

## BAB II

# COMPUTER SYSTEM ENGINEERING

Computer system engineering (Rekayasa Sistem Komputer) terdiri atas 2 bagian, yaitu :

- 📁 Hardware engineering
- 📁 Software engineering

Setiap disiplin ini berusaha menunjukkan pengembangan sistem berbasis komputer teknik engineering. Untuk hardware komputer telah sedemikian maju dan relatif jenuh. Sebaliknya software komputer mulai berkembang, dan saat ini menggantikan peranan hardware sebagai elemen sistem yang sulit direncanakan, sedikit kemungkinan untuk berhasil dengan biaya rendah dan waktu yang cepat, serta paling sukar untuk dikelola.

*Apa Sistem itu ?*

Sistem adalah sekumpulan elemen yang saling berinteraksi untuk mencapai suatu tujuan. Sedangkan **Computer Based System** diorganisir untuk mendapatkan beberapa metode, prosedur atau pengontrolan dengan cara mengelola informasi.

Elemen-elemen dari sistem berbasis komputer adalah :

1. Software

Program komputer, struktur data dan dokumentasi yang saling berhubungan dan memberikan efek pada metode, prosedur dan kontrol yang diinginkan.

2. Hardware

Peralatan elektronik, (misalnya CPU, memory) yang memberikan kemampuan komputasi serta peralatan elektromedia (misalnya sensor, motor, pompa) yang memberikan fungsi external.

3. People / Brainware

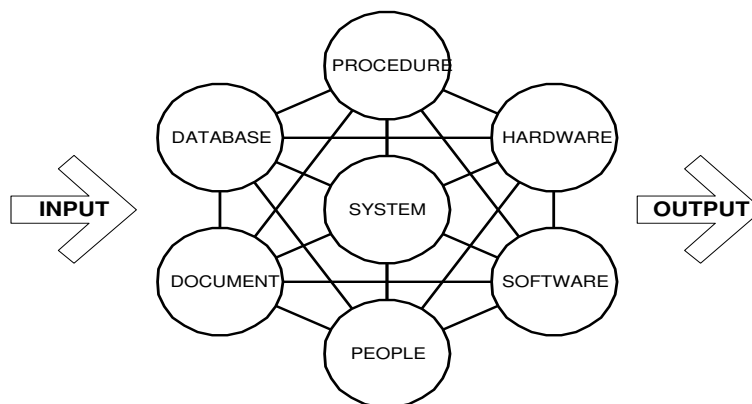
User dan operator dari hardware dan software

4. Database

Sekumpulan informasi yang besar, yang diorganisir agar dapat diakses oleh software dan merupakan bagian integral dari fungsi sistem.

5. Prosedur

Langkah-langkah yang menetapkan pemakaian khusus untuk setiap elemen sistem.



Keterangan :

Computer system engineering adalah suatu aktifitas pemecahan masalah fungsi sistem yang diinginkan, ditemukan, dianalisis, dan dialokasikan ke elemen-elemen sistem individu.

Computer System Engineering disebut juga *Sistem Analisis*, dimulai dengan :

1. Penetapan tujuan customer
2. Hambatan-hambatan dan representasi fungsi performance yang dapat dialokasikan ke masing-masing elemen sistem.

Segera setelah fungsi performance, hambatan dan interface ditetapkan, system engineering selanjutnya melakukan pekerjaan alokasi. Selama pengalokasian fungsi diserahkan kepada satu / lebih elemen sistem (misalnya software, hardware, people, dll) seringkali alokasi alternatif diusulkan dan dievaluasi.

Fungsi yang dialokasikan maksudnya adalah menentukan mana yang masuk ke hardware, ke software dan ke brainware

Berikut ini adalah kriteria pemilihan konfigurasi sistem berdasarkan alokasi fungsi dan performance ke elemen sistem :

1. Project Consideration - Pertimbangan Proyek
  - ☞ Dapatkah konfigurasi dihasilkan dengan biaya dan jadwal yang ditetapkan lebih awal ?
2. Business Consideration - Pertimbangan Bisnis
  - ☞ Dapatkah konfigurasi memberikan solusi yang paling menguntungkan ?
  - ☞ Dapatkah dipasarkan dengan sukses ?
  - ↳ Pertimbangan ini yang paling penting.
3. Technical Consideration - Pertimbangan tehnik
  - ☞ Apakah ada teknologi untuk mengembangkan semua elemen sistem ?
  - ☞ Dapatkah fungsi performance dijamin ?
  - ☞ Dapatkah konfigurasi dipelihara dengan cukup baik ?
4. Manufacturing Evaluation - Evaluasi Pabrikasi
  - ☞ Apakah fasilitas dan peralatan manufaktur tersedia ?
  - ☞ Apakah ada komponen yang diperlukan dengan segera ?
  - ☞ Apakah jaminan kualitas dapat dipercaya ?
5. Human Issues - Hal-hal yang berhubungan dengan manusia
  - ☞ Apakah tenaga kerja terlatih untuk pengembangan dan manufaktur tersedia ?
  - ☞ Apakah customer mengerti dengan apa yang akan dicapai oleh sistem ?
6. Environmental Interface - Berhubungan dengan lingkungan
  - ☞ Apakah konfigurasi yang diusulkan sudah cukup berhubungan dengan lingkungan external dari sistem ?
  - ☞ Apakah komunikasi mesin → manusia dan manusia → mesin sudah ditangani dengan baik ?
7. Legal Consideration - Pertimbangan hukum
  - ☞ Apakah pertimbangan yang dihasilkan sudah dilindungi oleh hukum ?

## **PERTIMBANGAN HARDWARE**

Computer System Engineering selalu mengalokasikan satu / lebih fungsi sistem ke hardware komputer.

*Elemen-elemen hardware*

1. CPU - Cenral Processing Units

2. Adalah unit yang melakukan pekerjaan aritmatik, logika, dan fungsi pengontrol serta berinteraksi dengan komponen lainnya. Sekarang ini, beberapa arsitektur komputer ditambahkan ko-prosesor untuk melakukan fungsi pengolahan khusus ( fungsi kalkulasi ) sehingga performance CPU dapat ditingkatkan.
3. BUS
4. Adalah alat komunikasi yang menghubungkan elemen satu dengan elemen lainnya untuk pengiriman instruksi, data dan informasi pengontrolan.
5. Memory
6. Memory memberikan tempat penyimpanan instruksi dan data yang dapat diakses langsung / tidak langsung melalui perintah yang dieksekusi oleh CPU dan ko-prosesornya.

Memory terbagi menjadi 2 bagian, yaitu :

A. Memori Primer / Primary Memory / Main Memory

Adalah memory yang terdapat di dalam komputer, terdiri atas 2 bagian yaitu :

i. RAM - Random Access Memory

Untuk menyimpan data / instruksi yang bersifat temporary

ii. ROM - Read Only Memory / Firmware

Untuk menyimpan perintah dan / atau data yang permanen.

ROM terbagi atas 2 golongan

a. PROM - Programmabel Read Only Memory

Memory ROM yang dapat ditulis / diprogram dan dapat dihapus dengan cara :

📁 EEPROM - Erasable Electrical Programmabel ROM

Dihapus dengan kejutan listrik tertentu

📁 UVROM - Ultra Violet Programmabel ROM

Dihapus dengan sinar ultra violet

b. MASK ROM

ROM yang terjual sudah diprogram pada saat dibuat oleh pabriknya.

B. Memory Sekunder

Sifat yang menonjol dari memory jenis ini adalah :

- i. Waktu akses lambat
- ii. Kapasitas besar sekali dibandingkan dengan memory primer
- iii. Waktu akses berkisar milidetik dengan kapasitas antara 400.000 sampai 1 billion byte
- iv. Contoh : Floppy disk, harddisk, hardcard, optical disk

## **APLIKASI HARDWARE**

Dapat dikelompokkan dalam 3 bagian besar, yaitu :

1. Pengelolaan informasi
2. Pengontrolan proses dan aplikasi real time
3. Tambahan intelegensi

## **REKAYASA HARDWARE**

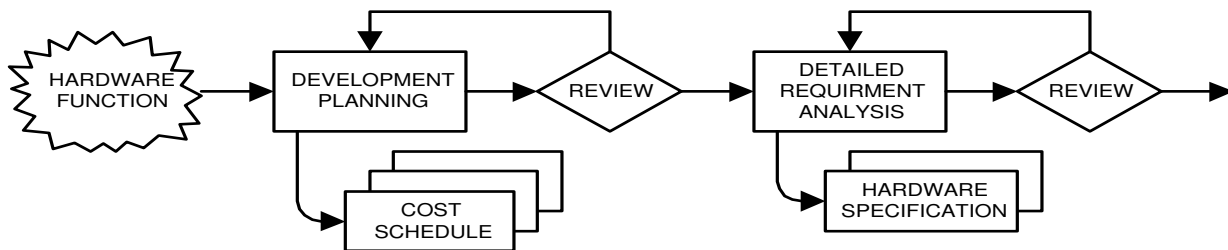
Untuk komputer digital yang dikembangkan dari perancangan elektronik, proses perancangannya terdiri dari 3 tahap :

1. Perencanaan dan spesifikasi

2. Perencanaan dan implementasi prototype
3. Manufaktur distribusi dan pelayanan

Sejara / sesudah analisis dan definisi dijalankan, fungsi dialokasikan ke hardware.

Fase I : Perencanaan dan Spesifikasi



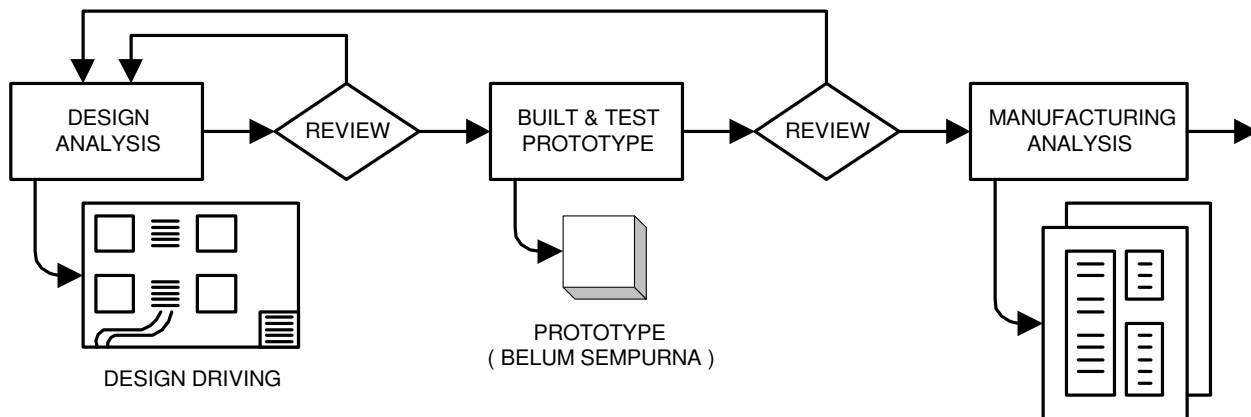
Fase I terdiri dari :

1. Perencanaan pengembangan
2. Analisis hardware

Perencanaan pengembangan dilaksanakan untuk menetapkan lingkup-lingkup dari usaha-usaha terhadap hardware, oleh karena itu menimbulkan beberapa pertanyaan, antara lain :

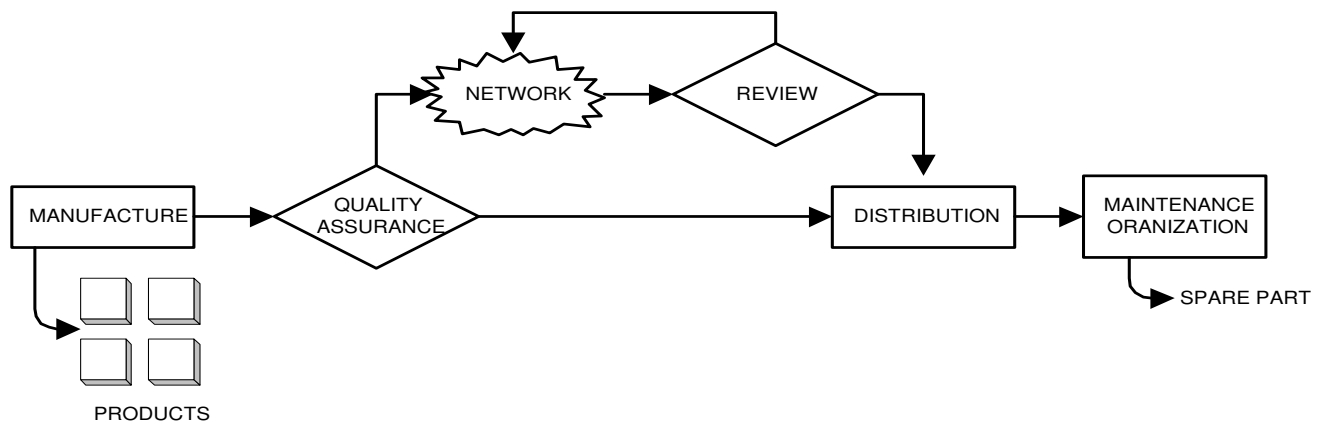
1. Jenis hardware apa yang terbaik untuk fungsi yang ditentukan?
2. Hardware yang mana yang tersedia untuk dijual, bagaimana biayanya, jenis interface yang diperlukan, dan apa yang harus dilakukan untuk merancang dan membangun ?

Fase II : Perencanaan dan Implementasi Prototype



Kebutuhan analisis dan konfigurasi hardware mulai dirancang, dilakukan tinjauan teknis demi mendapatkan spesifikasi rancangan yang benar. Komponen mulai dibuat dan prototype mulai diralat. Prototype diuji untuk menjamin bahwa prototype telah memenuhi semua persyaratan. Namun prototype sering menghadapi ketidakmiripan dengan prosedur yang dibuat. Karena itu perlu adanya spesifikasi pabrikan

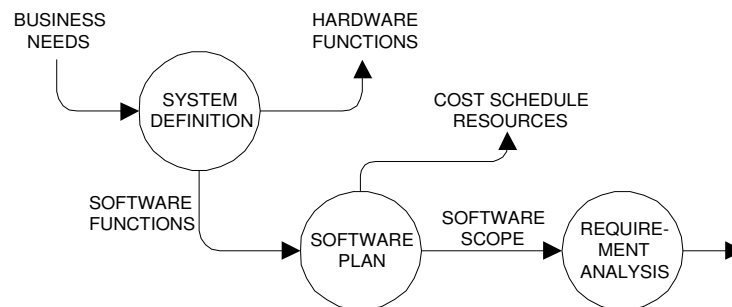
Fase III : Manufacture Distribution dan Pelayanan



Mulai dihasilkan prosedur-prosedur dengan penekanan pada kualitas produk. Dengan mekanisme distribusi produk terhadap fase ini, juga dibentuk bagian perbaikan dan maintenance

## PERENCANAAN SISTEM

Tahap perencanaan dari siklus hidup software adalah suatu proses definisi, analisis, spesifikasi, estimasi dan review.



Definisi sistem merupakan langkah pertama dalam fase perencanaan.

Tujuan dari definisi sistem ini adalah :

1. Evaluasi konsep sistem : feasibility, cost benefit, dan business needs
2. Jelaskan interface, function, dan performance sistem
3. Alokasi fungsi pada hardware, software dan elemen tambahan.

Tujuan dari perencanaan software adalah mengestimasi biaya dan waktu pengembangan. Untuk mencapai ini, lingkup software harus dimengerti dengan sempurna, dan sumber harus ditentukan dengan tepat.

Analisis kebutuhan software memperjelas :

1. Software interfaces
2. Atribut fungsional
3. Karakteristik performance
4. Kendala desain
5. Kriteria validasi

Timbul pertanyaan :

1. Berapa besar usaha yang akan diberikan pada fase perencanaan ?

- ↳ 10 s/d 20 % dari usaha keseluruhan proyek diberikan pada perencanaan dan analisis kebutuhan software.
- 2. Siapa yang mengerjakannya ?
- ↳ Analis yang berpengalaman dan terlatih memperkerjakan hampir semua pekerjaan yang berhubungan dengan fase perencanaan. Untuk proyek yang sangat besar, dapat dibentuk sebuah tim analis.
- 3. Mengapa begitu sulit ?
- ↳ Konsep yang tidak jelas harus ditransformasikan ke dalam elemen yang jelas.

## FEASIBILITY STUDI ( STUDI KELAYAKAN )

Semua proyek layak bila sumber dan waktunya tidak terbatas. Kenyataannya, pengembangan sistem berbasis komputer dibatasi oleh sumber dan waktu.

Ada 4 bidang utama yang menjadi konsentrasi dari feasibility studi, yaitu :

### 1. *Economic Feasibility* :

Evaluasi biaya (cost) dan manfaat (benefit) dalam pengembangan sistem.

### 2. *Tehcnical feasibilitu* :

Studi tentang fungsi, performance, dan hambatan yang berpengaruh terhadap kemampuan mendapatkan sistem yang baik.

### 3. *Legal Feasibility* :

Penentuan berbagai pelanggaran, kewajiban yang dapat terjadi dari pengembangan sistem.

### 4. *Alternative* :

Evaluasi sebagai alternatif untuk mengembangkan sistem

## ANALISIS COST BENEFIT

Menggambarkan biaya pengembangan proyek dan mempertimbangkan keuntungan sistem, baik yang *tangible* maupun *intangibile* (dapat diukur dan tidak dapat diukur).

Analisis cost benefit ini tergantung dari karakteristik sistem yang akan dikembangkan, ukuran relatif proyek (besar kecil proyek), dan ROI (Return On Investment) yang diharapkan dari proyek.

Keuntungan dari sistem baru selalu dibandingkan dengan keuntungan dari sistem yang ada.

### Contoh soal :

Suatu CAD (Computer Aided Design) akan menggantikan cara manual dalam membuat gambar-gambar teknik. Misalkan :

↳ t = waktu rata-rata menggambar = 4 jam

↳ c = biaya gambar perjam = \$20

↳ n = banyaknya gambar pertahun = 8000

↳ p = persentase gambar yang dihasilkan dengan sistem CAD = 60%

Penurunan waktu gambar menjadi  $\frac{1}{4}$  nya dengan adanya sistem CAD.

Jadi :

Biaya yang dapat ditekan (dihemat) sebesar :

$$= \frac{1}{4} \times t \times c \times n \times p = \frac{1}{4} \times 4 \times 20 \times 8000 \times 60\% = 96,000/\text{thn}$$

Bila untuk sistem CAD harus dikeluarkan biaya sebesar :

📁 biaya pengembangan / membeli = \$204,000,

📁 biaya pemeliharaan = \$32,000 per tahun,

maka masa pengembalian / payback periode dari proyek CAD ini adalah :

Biaya pengeluaran (cost) = biaya penghematan (benefit)

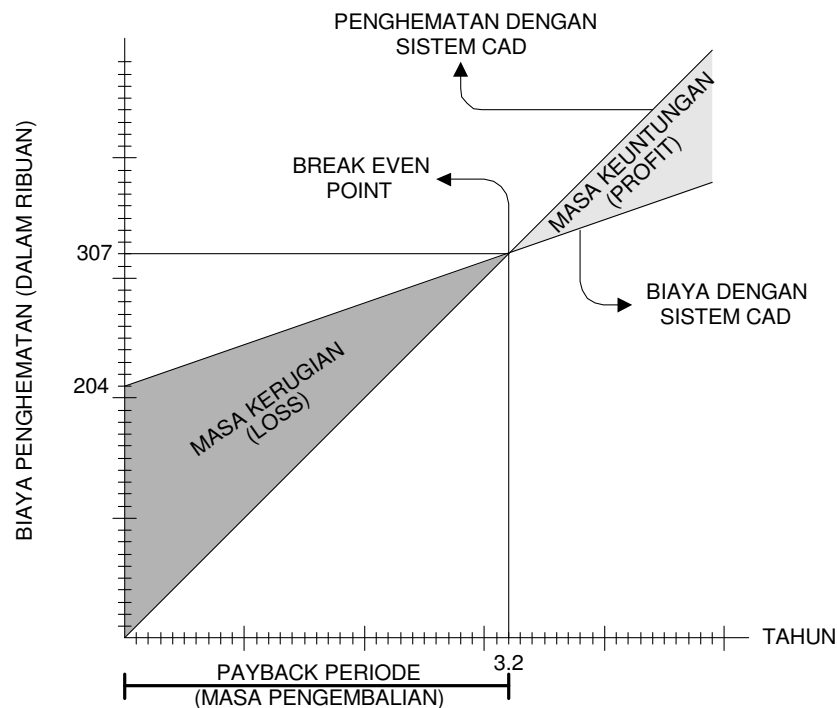
$$204,000 + x \cdot 32,000 = 96,000 \cdot x$$

$$64,000 \cdot x = 204,000$$

$$x = 3.2 \text{ tahun}$$

Ini berarti setelah 3.2 tahun, barulah proyek CAD ini memberikan titik impas (break even). Setelah ini barulah memberikan keuntungan.

Grafik :



## PERENCANAAN SOFTWARE

Untuk melaksanakan pengembangan proyek software dan berhasil, kita harus mengerti :

1. Ruang lingkup pekerjaan yang dilakukan
2. Sumber yang diinginkan
3. Usaha dan biaya
4. Jadwal yang dikehendaki

Perencanaan software adalah :

Langkah kedua dalam fase perencanaan, tetapi merupakan langkah pertama dalam proses rekayasa software yang akan memberikan pengertian kepada 4 hal di atas.

Perencanaan software mengkombinasikan 2 tugas, yaitu :

1. Riset
2. Estimasi

Tujuan perencanaan software :

Memberikan suatu kerangka yang memungkinkan manajer membuat estimasi yang beralasan tentang sumber, biaya dan jadwal.

Contoh soal :

TUGAS	USAHA, per orang per bulan
Requirement	1.5
Design	3.0
Code	1.0
Test	3.5
Usaha yang dihasilkan	9.0

Dari usaha ini, dihasilkan 2,900 LOC (banyaknya baris program). 200 LOC digunakan simulasi, dan testing tidak termasuk bagian dari software yang dioperasikan.

$$\begin{aligned} \text{Produktivitasnya :} &= \frac{2700 \text{ LOC}}{9.0 \text{ per orang per bulan}} \\ &= 300 \text{ LOC per orang per bulan.} \end{aligned}$$

Ada 4 orang software engineering yang masing-masing mampu menghasilkan 4000 LOC per tahun.

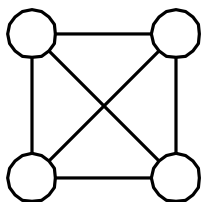
Bila mereka dipekerjakan dalam 1 team, maka proyek ada 6 jalur komunikasi yang mungkin (communication path). Setiap jalur komunikasi memerlukan waktu yang seharusnya digunakan untuk pengembangan Loding sebesar 240 LOC per tahun.

Bila proyek 1 tahun tersebut mengalami keterlambatan jadwal dan tinggal 1 bulan lagi, perlu penambahan 3 orang lagi kedalam team tersebut.

Bila dianggap terjadi pengurangan produktivitas team, untuk setiap jalur komunikasi adalah sama, baik untuk pegawai lama dan baru.

Hitung berapa produktivitas team sebelum dan sesudah penambahan 3 orang tersebut, sekaligus jangka waktunya berbeda !!!

Jawab :



Jalur komunikasi 4 orang → ada 6 jalur (lihat gambar).

$$\text{Produktivitas sebelum penambahan : } = ( 4 \times 4,000 ) - ( 240 \times 6 ) = 16,000 - 1,440 = 14,560$$

LOC per tahun

Produktivitas setelah penambahan :

📁 Jika jalur komunikasinya berbeda :

$$\begin{aligned} &= (4 \times 4000) + (3 \times \frac{4000}{12} \times 2) - (6 \times 240 + 15 \times \frac{240}{12} \times 2) \\ &= 16000 + 2000 - ( 1440 + 600 ) \end{aligned}$$



$$= 18000 - 2040$$

$$= 16960$$

📁 Jika jalur komunikasinya dianggap sama :

$$= (4 \times 4000) + (3 \times \frac{4000}{12} \times 2) - (6 \times 240 + 15)$$

$$= 16000 + 2000 - (1440 + 3600)$$

$$= 18000 - 5040$$

$$= 12960$$

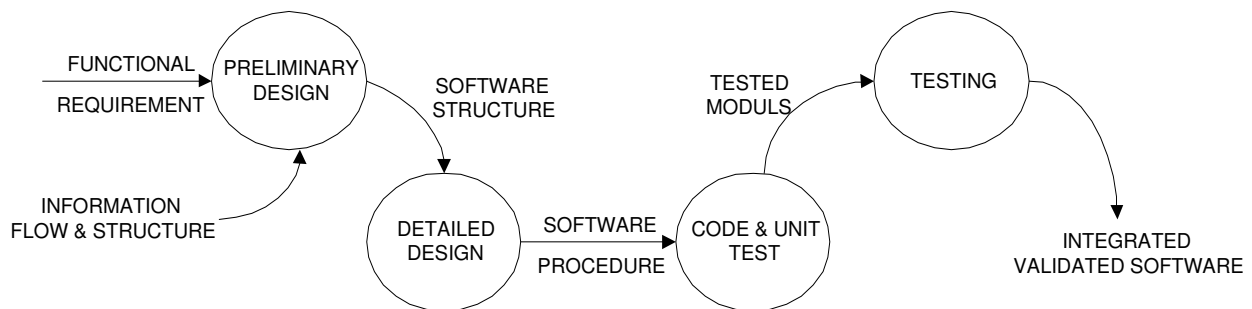
## PROSES DESAIN SOFTWARE

Desain dalam fase pengembangan merupakan langkah pertama, di mana fase pengembangan merupakan fase kedua dalam siklus hidup software.

Segera sesudah kebutuhan software ditetapkan, dimulailah fase pengembangan yang terdiri dari 4 langkah berbeda :

1. Desain awal (preliminary design)
2. Detailed Design (Desain primeir)
3. Coding
4. Testing

Aliran informasi selama fase pengembangan dapat dilihat pada gambar di bawah ini.



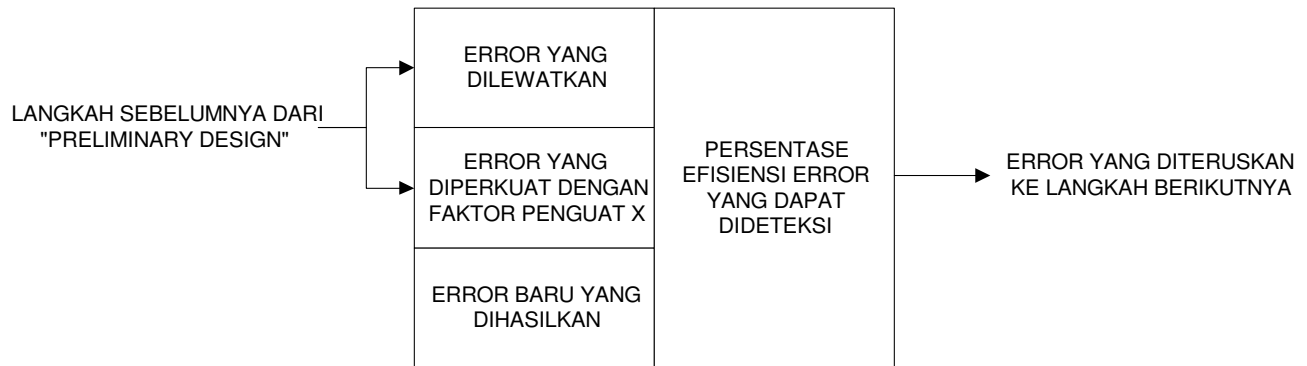
Kebutuhan software & aliran informasi ( Structure Information Software ) memulai langkah desain awal dengan menggunakan 1 dari beberapa metode desain struktur software untuk dikembangkan. Struktur software yang juga disebut Arsitektur Software ini menjelaskan tentang hubungan antar elemen utama dari sebuah program. Desain terinci menterjemahkan elemen-elemen struktural ke dalam penjelasan software ( prosedur software ).

Source Code dihasilkan dan pengetesan awal dilakukan selama langkah pengkodean dan pengetesan unit hasilnya berupa model-model program yang sudah ditest.

Langkah terakhir dalam fase pengembangan adalah dilakukannya pengetesan integrasi dan validasi.

Fase pengembangan paling sedikit menyerap 75% dari biaya software baru. Ini berarti pengambilan keputusan dalam fase ini akan sangat mempengaruhi keberhasilan dalam implementasi software dan kemudahan dalam pemeliharaan software.

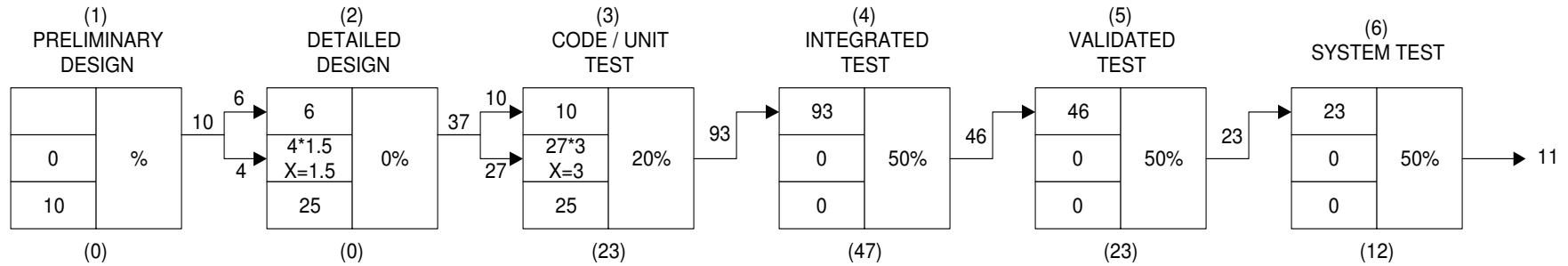
## DEFECT APPLICATION MODEL ( DAM )



DAM digunakan untuk memberikan gambaran tentang pembentukan dan pendeteksian error selama desain awal dari Desain Terinci dan Pengkodean.

Dengan model ini, kita dapat membandingkan besarnya biaya yang dikeluarkan dengan adanya error, baik untuk review maupun tanpa review.

Contoh DAM tanpa REVIEW :



Cara Kerja :

2.  $6 + (4 \times 1,5) + 25 = 37$

$37 \times 0\% = 0$

$37 - 0 = 37$

3.  $40 + (27 \times 3) + 25 = 116$

$116 \times 20\% = 2320\% = 23,2 \approx 23$

$116 - 23 = 93$

4.  $93 + 0 + 0$

$93 \times 50\% = 4650\% = 46,5 \approx 46$

$93 - 46 = 47$

5.  $46 + 0 + 0 = 46$

$46 \times 50\% = 2300\% = 23$

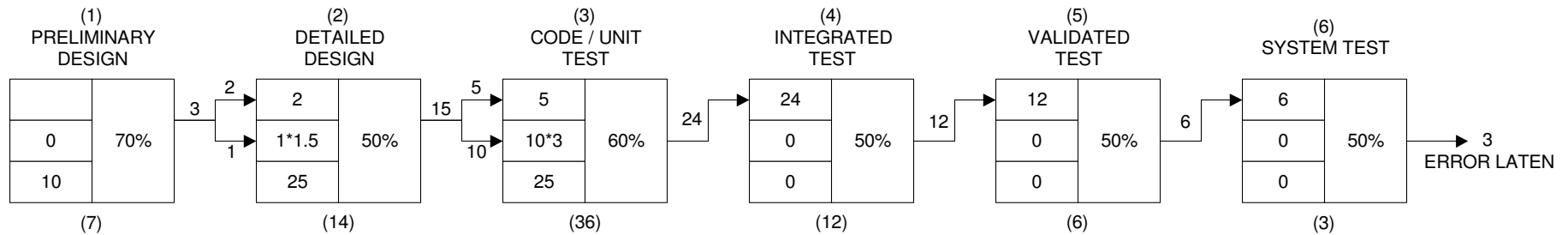
$46 - 23 = 23$

6.  $23 + 0 + 0 = 23$

$23 \times 50\% = 1150\% = 11,5 \approx 11$

$23 - 11 = 12$

Contoh DAM dengan REVIEW :



$$1. (10 + 0) \times 70\% = 7$$

$$2. 5 + (1 \times 1,5) + 25 = 28,5$$

$$28,5 \times 50\% = 1425\% = 14,25 \approx 14$$

$$28,5 - 14 = 14,5 \approx 15$$

$$3. (10 \times 3) + 5 + 25 = 60$$

$$60 \times 60\% = 3600\% = 26$$

$$60 - 36 = 24$$

$$4. 0 + 0 + 24 = 24$$

$$24 \times 50\% = 1200\% = 12$$

$$24 - 12 = 12$$

$$5. 12 + 0 + 0 = 12$$

$$12 \times 50\% = 600\% = 6$$

$$12 - 6 = 6$$

$$6. 6 + 0 + 0 = 6$$

$$6 \times 50\% = 3$$

$$6 - 3 = 3$$

KEGIATAN	BIAYA per ERROR	NON REVIEW	REVIEW
Selama desain	1	0 x 1 = 0	21 x 1 = 21
Sebelum test ( coding )	5	23 x 5 = 115	36 x 5 = 180
Selama test ( validated & integrated test )	10	82 x 10 = 820	21 x 10 = 210
Setelah dipasarkan ( error laten )	100	11 x 100 = 1100	3 x 100 = 300
Total biaya		2035	711

Soal Latihan no. 1 :

Diketahui :

5 software engineering masing-masing mampu menyelesaikan 4000 LOC per tahun bila bekerja secara individu. Mereka bekerja sama dalam 1 team. Bila proyek 1 tahun tersebut mengalami keterlambatan dan tinggal 1 bulan lagi, perlu tambahan 1 software engineering lagi ke dalam team tersebut. Pengurangan produktivitas team untuk setiap jalur komunikasi adalah sama ( 200 LOC per tahun ) untuk menunjuk adanya proses belajar bagi staff baru.

Ditanya :

Hitung produktivitas team sebelum dan sesudah penambahan seorang software engineering !!!

*Jawab :*

↳ Sebelum :

$$\begin{aligned} &= ( 4000 \times 5 ) - ( 10 \times 200 ) \\ &= 20000 - 2000 = 18000 \text{ LOC / tahun} \end{aligned}$$

↳ Sesudah

↳ Jalur sama :

$$\begin{aligned} &= \left\{ (5 \times 4000 + 1 \times \frac{4000}{12} \times 1) - (10 \times 200 + 5 \times 200) \right\} \\ &= \{ (20000 + 333,3) - 3000 \} \\ &= 20333,3 - 3000 = 17333,3 \text{ LOC / tahun} \end{aligned}$$

↳ Jalur beda :

$$\begin{aligned} &= 20333,3 - (2000 + 5 \times \frac{200}{12} \times 1) \\ &= 20333,3 - 2083,3 = 18250 \text{ LOC / tahun} \end{aligned}$$

Soal Latihan No. 2a :

Gambar dan buatlah perbandingan biaya, baik untuk Review maupun NonReview dari ilustrasi berikut ini:

1. Pada tahap rancangan awal :
  - A. Kesalahan yang timbul = 10
  - B. Efisiensi dengan review = 70%
2. Pada tahap rancangan terinci :
  - A. Sebanyak 50%, kesalahan dilewatkan, sisanya diperkuat dengan faktor penguat = 2
  - B. Kesalahan baru yang muncul = 25
  - C. Efisiensi dengan review = 50%
3. Pada tahap coding / unit test
  - A. sebanyak 40% kesalahan dilewatkan, dan sisanya diperkuat dengan faktor penguat 3.
  - B. Efisiensi dengan review 80%, dan non review 50%.
  - C. Kesalahan baru yang muncul = 20%.
4. Pada tahap selanjutnya dilakukan perbaikan dengan efisiensi masing-masing = 50%
5. Biaya yang harus ditanggung untuk setiap kesalahan adalah :
  - A. Selama rancangan = 2 satuan harga
  - B. Sebelum test = 5 satuan harga
  - C. Selama test = 20 satuan harga
  - D. Setelah dipasarkan = 60 satuan harga

*Jawab :*

**TABEL PERBANDINGAN BIAYA ANTARA DENGAN REVIEW & NON REVIEW**

KEGIATAN	BIAYA per ERROR	NON REVIEW	REVIEW
Selama desain			
Sebelum test ( coding )			
Selama test ( validated & integrated test )			
Setelah dipasarkan ( error laten )			
Total biaya			

Soal Latihan No. 2b :

Gambar dan buatlah perbandingan biaya, baik untuk Review maupun NonReview dari ilustrasi berikut ini:

1. Pada tahap rancangan awal :
  - A. Kesalahan yang timbul = 10
  - B. Efisiensi dengan review = 80%
2. Pada tahap rancangan terinci :
  - A. Sebanyak 60%, kesalahan dilewatkan, sisanya diperkuat dengan faktor penguat = 3
  - B. Kesalahan baru yang muncul = 20
  - C. Efisiensi dengan review = 40%
3. Pada tahap coding / unit test
  - A. sebanyak 30% kesalahan dilewatkan, dan sisanya diperkuat dengan faktor penguat 2.
  - B. Efisiensi dengan review 75%, dan non review 45%.
  - C. Kesalahan baru yang muncul = 25%.
4. Pada tahap selanjutnya dilakukan perbaikan dengan efisiensi masing-masing = 60%
5. Biaya yang harus ditanggung untuk setiap kesalahan adalah :
  - A. Selama rancangan = 3 satuan harga
  - B. Sebelum test = 10 satuan harga
  - C. Selama test = 30 satuan harga
  - D. Setelah dipasarkan = 70 satuan harga

*Jawab :*

**TABEL PERBANDINGAN BIAYA ANTARA DENGAN REVIEW & NON REVIEW**

KEGIATAN	BIAYA per ERROR	NON REVIEW	REVIEW
Selama desain			
Sebelum test ( coding )			
Selama test ( validated & integrated test )			
Setelah dipasarkan ( error laten )			
Total biaya			