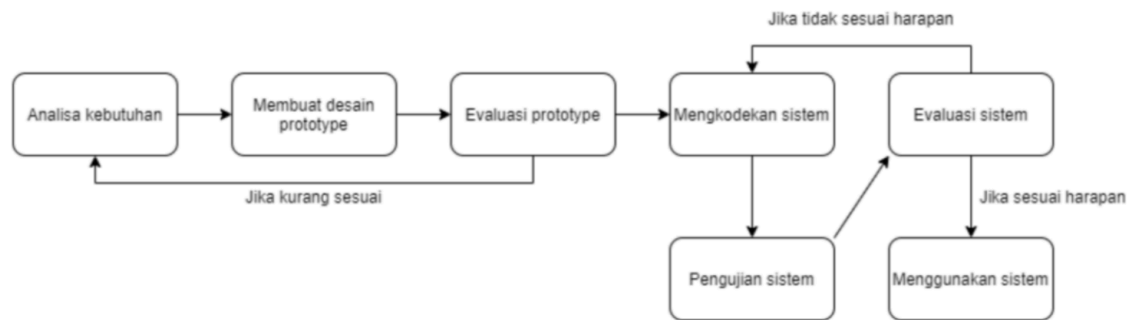
- **Verification**

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**

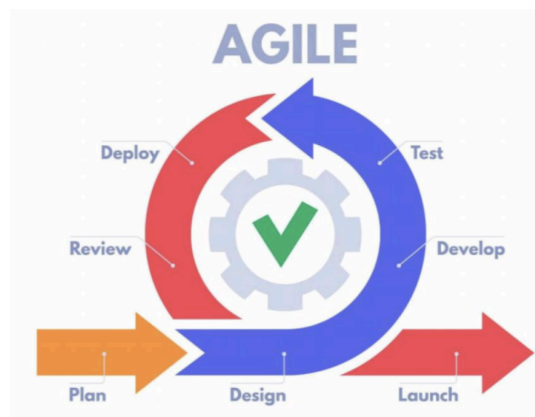
Pengguna melakukan evaluasi apakah perangkat lunak sudah sesuai dengan apa yang diharapkan atau tidak. Jika ya, lakukan tahap selanjutnya. Jika tidak, ulangi tahap mengkodekan sistem dan pengujian sistem.

- **Penggunaan sistem**

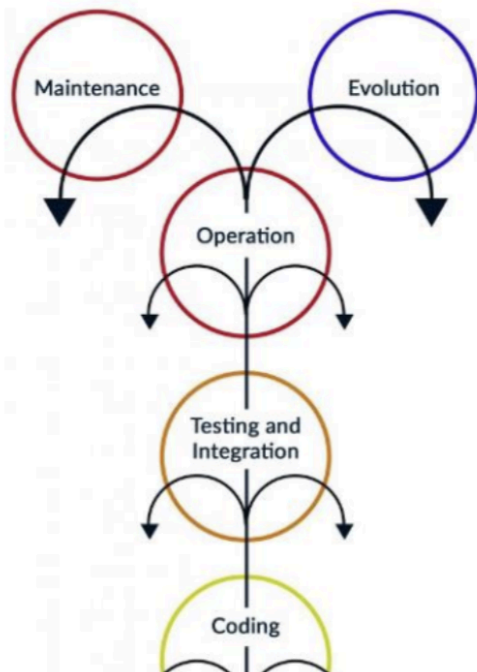
Perangkat lunak yang telah diuji dan disetujui siap untuk digunakan.

### 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.

## COCOMO

COCOMO (Constructive Cost Model) adalah model berdasarkan LOC, yaitu jumlah baris kode. Penghitungan COCOMO merupakan model perkiraan biaya prosedural untuk proyek perangkat lunak dan sering digunakan sebagai proses untuk memprediksi berbagai parameter terkait dengan pengembangan proyek seperti ukuran, usaha, biaya, waktu, dan kualitas. Model Cocomo diusulkan untuk memprediksi perkiraan biaya pada tingkat yang berbeda, berdasarkan jumlah akurasi dan realita yang diperlukan. Model ini dapat diterapkan pada berbagai proyek, yang karakteristiknya menentukan nilai konstanta yang akan digunakan dalam perhitungan selanjutnya. Karakteristik yang berkaitan dengan jenis sistem yang berbeda disebutkan di bawah ini. Definisi Boehm tentang sistem organic, semi-detached, and embedded:

- **Organic:** proyek perangkat lunak dengan ukuran relatif kecil, dengan anggota tim yang sudah berpengalaman, dan mampu bekerja pada permintaan yang relatif fleksibel
- **Semi-detached:** proyek yang memiliki ukuran dan tingkat kerumitan yang sedang, dan tiap anggota tim memiliki tingkat keahlian yang berbeda
- **Embedded:** proyek kompleks yang dibangun dengan spesifikasi dan operasi yang ketat

Berikut rumus perhitungan biaya pembuatan sistem menggunakan COCOMO :

$$E = a(KLOC)^b$$

$$T = c(E)^d$$

$$P = E / D$$

E adalah usaha (effort) orang per bulan, T adalah waktu (time) pengerjaan dalam satuan bulan, dan P adalah jumlah orang (person) yang diperlukan. Sedangkan untuk KLOC adalah estimasi jumlah baris kode dalam ribuan. Variabel a, b, c, dan d tabel koefisiennya dapat dilihat

pada tabel dibawah ini:

Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Untuk memperjelas penjelasan sebelumnya, berikut ini contoh penghitungan estimasi biaya menggunakan COCOMO.

<p><b>Organik :</b></p> $E = a(Kloc)^b$ <p>→ 2,4 (4)<sup>1,05</sup></p> <p>→ 10,28 mm</p> $D = c(E)^d$ <p>→ 2,5 (10,28)<sup>0,38</sup></p> <p>→ 6,06 bulan</p> $P = E/D$ <p>→ 1,69 bulan</p> <p><i>Digenapkan ke dalam :</i></p> <p><i>3 bulan dengan jumlah pekerja 3 orang</i></p>	<p><b>Semi Detached :</b></p> $E = a(Kloc)^b$ <p>→ 3,0 (4)<sup>1,12</sup></p> <p>→ 14,17 mm</p> $D = c(E)^d$ <p>→ 2,5 (14,17)<sup>0,35</sup></p> <p>→ 6,32 bulan</p> $P = E/D$ <p>→ 2,24 orang</p> <p><i>Digenapkan ke dalam :</i></p> <p><i>3 bulan dengan jumlah pekerja 4 orang</i></p>	<p><b>Embedded :</b></p> $E = a(Kloc)^b$ <p>→ 3,6 (4)<sup>1,20</sup></p> <p>→ 19 mm</p> $D = c(E)^d$ <p>→ 2,5 (19)<sup>0,32</sup></p> <p>→ 6,4 bulan</p> $P = E/D$ <p>→ 2,96 orang</p> <p><i>Digenapkan ke dalam :</i></p> <p><i>3 bulan dengan jumlah pekerja 6 orang</i></p>
--	--	--

Berdasarkan nilai yang didapatkan pada penghitungan di atas, maka dapat ditentukan estimasi biaya yang dibutuhkan seperti di bawah ini:



- **Organik :**  
 $(\text{baris kode} * \text{harga perbaris}) + (\text{jumlah karyawan} * \text{gaji perbulan}) * \text{lama pengerjaan} + \text{keuntungan \%}$   
 $= (4.000 * 5.000) + (3 * 3.000.000) * 3\text{bulan} + 15\%$   
 $= 20.000.000 + 9.000.000 * 3\text{bulan} + 15\%$   
 $= 20.000.000 + 31.050.000$   
 $= \text{Rp. } 51.050.000$
- **Semi Detached :**  
 $(\text{baris kode} * \text{harga perbaris}) + (\text{jumlah karyawan} * \text{gaji perbulan}) * \text{lama pengerjaan} + \text{keuntungan \%}$   
 $= (4.000 * 5.000) + (4 * 3.000.000) * 3\text{bulan} + 15\%$   
 $= 20.000.000 + 12.000.000 * 3\text{bulan} + 15\%$   
 $= 20.000.000 + 41.400.000$   
 $= \text{Rp. } 61.400.000$
- **Embedded :**  
 $(\text{baris kode} * \text{harga perbaris}) + (\text{jumlah karyawan} * \text{gaji perbulan}) * \text{lama pengerjaan} + \text{keuntungan \%}$   
 $= (4.000 * 5.000) + (6 * 3.000.000) * 3\text{bulan} + 15\%$   
 $= 20.000.000 + 18.000.000 * 3\text{bulan} + 15\%$   
 $= 20.000.000 + 62.100.000$   
 $= \text{Rp. } 82.100.000$

---

## REFERENSI

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

---

## LEMBAR KERJA

### PRE-PRAKTIKUM

1. Pilih salah satu metode SDLC yang paling menarik dan dirasa paling cocok untuk digunakan dalam pengembangan software menurut satu kelompok.
2. Hitung estimasi biaya produksi dari proyek yang kelompok kembangkan menggunakan penghitungan COCOMO (organic, semi-detached, embedded).
3. Diskusikan kemudian sampaikan progress kepada asisten di pekan **latihan**.

### PRAKTIKUM

Buat dokumen yang berisi:

1. Penjelasan alasan tentang SDLC yang dipilih, kekurangan dan kelebihan SDLC yang dipilih.
2. Estimasi biaya produksi menggunakan perhitungan COCOMO (Organic, Semi-detached, Embedded).
3. Presentasikan dokumen sesuai masing-masing role kepada asisten pada pekan **demo**.

**Note : kelompok dan business case akan berlaku hingga Modul 6**

---

### RUBRIK PENILAIAN

Pre-praktikum (Progress memilih SDLC dan menghitung estimasi)	20%
Praktikum (Presentasi dokumen penjelasan SDLC, perhitungan estimasi, lengkap sesuai arahan modul)	80%
Total	100%

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