

# REKAYASA PERANGKAT LUNAK (RPL)



**Tim Dosen RPL**  
**Program Studi D3 Teknik Informatika**  
**Fakultas Ilmu Terapan**



# Aturan Perkuliahan

Mata kuliah diadakan sebanyak 3 SKS:

- 1 x 50 menit → teori di kelas
- 4 x 50 menit → praktikum bersama  
(Tugas Pendahuluan, Tes Awal, Diskusi, Jurnal)

# Penilaian

## Prosentase Penilaian RPL

**Assessment I 25% (Minggu ke-5 & 6)**

**Assessment II 25% (Minggu ke-11 & 12)**

**TUBES 25% (Minggu ke 15 & 16)**

**Praktikum 25%**

(Tugas Pendahuluan (20%); Jurnal (70%); Skill (10%))

Softskill yang dimasukkan dalam pengajaran: leadership, communication, dan teamwork.

# Belajar Apa ?

## I. KONSEP DASAR PERANGKAT LUNAK DAN PEMODELAN PROSES

- Peran PL
- Definisi PL
- Aktivitas Fundamental dari proses PL  
(Pengenalan Power Designer)
- Model Proses PL
- Prescriptive Model
- Specialized Process Model
- The Unified Process  
(Rational Software Architect)

# Belajar Apa (lanjutan) ?

## II. ANALISIS KEBUTUHAN PERANGKAT LUNAK

### *Understanding Requirement :*

- Requirement Engineering
- Developing Usecase
- Building analysis Model  
(Use Case Diagram)

### *Requirement Modeling :*

- Requirement Analysis
- UML Modeling (Usecase and developing activity diagram)  
(Class Diagram)
- Software Requirement Specification (SRS)  
(Activity Diagram)

# Belajar Apa (lanjutan) ?

## III. DESAIN PERANGKAT LUNAK

*Konsep dan prinsip desain Sistem :*

- Proses dan Prinsip desain
- Konsep desain

(Sequence Diagram dan Communication Diagram)

*The Design Model :*

- Data Design Element
- Architectural Design Element
- Interface Design Element
- Component-level Design Element
- Deployment Design Element

(Component Diagram dan Deployment Diagram)

- Architectural Design :
- SW Architecture
- Architectural Consideration
- Architectural Design

(State Machine Diagram)

- Mobile App Design

(Strategy Pattern)

- Dokumentasi desain
- (SW Design Document)

(Factory Method Pattern)

# Belajar apa (lanjutan) :

## IV. PENGUJIAN PERANGKAT LUNAK

Metode Pengujian PL :

- Dasar – dasar, prinsip & objektif pengujian SW
- desain Kasus Uji
- Metoda Testing

(Implementasi Decorator Pattern)

# Belajar apa (lanjutan) :

## **V. JAMINAN KUALITAS PERANGKAT LUNAK**

SW Quality Assurance

(Asistensi Tugas Besar)



# Belajar apa (lanjutan) :

**PROYEK PERANGKAT LUNAK (MOBILE APPLICATION)**

Presentasi TUBES

(Pengujian Akhir)

## **Definisi Perangkat Lunak :**



**Suatu kumpulan instruksi dari beberapa program komputer yang apabila di eksekusi akan menyediakan beberapa fungsi dan pekerjaan yang diinginkan.**

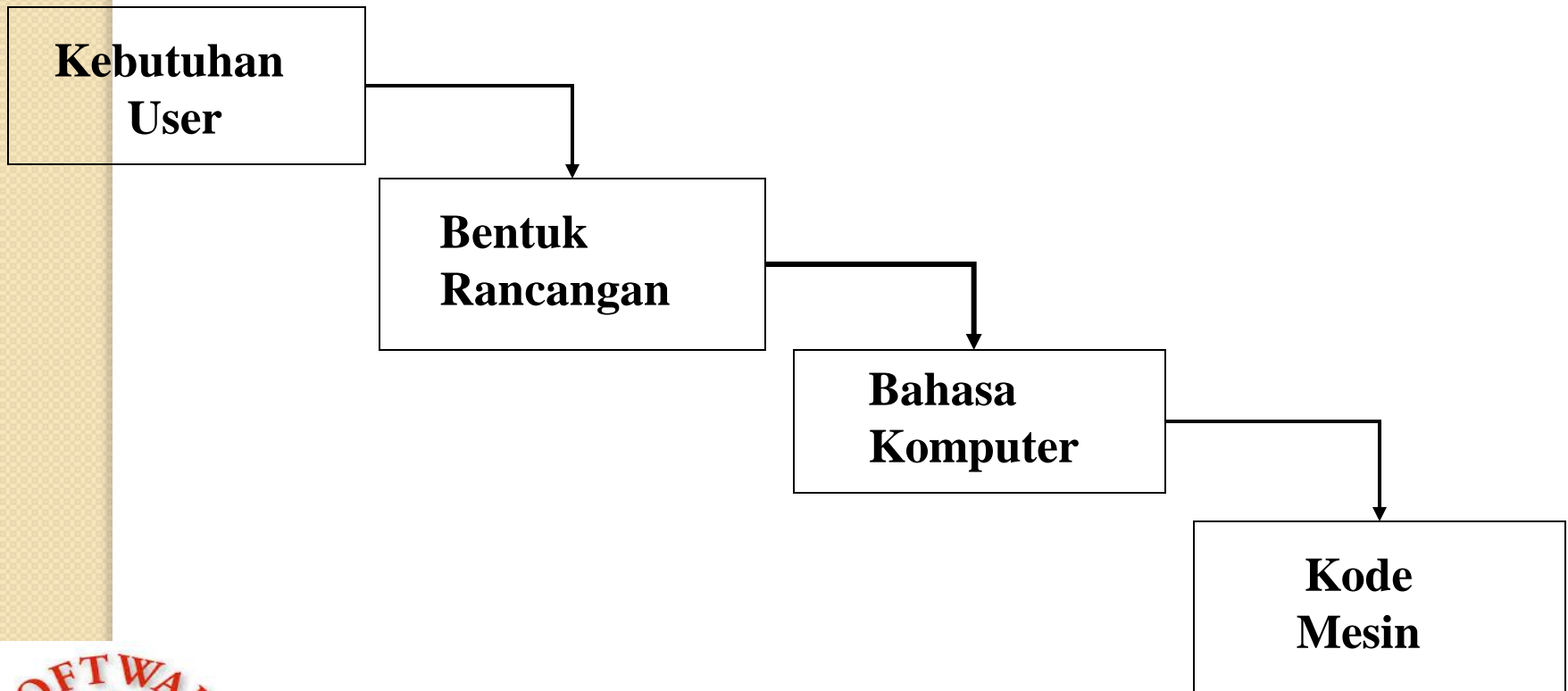
**Kumpulan data yang membuat suatu program komputer dapat memanipulasi informasi.**

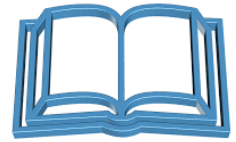
## Karakteristik perangkat lunak yang direkayasa dengan baik adalah sebagai berikut :



- Mudah dirawat → dilengkapi dokumentasi & perubahan dapat dilakukan dengan biaya minimum.
- Dapat diandalkan → bekerja seperti yg diharapkan & gagal hanya bila keluar dari spesifikasinya.
- Bekerja efisien → tidak memboroskan sumber daya seperti : memory, prosesor, penyimpanan.
- Dirancang sesuai dengan tingkat kemampuan pemakai.

- ❖ Produk perangkat lunak dikembangkan dari serangkaian perubahan
- ❖ Dari user REQUIREMENTS menjadi kode eksekusi untuk mesin.





**Rekayasa Perangkat Lunak berupaya untuk menghasilkan :**

- 1. Komponen PL yg dapat dipakai ulang (Reusability) dengan maksud : dapat dimanfaatkan pada berbagai program.**
- 2. Komponen PL (Reusable Component) terdiri dari :**

SUBROUTINE

= Berisi Algoritma

OBJECT  
CLASS

= Berisi Algoritma & Struktur Data

- 3. Produk PL yang lengkap dengan dokumentasinya.**



## **Dua macam produk Perangkat Lunak :**

- 1. Generik = Produk yg dikembangkan utk dijual kepada publik.**
- 2. Spesifik = Produk yg dikembangkan khusus utk sebuah company.**

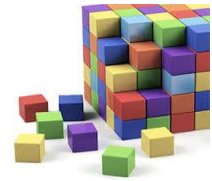
## **Jenis-jenis Produk Perangkat Lunak adl sbb :**

### **\* System Software**

- Program utk mengatur/ melayani program-program lain.**
- Banyak berinteraksi dengan Perangkat Lunak.**

### **\* Real Time Software**

- Perangkat lunak yang memonitor, menganalisa, mengendalikan suatu kejadian atau peristiwa yg sedang terjadi.**
- Waktu tanggap (response) sangat singkat ( milidetik).**



## **\* Business Software**

**Adl Perangkat Lunak yang menangani bermacam aplikasi seperti :**

- Penggajian**
- Penjualan**
- Persediaan Barang**
- dan lain- lain**
- Berbagai aplikasi tersebut bisa disatukan menjadi suatu SIM (Sistem Informasi Manajemen).**

## **\* Engineering & Scientific Software**

**Adl suatu aplikasi Perangkat Lunak yg banyak memproses angka-angka seperti :**

- Astronomi**
- Otomotif**
- Peramalan cuaca**
- dan lain - lain**

# Mitos Tentang Perangkat Lunak



- \* **Banyak permasalahan dalam sebuah PL yg datang dari asumsi-asumsi yg kebenarannya tidak dapat dipertanggungjawabkan.**
- \* **Tiga kelompok yg terkait dalam RPL adalah sbb :**
  - 1. Manajemen**  
**manajemen RPL harus dapat mengatur anggaran, menjaga jadwal dari kelambatan, & meningkatkan kualitas.**
  - 2. Customer ( Pemakai )**  
**Customer ingin PL dikembangkan untuk mendukung rekan kerja, divisi/ departemen, dan pihak luar berdasarkan kontrak.**
  - 3. Practitioner ( Pengembang )**  
**yang mengembangkan/ membuat Perangkat Lunak, diantaranya Programmer.**



# Mitos Tentang Perangkat Lunak

(Mitos dipihak manajemen )



**Mitos**



**Adanya panduan & prosedur pasti lancar**

**Kenyataan**



**Apakah ...**

- disadari keberadaannya ?
- lengkap ?
- dipakai ?
- sesuai kebutuhan ?

# Mitos Tentang Perangkat Lunak

(Mitos dipihak manajemen )



**Mitos** → **Peralatan baru & modern**

**Kenyataan** → **Penguasaan Tools jauh lebih penting dari Hardware dan Software.**

**Mitos** → **Bila terlambat dari jadwal tambah programmer**

**Kenyataan** → **Tambah jumlah programmer mengakibatkan waktu Pekerjaan semakin lambat.**

# Mitos Tentang Perangkat Lunak

(Mitos dipihak pemakai )



**Mitos** →

**Tujuan sistem scr umum cukup utk membuat PL,  
Rincian menyusul saat program dikembangkan.**

**Kenyataan** →

**Rincian kebutuhan sangat penting dimulai dari :**

- + Fungsi**
- + Performance**
- + Interface**
- + Batasan rancangan**
- + Kriteria validasi**

**Hanya dapat diperoleh dengan komunikasi yg intensif**

# Mitos Tentang Perangkat Lunak

(Mitos dipihak pengembang )



**Mitos** → **Kualitas hanya bisa diketahui setelah program berjalan ( Running ).**

**Kenyataan** → **Kualitas dapat dijaga sejak PL dikembangkan.**

**Mitos** → **Yang diserahkan ke User adalah program.**

**Kenyataan** → **Yang diserahkan adalah konfigurasi PL, rincian program Dan dokumentasi.**



**4 AKTIVITAS UTAMA yang dilakukan oleh perekayasa perangkat Lunak adalah sebagai berikut :**

**1. SPESIFIKASI**

**Berisi tentang spesifikasi perangkat lunak & batasan operasional**

**2. PENGEMBANGAN**

**Berisi tentang tahapan pengembangan sesuai spesifikasi.**

**3. VALIDASI**

**Berisi tentang tahapan pengujian agar sesuai dengan spesifikasi.**

**4. EVOLUSI**

**Melakukan penyesuaian yang mengikuti perubahan kebutuhan.**

# **THE WATERFALL MODEL / THE LINEAR SEQUENTIAL MODEL**

## **Waterfall Model / The linear Sequential Model**

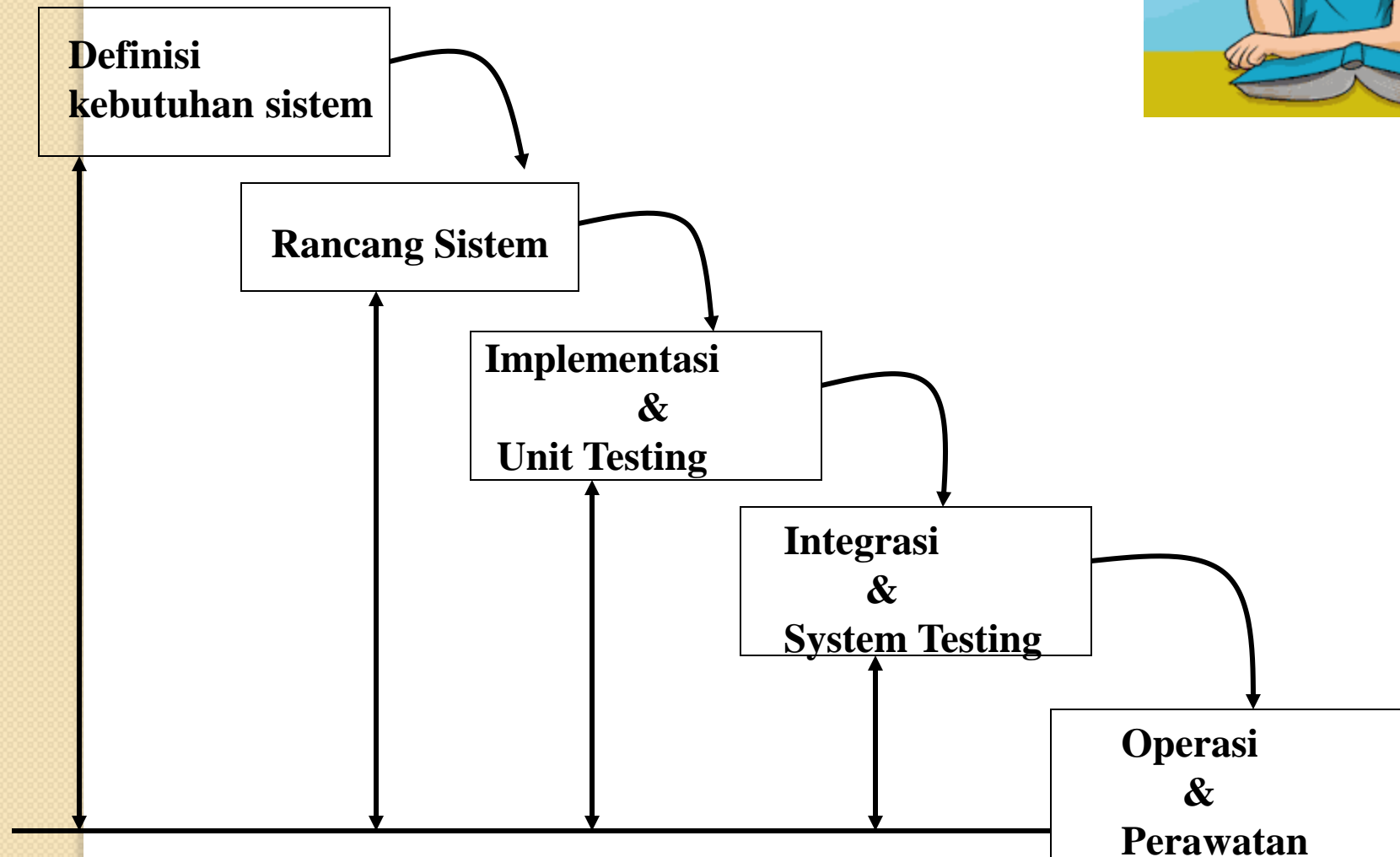
**adalah suatu daur hidup klasik ( the oldest life cycle model ) yang  
dijadikan acuan pada pengembangan suatu perangkat lunak yang  
berisi**

**Tahapan sebagai berikut :**

- 1. Definisi kebutuhan sistem**
- 2. Rancangan sistem**
- 3. Implementasi & unit testing**
- 4. Integrasi & sistem testing**
- 5. Operasi & perawatan**



## BAGAN ALIR WATERFALL MODEL

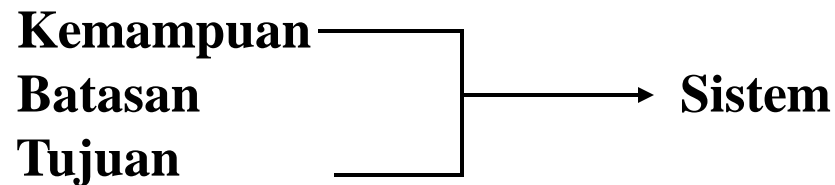




## **PENJELASAN TAHAPAN DALAM WATERFALL MODEL**

### **1. ANALISA & DEFINISI KEBUTUHAN SISTEM**

**Diuraikan tentang :**



### **2. RANCANG SISTEM & PERANGKAT LUNAK**

**Adalah transformasi kebutuhan User kedalam bentuk perangkat lunak yang terdiri dari :**

**Arsitektur Sistem**  
**Kebutuhan Hardware**  
**Kebutuhan Software**



### **3. IMPLEMENTASI & UNIT TESTING**

- Pembuatan program komputer dimulai untuk membentuk sebuah SW
- Diuji kesesuaian dan ketepatannya.

### **4. INTEGRASI & SISTEM TESTING**

- Pembentukan sebuah sistem.
  - \* unit-unit diintegrasikan
  - \* diuji sebagai sebuah sistem

### **5. OPERASI & PERAWATAN**

- Pemakaian & penyesuaian
  - \* Sistem dimanfaatkan
  - \* Perbaikan, perubahan & pengembangan.





## **PENJELASAN WATERFALL MODEL**

**Disebut juga daur hidup klasik, merupakan paradigma yang sudah lama, Namun tetap digunakan sampai saat ini**

**Problema yang dihadapi paradigma adalah sebagai berikut :**

- 1. Tahapan proyek yang sesungguhnya tidak sequential.**
- 2. Tahapan proyek banyak mengalami iterasi/ pengulangan.**
- 3. Pada dasarnya sulit mendefinisikan kebutuhan secara jelas.**
- 4. Pada paradigma ini bentuk kerja lambat terlihat.**
- 5. Kesalahan di awal tahap berakibat sangat fatal.**

**Paradigma yang paling banyak dipakai :**

- 1. Paling banyak diikuti & diterapkan**
- 2. Masih dianggap sesuai dengan keadaan sekarang**
- 3. Walaupun dengan segala kekurangan yang dimiliki**



# **PROTOTYPE**

**Paradigma Prototype/ Prototyping digunakan apabila ditemui kondisi-kondisi sbagai berikut :**

**Definisi user bersifat umum.**

- User tidak tahu pasti apa yang diinginkan.**

**Definisi user tidak bersifat rinci**

- User tidak tahu pasti apa & bagaimana bentuk :**
  - + masukan**
  - + Proses**
  - + Keluaran**

**Pengembang merasa tidak pasti tentang :**

- + pilihan algoritma yang akan dipakai**
- + Bagaimana lingkungan sistem yang akan dikembangkan**
- + Bentuk, sifat, & karakteristik antar muka pemakai.**



**Intinya ada ketidak pastian :**

- Dipihak User**
  - + Tentang apa yang diinginkan**
  
- Dipihak Pengembang**
  - + Apa yang harus dilakukan**

**Terdapat 2 macam bentuk Prototyping :**

**1. EVOLUTIONARY**

**Dimulai dari model yang dikembangkan lalu akhirnya dimanfaatkan.**

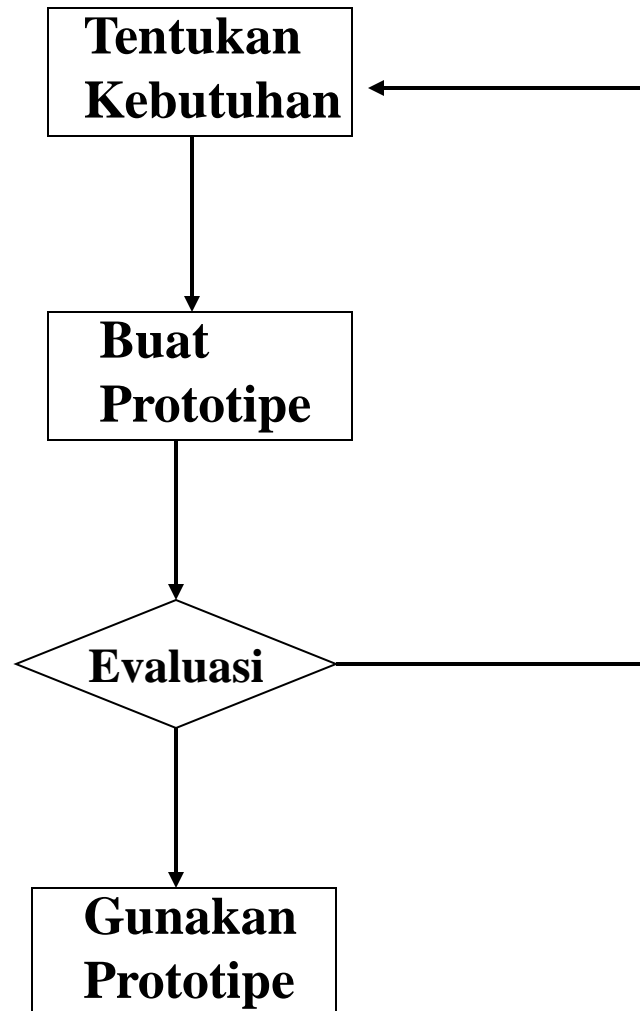
**2. THROWAWAY**

**Hanya dibuat sebagai model untuk mencari bentuk yang diinginkan ( cetak biru ).**



## BAGAN ALIR PROTOTYPING/ PROTOTYPE

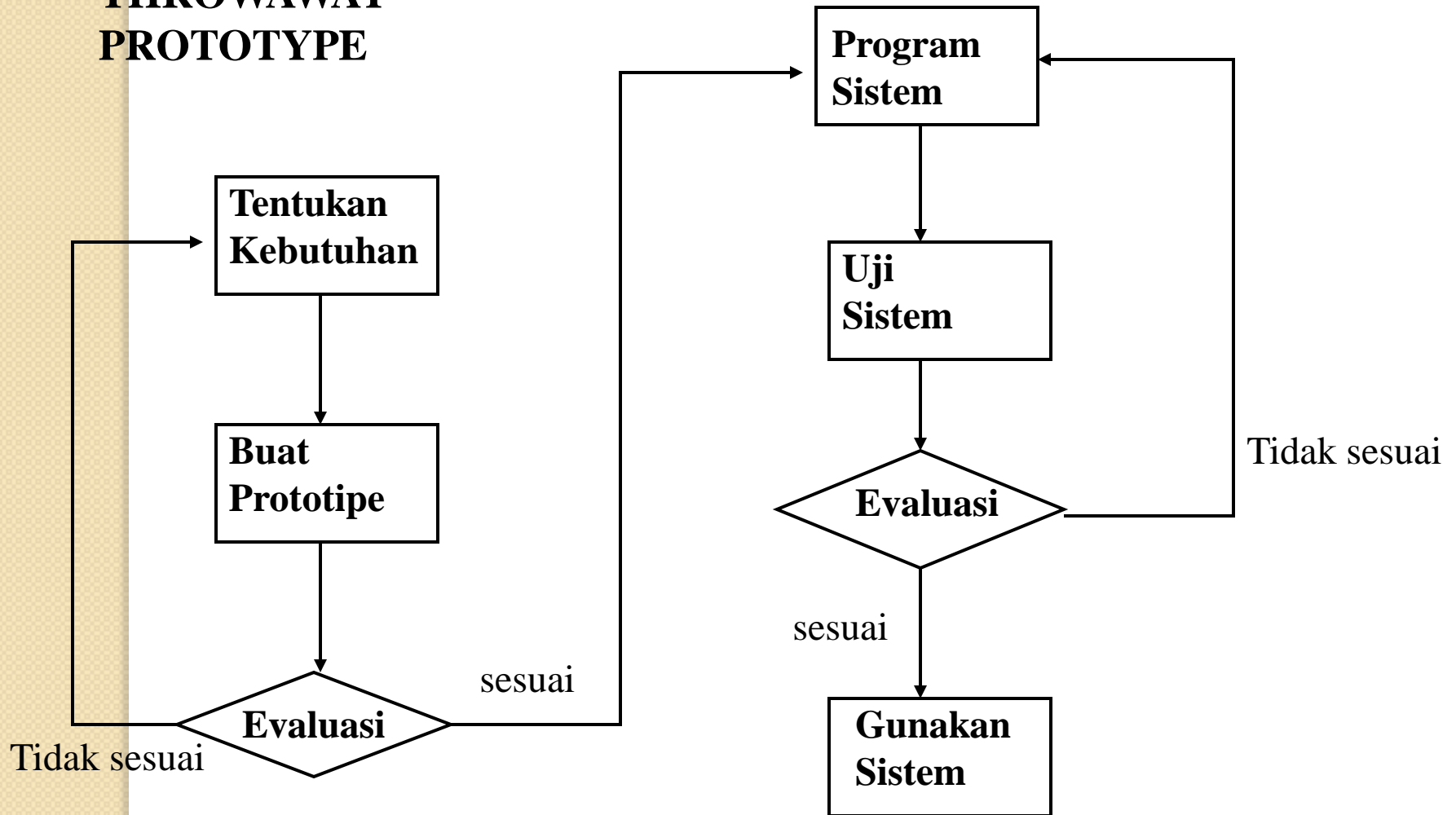
### EVOLUTIONARY PROTOTYPE





## BAGAN ALIR PROTOTYPING/ PROTOTYPE

### THROWAWAY PROTOTYPE





## **EMPAT MODEL PROTOTIPE**

### **1. PROTOTIPE KERTAS**

**Gambaran sistem dibuat pada media kertas.**

- + Tidak mempunyai bagian yang operasional (berbentuk program), dapat diujicoba ( test), dapat diimplementasikan ( run/ execute )**

### **2. PROTOTIPE BERBASIS PC**

- + Pemodelan dilakukan dengan memakai program aplikasi.**
- + Program-program presentasi.**
- + Untuk memperlihatkan interaksi manusia-komputer.**

### **3. PROTOTIPE KERJA**

- + Implementasi sebagian fungsi sistem.**
- + Fungsi yang ingin dilihat karakteristiknya.**
- + Dibuatkan programnya.**

#### **4. PROTOTIPE PROGRAM**

- + Program benar-benar dibuat & bisa bekerja**
- + Bagian program yang sudah berfungsi**
- + Terus menerus ditambah & dilengkapi.**







## **KEUNGGULAN PROTOTIPE :**

- 1. Komunikasi User- Developer**
  - + Frekuensi komunikasi meningkat**
  - + Pengembang akan selalu meminta pendapat User.**
- 2. Membantu Analis**
  - + Menentukan kebutuhan user yang sebenarnya**
  - + Meminimalkan salah persepsi.**
- 3. Peran User meningkat**
  - + Evaluasi oleh User berkali-kali**
  - + User bisa memberikan masukan setiap saat.**
- 4. Pengembangan lebih cepat**
  - + Program bisa langsung dibuat.**
  - + User melihat perkembangan tahap demi tahap.**

## **5. Implementasi mudah**

- + User sudah mengenal perangkat lunak yang dikembangkan.**
- + User tidak akan merasa asing.**
- + Sejak awal User sudah merasa memiliki.**





## **KELEMAHAN PROTOTIPE :**

### **1. Pemakai sibuk :**

- + User & Pengembang harus sama-sama/ memiliki komitmen.**
- + Menyediakan waktu untuk bertemu.**
- + Keduanya sepakat untuk kerja sama.**

### **2. Pemakai sulit melakukan evaluasi :**

- + Bentuk prototipe sering berubah**
- + Disesuaikan dengan kebutuhan User itu sendiri.**

### **3. User ingin cepat selesai**

- + Bentuk program sudah terlihat dari awal.**
- + User merasa tidak akan lama lagi selesai.**
- + Pengembang sering mengabaikan dokumentasi.**



**4. User berharap terlalu banyak**

- + Keberhasilannya membawa dampak.**
- + Sering evaluasi & komunikasi membuat user menjadi sering berubah keinginan dan tidak pasti dengan kebutuhan.**

**5. Prototipe bekerja tidak efisien**

- + lebih mementingkan keberhasilan.**

**PROTOTIPE/ PROTOTYPING** baik dipakai pada kondisi/ keadaan sebagai berikut :

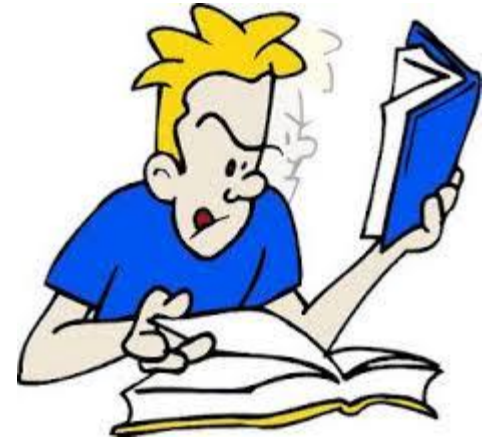
- 1. Sistem mempunyai resiko tinggi.**
  - + Tidak jelas permasalahannya.**
  - + Tidak jelas kebutuhan & keinginan.**
  - + Tidak pasti apa yang ingin dilakukan.**
- 2. Perancangan dialog User – Komputer.**
  - + Bagaimana membuat dialog amah, dan mudah.**
- 3. Sistem diminati oleh banyak pemakai**
  - + Mencari kesepakatan**
  - + Basis untuk menyamakan persepsi.**
- 4. User ingin cepat selesai**
  - + User tidak sabar menunggu.**
  - + Prototipe segera memperlihatkan bentuk kerja sistem**



5. **Masa pakai singkat**
  - +Sistem hanya dipakai beberapa kali saja.
6. **Ingin menunjukkan inovasi**
  - +Pengembang dapat menunjukkan kecanggihan.
  - + Sistem cepat terlihat ( mungkin cepat selesai ).
7. **Kebutuhan berubah rubah**
  - +User sulit menjelaskan kebutuhan.
  - +Menjadi keadaan yang paling umum untuk memakai Prototyping.



## **MODEL SPIRAL**



- **EVOLUTIONARY PROCESS**
  - + Pengembangan bertingkat
- **Menggabungkan keunggulan**
  - + Prototyping
  - + Waterfall
- **Memungkinkan dikembangkan perangkat lunak**
  - + secara bertahap (Incremental)
  - + dengan cepat
- **Terbagi atas 6 tahapan**
  1. Customer Communication.
  2. Planning.
  3. Risk Analysis.
  4. Engineering.
  5. Construction & Release.
  6. Customer Evaluation.

# MODEL SPIRAL



**PLANNING**

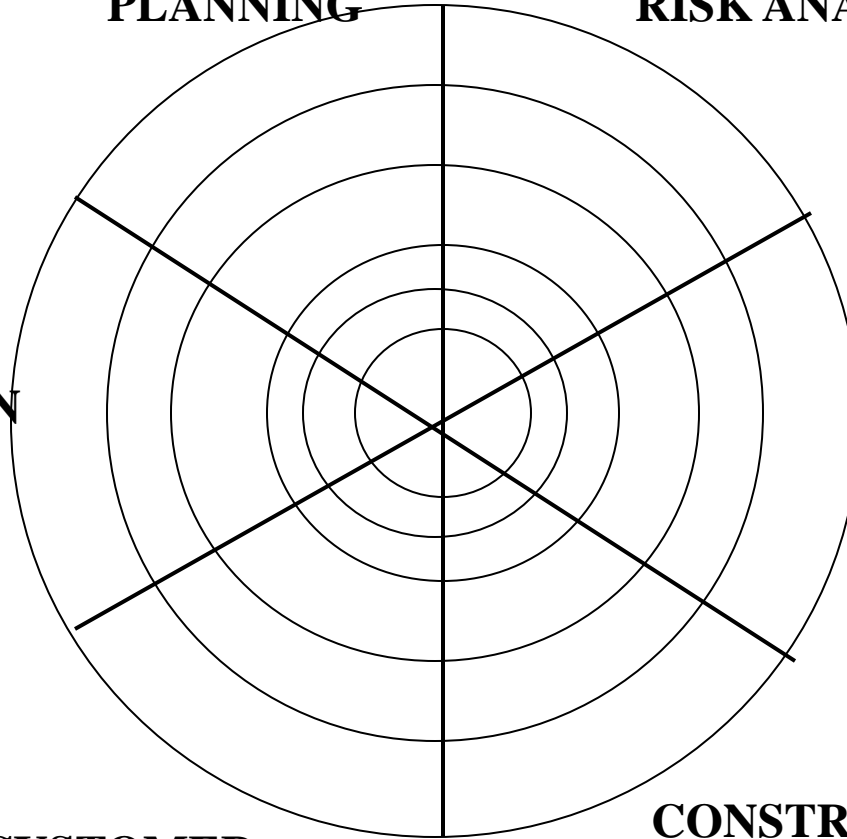
**RISK ANALYSIS**

**ENGINEERING**

**CONSTRUCTION  
& RELEASE**

**CUSTOMER  
EVALUATION**

**CUSTOMER  
COMMUNICATION**







**Penjelasan dari tahap-tahap prototipe adl sebagai berikut :**

- 1. Customer communication**
  - \* Penerapan komunikasi antara usesr dengan developer.**
- 2. Planning**
  - \* Menentukan tujuan, alternatif, batasan sistem**
  - \* Penentuan kebutuhan awal**
  - \* Dilanjutkan dengan hasil evaluasi user.**
- 3. Risk analysis**
  - \* Analisa resiko yg didasari oleh evaluasi user.**
  - \* Analisa resiko berdasarkan kebutuhan awal.**
  - \* Identifikasi resiko.**
  - \* Penanganan resiko.**
- 4. Engineering**
  - \* Pengembangan produk**
  - \* Dimulai dengan prototipe awal**
  - \* Sampai menjadi produk jadi.**



**5. Construction & release**

- \* Tahap konstruksi, test, install**
- \* Penyiapan user support (dokumentasi ).**

**6. Customer evaluation**

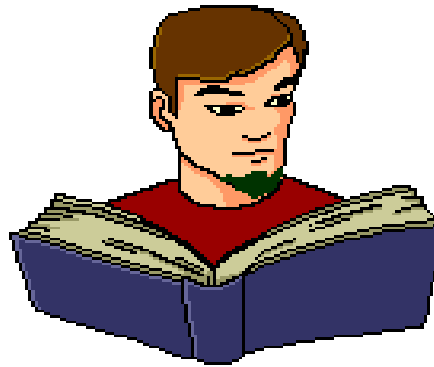
- \* Penilaian hasil pengembangan produk oleh user pada tahap pengembangan maupun tahap instalasi.**

**Perbedaan penting antara model spiral dengan model proses perangkat lunak lainnya adl :**

**Dilakukannya pertimbangan resiko scr eksplisist pada model spiral.**

**Secara informal, risiko adl sesuatu yg bisa berjalan salah.**





Selamat Belajar

