



RPL101

Pengantar Rekayasa Perangkat Lunak *Introduction to Software Engineering*

Semester Ganjil Tahun Ajaran 2025-2026

Penamaan pada zoom seperti berikut: **Kelas_NIM_Nama**

Contoh: A_43425010xx_Budi Berbudi
B_43425010xx_Cici Kelinci
C_43425010xx_Didi Mariadi



PROGRAM STUDI
**TEKNOLOGI REKAYASA
PERANGKAT LUNAK**
POLIBATAM

5. Implementasi *Implementation*



Tahapan Implementasi

Tahapan implementasi terdiri dari pengembangan (*development*) dan dokumentasi

- Mengubah spesifikasi program menjadi kode
- Merancang berbagai pengujian yang akan dilakukan terhadap sistem yang baru
- Memfinalisasi dokumentasi sistem (*system documentation*) dan membuat dokumentasi untuk pengguna (*user documentation*)

Tujuan tahapan ini adalah untuk mengimplementasikan desain sebaik mungkin.

Panduan dalam Implementasi

1. Standar dan Pedoman Coding

- Standar penamaan
 - Variabel: huruf kecil dengan underscores (misal `nama_keluarga`)
 - Function: huruf kecil dengan underscores (misal `akar_kuadrat`)
 - Class: huruf pertama huruf kapital (misal `Mobil`)
 - Konstanta: huruf kapital (misal `PI`, `GRAVITASI`)
 - Prinsip utama: Nama mudah dipahami, bukan singkatan kode
- Layout Kode
 - Gunakan baris kosong untuk memisahkan antar bagian
 - Gunakan indentansi untuk menunjukkan struktur bersarang
 - Konsisten dengan spasi diantara operator
 - Panjang kalimat dibatasi (misal 79 karakter)

Panduan dalam Implementasi

1. Standar dan Pedoman Coding

- Dokumentasi dan Komentar
 - Blok komentar untuk menjelaskan tujuan, bukan detail implementasi
 - Hindari menggunakan komentar dalam kode untuk klarifikasi

2. Gunakan Library Open Source

- Meningkatkan produktivitas secara signifikan dengan menggunakan kode yang sudah ada
- Contoh Python: Lebih dari 500,000 project tersedia di PyPI
- Kebiasaan baik: Cari library yang tersedia sebelum menulis kode baru

Mengelola Kode

- Organisasi
 - Struktur oleh arsitektur: Folder disusun berdasarkan komponen aplikasi
 - Layout standar proyek: Pisahkan source code, test, dokumentasi, data, dan konfigurasi
 - Struktur tipikal:
 - `/myapp/` (source code)
 - `/tests/` (test scripts)
 - `/docs/` (dokumentasi)
 - `/data/` (data)
 - File konfigurasi (`requirements.txt`, `setup.py`, `README.md`, `LICENSE`)

Mengelola Kode

- *Source Code Control*
Gunakan tools seperti Git dan GitHub untuk *version management*:
- Kebutuhan Seorang Developer
 - Mengembalikan keadaan semula: *Commit checkpoints* untuk kemampuan rollback
 - Pengembangan parallel: Cabang untuk banyak fitur, bergabung saat selesai
 - Perintah: git commit, git checkout, git merge
- Kebutuhan Tim Developer
 - *Repository cloning*: git clone untuk mendapatkan proyek secara lokal
 - *Synchronization*: git pull untuk memperbarui copy lokal dengan perubahan
 - *Contribution*: git push untuk mengirim perubahan ke repository
 - *Code review*: Pull requests (PRs) sebelum bergabung ke cabang utama

Menulis Kode

1. *Incremental Coding Process*

- Menulis kode sedikit demi sedikit
- Test setiap penambahan sebelum menambahkan fungsi selanjutnya
- Keuntungan: Debugging lebih mudah karena error pasti berasal dari penambahan terakhir
- Kebutuhan: Test scripts otomatis untuk pengujian yang sering

2. *Test-Driven Development (TDD)*

- Proses: Tuliskan test dulu, kemudian tuliskan kode agar lulus test
- Keuntungan: Kode tersinkronisasi dengan test, prioritas berdasarkan fitur
- Tantangan: Mungkin ada edge cases yang tidak tercover oleh test

3. *Pair Programming*

- Dua programmer bekerja bersama dalam kode yang sama
- Satu orang mengetik sedangkan orang yang lain dapat mereviu dan berdiskusi
- Keuntungan: Reviu kode secara kontinu
- Kerugian: Berpotensi produktivitas rendah tapi kualitas tinggi

1. Ukuran besar

- *Lines of Code* (LOC): Paling umum, baris yang bukan komentar atau baris kosong
- *Halstead Metrics*: Berdasarkan operator dan operand
 - Vocabulary: $n = n_1 + n_2$ (jumlah operators + operands)
 - Length: $N = N_1 + N_2$ (total kemunculan)
 - Volume: $V = N \times \log_2(n)$

2. Matrik kompleksitas

- *Cyclomatic Complexity*
 - Definisi: Jumlah decision points + 1
 - Dasar teori graf: $e - n + p$ (edges - nodes + komponen terhubung)
 - Penggunaan: Disarankan untuk menjaga kompleksitas di bawah 10
- *Halstead Complexity*
 - Difficulty: $D = (n_1/2) \times (N_2/n_2)$
 - Mengukur kesulitan berdasarkan kemunculan operator dan penggunaan operan

Isu Lain dalam Implementasi

1. Reuse

Sebagian besar perangkat lunak modern dibangun dengan menggunakan kembali komponen atau sistem yang sudah ada. Saat mengembangkan perangkat lunak, kita harus memanfaatkan kode yang sudah ada semaksimal mungkin.

2. Configuration management

Selama proses pengembangan, kita harus melacak banyak versi berbeda dari setiap komponen perangkat lunak menggunakan manajemen konfigurasi.

3. Host-target development

Perangkat lunak produksi biasanya tidak dijalankan di komputer yang sama dengan lingkungan pengembangan perangkat lunak. Biasanya kita mengembangkan di satu komputer (*host*) dan menjalankannya di komputer yang lain (*target*).

Level Penggunaan Kembali

1. The **abstraction** level

Pada level ini, kita tidak menggunakan kembali perangkat lunak secara langsung, tetapi menggunakan pengetahuan tentang abstraksi dalam desain perangkat lunak.

2. The **object** level

Pada level ini, kita langsung menggunakan kembali objek dari library daripada menulis kode sendiri.

3. The **component** level

Komponen adalah kumpulan objek dan class yang kita gunakan kembali dalam sistem.

4. The **system** level

Pada level ini, kita menggunakan kembali seluruh sistem.

Configuration Management

- *Configuration management* adalah istilah yang diberikan untuk proses umum pengelolaan perangkat lunak yang terus berubah.
- Tujuan *configuration management* adalah untuk mendukung proses integrasi sistem sehingga semua pengembang dapat mengakses kode dan dokumen proyek secara terkendali, mengetahui perubahan yang telah dibuat, serta mengompilasi dan menghubungkan komponen untuk menciptakan sebuah sistem.
- Aktivitas dalam *configuration management*:
 - Version management: menyediakan dukungan untuk melacak berbagai versi komponen perangkat lunak.
 - System integration: menyediakan dukungan untuk membantu pengembang menentukan versi komponen apa yang digunakan untuk membuat setiap versi sistem.
 - Problem tracking: menyediakan dukungan untuk memungkinkan pengguna melaporkan bug dan masalah lainnya, dan untuk memungkinkan semua pengembang melihat siapa yang sedang menangani masalah ini dan kapan masalah tersebut diperbaiki.

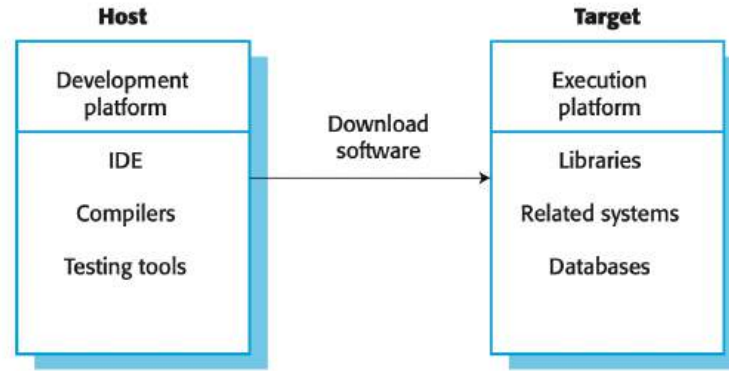
Host-target development

- Sebagian besar perangkat lunak dikembangkan di satu komputer (*host*), tetapi berjalan di mesin yang lain (*target*). Secara umum, disebut sebagai platform pengembangan dan platform eksekusi.
- Platform lebih dari sekadar perangkat keras. Platform mencakup sistem operasi yang terpasang ditambah perangkat lunak pendukung lainnya seperti sistem manajemen basis data atau, untuk platform pengembangan, lingkungan pengembangan interaktif.
- Platform pengembangan biasanya memiliki perangkat lunak yang berbeda dengan platform eksekusi; platform-platform ini mungkin memiliki arsitektur yang berbeda.

Host-target development

Tool pengembangan perangkat lunak sering dikelompokkan untuk menciptakan *integrated development environment* (IDE).

IDE dibuat untuk mendukung pengembangan dalam bahasa pemrograman tertentu seperti Java.



Jika suatu komponen dirancang untuk arsitektur perangkat keras tertentu, atau bergantung pada sistem perangkat lunak lain, komponen tersebut jelas harus diterapkan pada platform yang menyediakan dukungan perangkat keras dan perangkat lunak yang diperlukan.

Pengembangan Open Source

Merupakan pendekatan pengembangan perangkat lunak dimana source code dipublikasikan dan sukarelawan diajak untuk berpartisipasi dalam proses pengembangannya.

Berawal dari Free Software Foundation (www.fsf.org)

Keuntungan:

- Open-source software biasanya murah atau bahkan gratis, tapi jika kita membutuhkan dokumentasi atau support, kita harus membayar, walaupun biasanya sangat murah.
- Open-source software biasanya sangat reliable karena memiliki pengguna dalam jumlah besar yang bersama-sama memperbaiki masalah yang ditemukan.

Contoh open source software dan tempat para sukarelawan saling bekerjasama untuk mengatasi masalah/issue yang ditemukan:

https://issues.apache.org/jira/



Dashboards ▾ Projects ▾ Issues ▾

Search

Log In

New search

«

All issues

Discard changes

Export ▾

Tools ▾

Find filters

Apache Scienc... ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ Contains text More ▾ Search Advanced

FILTERS

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

FAVOURITE FILTERS

You must be *logged in*
to view favourite filters.

Order by Created ▾



SDAP-368
Update matchup device and platform...

SDAP-367
Improve deletebyquery tool performa...

SDAP-366
Fix CSV and NetCDF outputs for mat...

SDAP-364
Upgrade Solr to 8.11.1

SDAP-363
Update SwaggerUI to display list of a...

SDAP-362
Update tile format for gridded and L2...

SDAP-361
Add unit test for time series algorithm

1 2 3 4 5 >



Apache Science Data Analytics Platform / SDAP-362

6 of 352 ^ ▾

Update tile format for gridded and L2 data

Export ▾

Details

Type: Improvement Status: TO DO
Priority: Major Resolution: Unresolved
Affects Version/s: None Fix Version/s: None
Component/s: None
Labels: None

Issue Links

links to

GitHub Pull Request #151

Activity

All Comments Work Log History Activity Transitions


People

Assignee: Unassigned
Reporter: Stepheny Perez
Votes: 0 Vote for this issue
Watchers: 1 Start watching this issue

Dates

Created: 15/Dec/21 19:
Updated: 26/Jan/22 21:

https://jira.mongodb.org/

 MongoDB

Dashboards ▾ Projects ▾ Issues ▾

Search

🔊 ? Log In

New search

«

All issues

Discard changes

Export ▾ Tools ▾

Find filters

MongoDB Data... ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ Contains text More ▾ Search Advanced

FILTERS

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

FAVORITE FILTERS

You must be logged in to view favorite filters.

Order by Created ▾ ↓

TOOLS-3074
Mongod hangs on shutdown and RA...

TOOLS-3073
Add support for FLE to mongo tools

TOOLS-3072
Install and configure MongoDB in a W...

TOOLS-3071
location and file ownership of tools bi...


TOOLS-3070
where to download sha-256 for data...

TOOLS-3063
Release Tools with Amazon Linux 2022

TOOLS-3062
Add Amazon Linux 2022 to Tools

TOOLS-3060

1 2 3 4 5 ▸

 MongoDB Database Tools / TOOLS-3074

1 of 2517 ▴ ▾

Export ▾

Mongod hangs on shutdown and RAM usage reaches upto 98%

Details

Type: Bug

Priority: Major - P3

Affects Version/s: 3.6.2

Component/s: None

Labels: None

Environment: Production

Status: NEEDS TRIAGE

Resolution: Unresolved

Fix Version/s: None

People

Assignee: Unassigned

Reporter: Ishrat Jahan

Votes: 0 Vote for this issue

Watchers: 2 Start watching this issue

Dates


Created: Mar 23 2022 06:15:04 AM UTC

Description

Problem Statement/Rationale

On trying to stop the mongod process using the command - `sudo systemctl stop mongod`, the process got stuck. And the RAM usage started increasing and reached upto 98%. After the systemd timeout, mongod process was killed. We saw below log for this

https://issues.redhat.com/



[Dashboards](#) [Projects](#) [Issues](#) [Create](#)

[Log In](#)

[New search](#)

[Find filters](#)

FILTERS

[My open issues](#)

[Reported by me](#)

[All issues](#)

[Open issues](#)

[Done issues](#)

[Viewed recently](#)

[Created recently](#)

[Resolved recently](#)

[Updated recently](#)

FAVORITE FILTERS

You must be logged in to view favorite filters.

All issues

Discard changes

Order by Created

Order by icon

RHBZ-14

Wrong branding in Red Hat Enterpris...

RHBZ-13

Downloading operator from one quay...

RHBZ-12

Please create new Release for Produ...

RHBZ-11

We need to resync bugzilla compone...

RHBZ-10

Image pruning in ocp3.11 is not deleti...


RHBZ-9

NoExecute taint is being applied whe...

RHBZ-8

Creating worker nodes using machin...

RHBZ-7



Red Hat Bugzilla for Red Hat Developers

RHBZ-14

Wrong branding in Red Hat Enterprise Linux Beta 4 in gnome-tour and g-i-s

Details

Type: Bug

Priority: Minor

Component/s: None

Labels: None

Steps to Reproduce:

Install Red Hat Enterprise Linux Beta 4

The gnome-tour and gnome-initial setup still using Fedora 34 Branding:

(pretty sure someone has seen this before me nothing major)

Status: OPEN

Resolution: Unresolved

People

Assignee: Unassigned

Reporter: Luna Jernberg

Votes: 0 Vote for this issue

Watchers: 1 Start watching this issue

Dates

Created: 2022/03/15 2:29 PM

Export

Tools

Red Hat Bugzilla...

Type: All

Status: All

Assignee: All

Contains text

More

Search

Advanced

1 of 14

Export

Aktiviteter

Rundtur

15

Stäng

Selesai. 😊