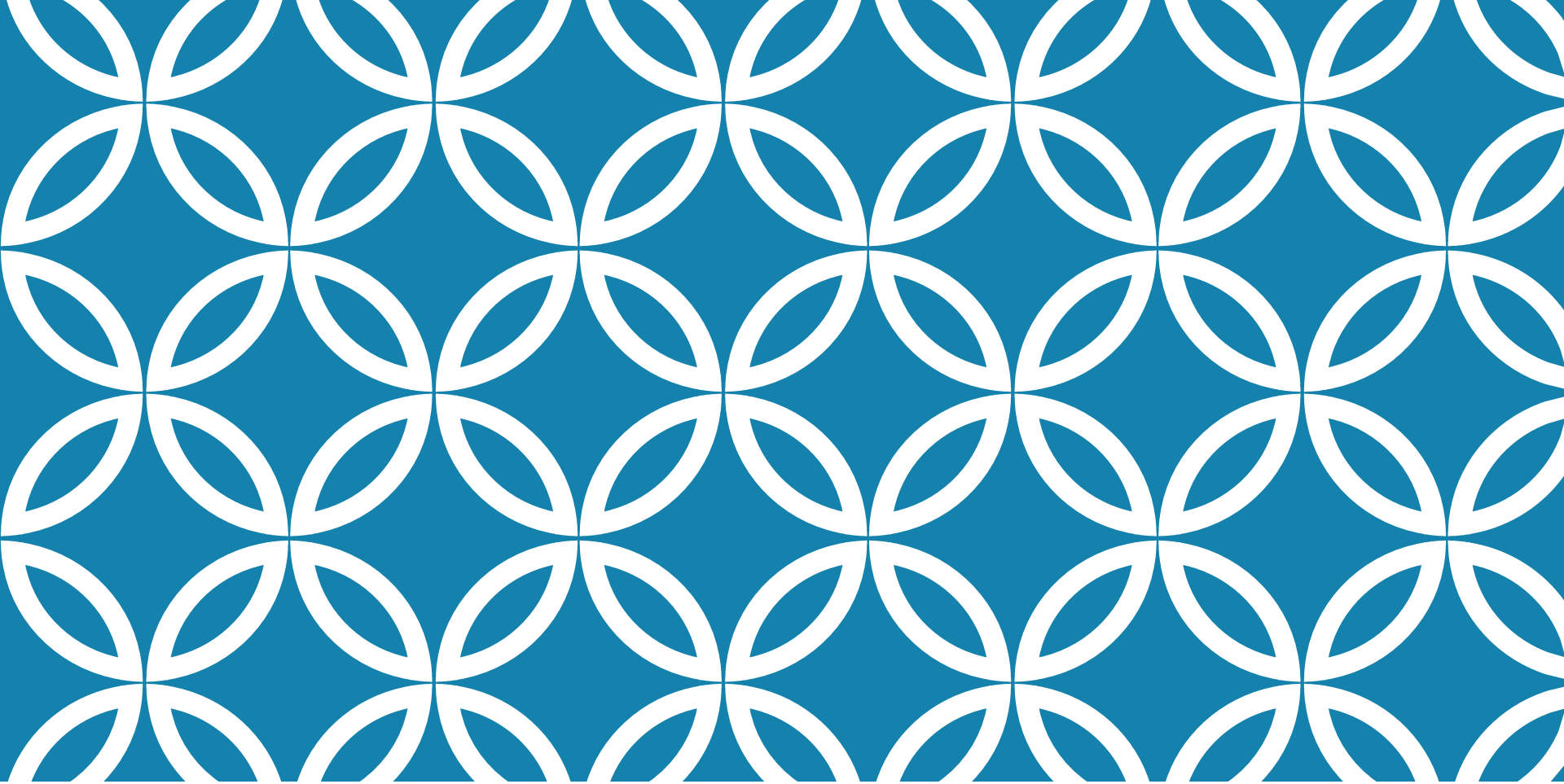


# PEMELIHARAAN PERANGKAT LUNAK

# TUJUAN PERAWATAN PIRANTI LUNAK

Agar software dapat tetap mendukung bisnis yang terus berubah.

# FAKTA TERKAIT MAINTENANCE

- Biaya untuk maintain sebuah program adalah >= 40% dari biaya untuk mendevelop program tersebut.
- Memperbaiki sebuah *defect* memberi peluang (20% s/d 50%) munculnya *defect* yang lain (Fred Brooks, The Mythical Man-Month).
- Software yang sukses pun tetap akan memerlukan maintenance.

# SOFTWARE MAINTENANCE

Proses perubahan sistem yang dilakukan setelah sistem tersebut di deliver.

3 Tipe software maintenance:

- *Maintenance* untuk memperbaiki kesalahan di dalam software
- *Maintenance* untuk meningkatkan adaptasi software terhadap lingkungan operasionalnya
- *Maintenance* untuk menambah dan memodifikasi fungsi sistem

# TEHNIK PEMELIHARAAN PERANGKAT LUNAK

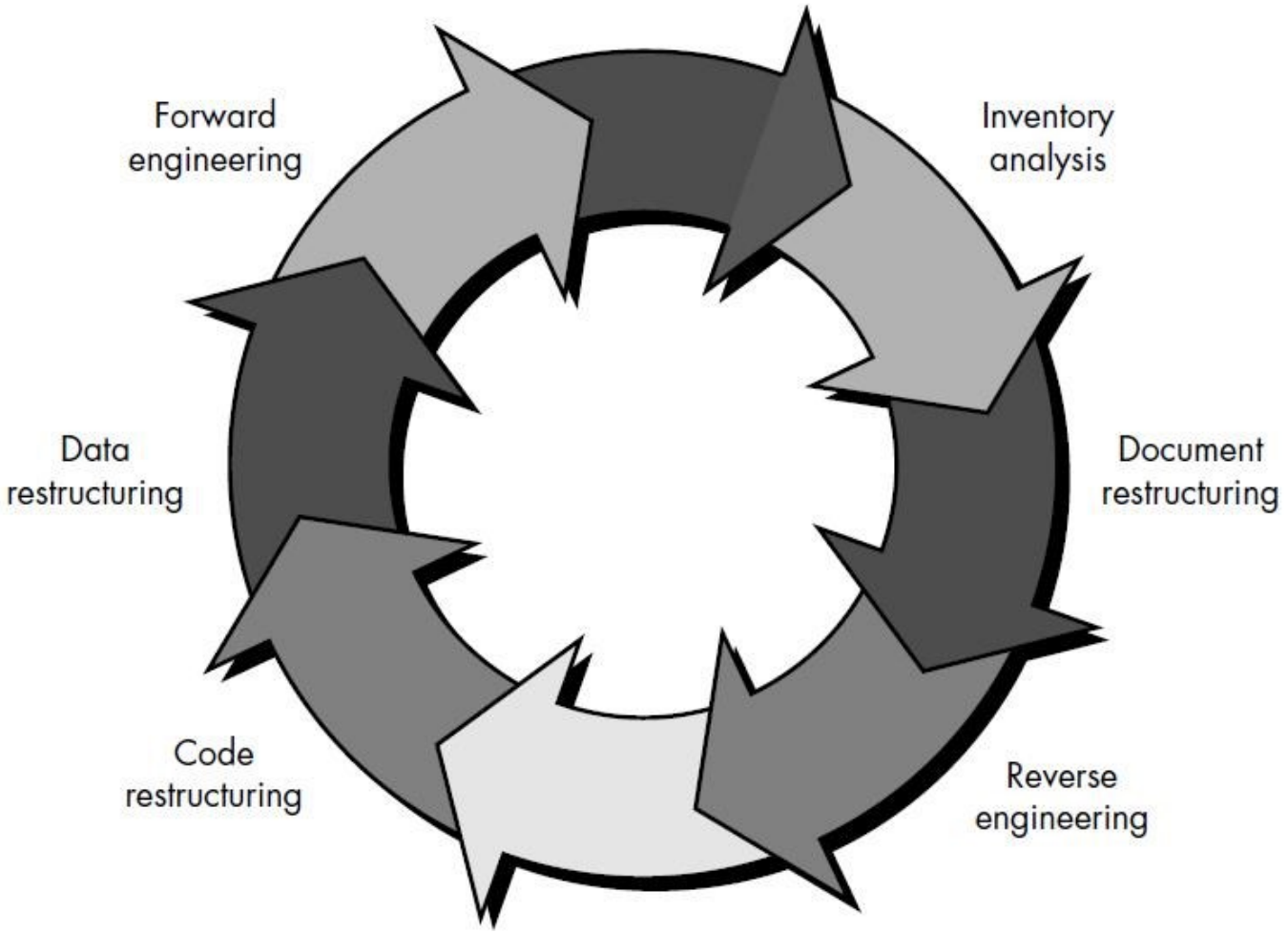
- Corrective Maintenance: Modifikasi software yang bersifat reaktif, yang dilakukan untuk memperbaiki permasalahan yang ditemukan.
- Adaptive Maintenance: Modifikasi software yang bertujuan agar software tersebut, tetap dapat digunakan walaupun terjadi perubahan lingkungan.
- Perfective Maintenance: Modifikasi software yang bertujuan meningkatkan performance atau maintainability.
- Preventive Maintenance: Modifikasi software yang bertujuan untuk mendeteksi dan memperbaiki potensi-potensi kesalahan, sebelum kesalahan tersebut benar-benar terjadi.

# BIAYA MAINTENANCE SOFTWARE

Hal yang mempengaruhi biaya maintenance:

- Stabilitas tim
- Kontrak maintenance
- Skill/kemampuan dari staff
- Struktur dan usia program

# MODEL PROSES REENGINEERING SOFTWARE

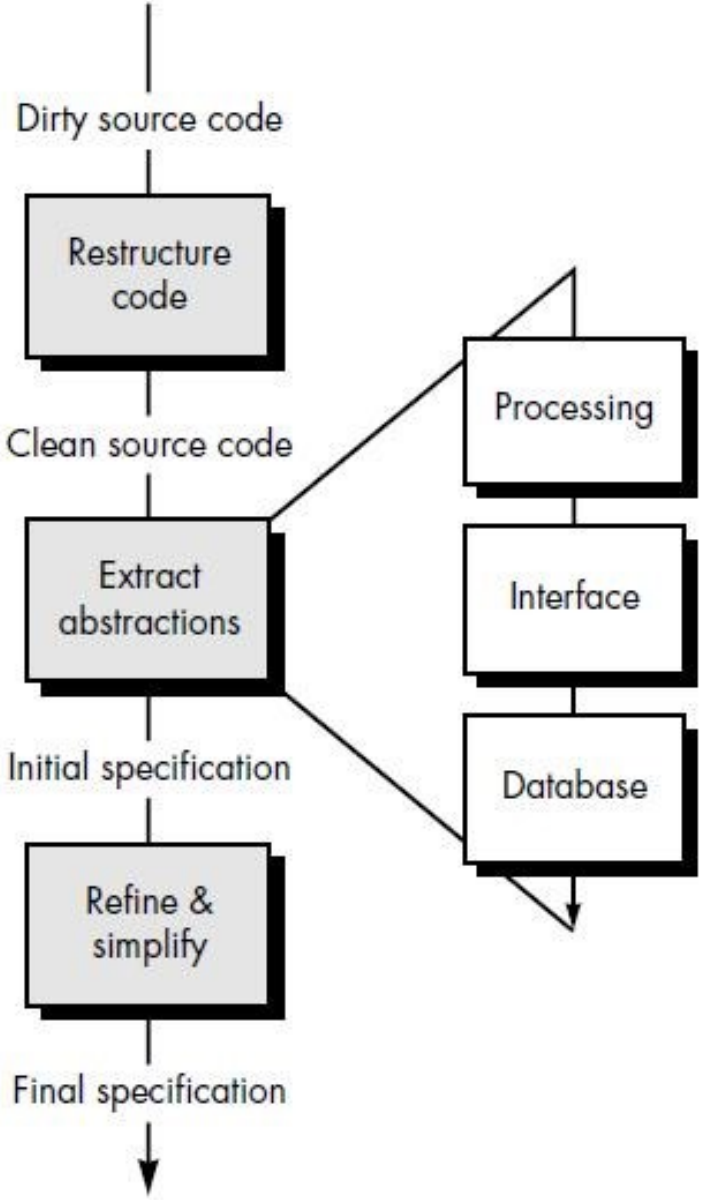


# MODEL PROSES REENGINEERING SOFTWARE

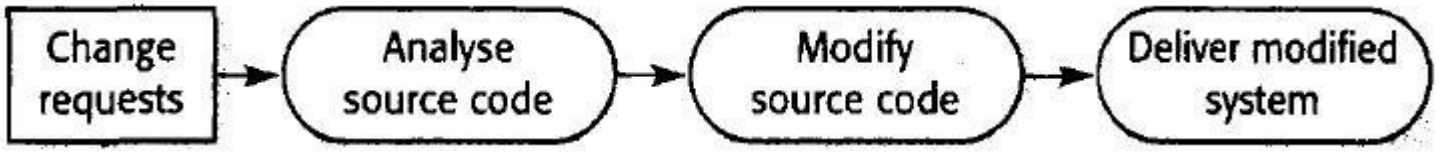
- Analisa Inventory
- Restrukturisasi Dokumen
- Reverse Engineering
- Restrukturisasi Data & Program
- Forward Engineering

# REVERSE ENGINEERING

- Menganalisa program untuk membuat abstraksi high-level dari program tsb
- Sebuah proses perbaikan terhadap desain.
- Source code/program → desain data, desain arsitektur dan desain prosedur/alur kerja sistem
- Beberapa CASE tools dapat membantu melakukan reverse engineering secara otomatis


Dirty source code
Restructure code
Clean source code
Extract abstractions
Initial specification
Refine & simplify
Final specification
Processing
Interface
Database

# PERBAIKAN DALAM KEADAAN DARURAT




Keadaan darurat biasanya terjadi saat:

- Terjadi masalah terhadap sistem yang kritikal.
- Permasalahan mempengaruhi lingkungan operasional
- Munculnya pesaing baru atau peraturan baru

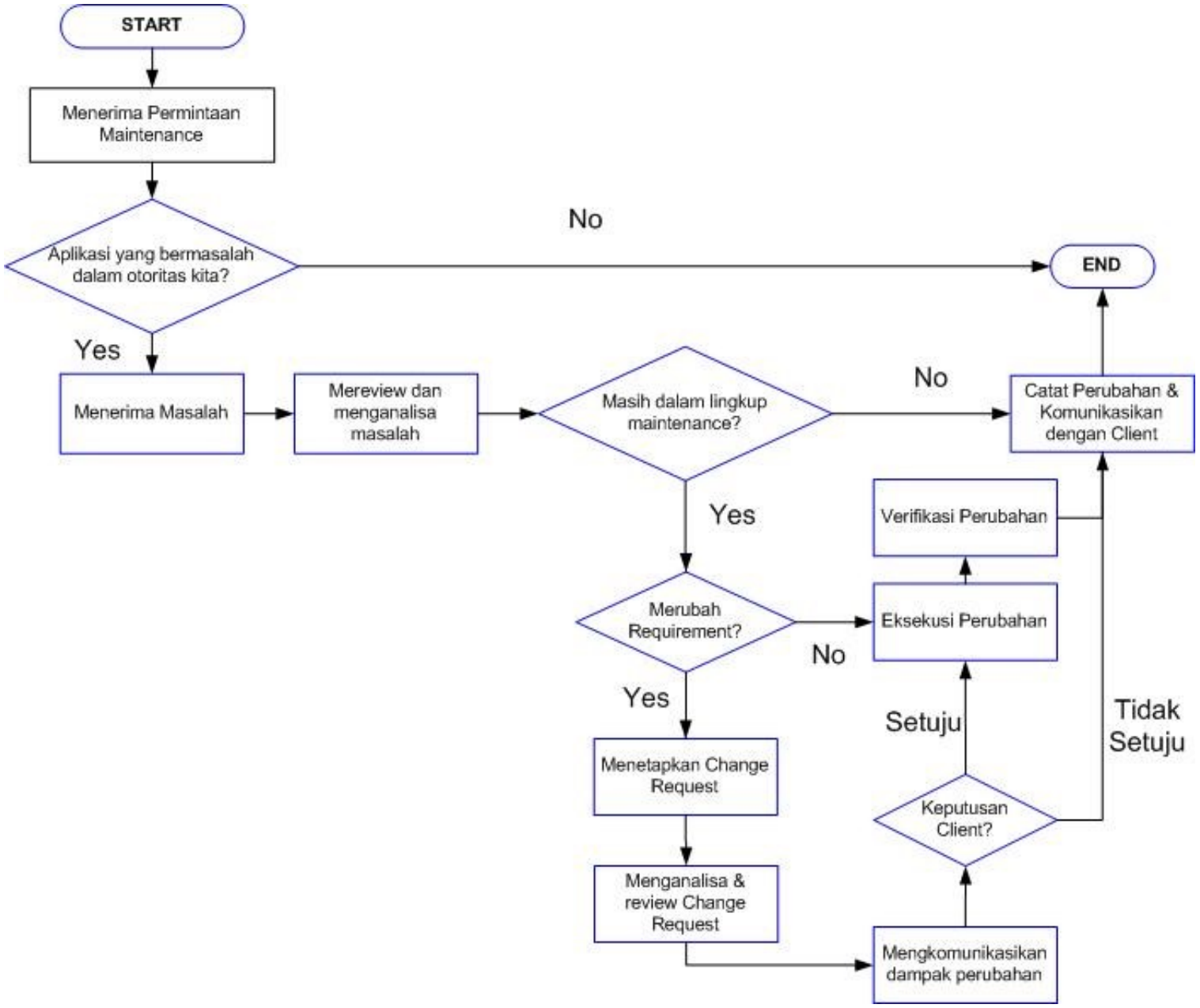
$\Rightarrow$ Dapat menyebabkan ketidaksesuaian antara requirement, desain dengan coding.

# FORWARD ENGINEERING VS RE-ENGINEERING


Forward engineering


Software re-engineering

# ALUR KERJA MAINTENANCE SOFTWARE



# HUKUM LEHMAN

Perilaku yang dirumuskan oleh Lehman & Belady, terkait dengan perawatan (maintenance) & evolusi software.

Terdiri dari:

- Perubahan yang kontinu: Sebuah software harus berubah agar tetap digunakan
- Kompleksitas yang meningkat: Semakin sebuah software berevolusi, semakin meningkat kompleksitasnya, kecuali ada usaha untuk mengelolanya.
- Evolusi program yang besar
- Stabilitas organisasi
- Konservasi familiaritas
- Perkembangan yang kontinu: Fungsi sebuah software harus terus berkembang agar kenyamanan user tetap terjaga.
- Kualitas yang menurun: Kualitas software akan selalu menurun, kecuali ada usaha untuk merawat dan beradaptasi dengan lingkungan.
- Sistem umpan balik (feedback system)

# PENGUKURAN PERAWATAN SOFTWARE

- Software Maturity Index (SMI): mengukur stabilitas sebuah software.
- SMI = [MT – (Fa + Fc + Fd) / MT]
  - MT: Jumlah modul keseluruhan yang ada di versi saat ini
  - Fc: Jumlah modul yang mengalami perubahan yang ada di versi saat ini
  - Fa: Jumlah modul yang ditambah yang ada di versi saat ini
  - Fd: Jumlah modul yang dihapus/delete
- SMI mendekati nilai 1 maka software tersebut semakin stabil

# CODE REFACTOR

Proses merubah *source code*

Tujuan:

- Source code lebih mudah dibaca
- Mengurangi kompleksitas
- Meningkatkan kemudahan maintenance

Contoh refactoring:

- Merubah nama Method dan Field menjadi nama yang lebih menggambarkan kegunaannya.