

EL 2007 Sinyal dan Sistem

Tim Sinyal dan Sistem

2025-08-02

Table of contents

Preface	3
1 Introduction	4
2 Summary	5
References	6
3 Panduan Penyiapan Repositori GitHub	7
3.1 Langkah 2: Membangun Struktur Direktori Dasar	7
3.2 Langkah 3: Menyiapkan Konten Awal	8
3.3 Langkah 4: Mengatur Alur Kerja dan Instruksi untuk Mahasiswa	9
3.4 Langkah 5: Proses Penilaian dan Umpan Balik	9
3.5 Langkah 6 (Opsional, Sangat Direkomendasikan): Otomatisasi dengan GitHub Actions	10

Preface

This is a Quarto book.

To learn more about Quarto books visit <https://quarto.org/docs/books>.

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

2 Summary

In summary, this book has no content whatsoever.

References

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.

3 Panduan Penyiapan Repositori GitHub

Proses ini akan menciptakan ekosistem digital yang terpusat, transparan, dan kolaboratif untuk seluruh aktivitas perkuliahan. ## Langkah 1: Membuat Repositori Utama (Oleh Dosen/Admin)

Ini adalah langkah pertama untuk menciptakan “rumah” bagi seluruh materi dan aktivitas kuliah.

1. Buat Akun GitHub: Jika belum ada, buat akun GitHub untuk dosen atau laboratorium yang akan mengelola mata kuliah.
2. Buat Repositori Baru:
 - Klik tombol “New” di halaman utama GitHub Anda.
 - Nama Repositori: Beri nama yang jelas dan konsisten, misalnya Sinyal-Sistem-TE-ITB.1
 - Deskripsi: Tambahkan deskripsi singkat, contohnya “Repositori Resmi Mata Kuliah EL2007 Sinyal dan Sistem - The Sinyal Hunter Initiative”.
 - Visibilitas: Set ke Public agar dapat diakses oleh semua mahasiswa.
 - Inisialisasi: Centang “Add a README file” untuk membuat file deskripsi awal.
3. Klik “Create repository”.

3.1 Langkah 2: Membangun Struktur Direktori Dasar

Struktur folder yang rapi adalah kunci untuk menjaga agar repositori tetap terorganisir sepanjang semester.1

1. Clone Repositori: Di komputer Anda, buka terminal atau Git client, dan jalankan perintah git clone.
2. Buat Folder-folder Utama: Di dalam folder repositori yang baru saja di-clone, buat struktur direktori berikut:
 - 00_Materi_Ajar/
 - 01_Tantangan_Treasure_Hunt/

- 02_Knowledge_Maps/
 - 03_Hunter_Journals/
 - 04_Treasure_Submissions/
 - 05_Legacy_Archives/
3. Tambahkan File Placeholder: Git tidak melacak folder kosong. Agar struktur ini bisa di-commit, tambahkan file kosong bernama `.gitkeep` di dalam setiap folder.
 4. Commit & Push Struktur:
 - Jalankan `git add`.
 - Jalankan `git commit -m "feat: initialize course directory structure"`
 - Jalankan `git push origin main`

3.2 Langkah 3: Menyiapkan Konten Awal

Sebelum semester dimulai, isi repositori dengan materi-materi esensial.

1. Materi Ajar: Masukkan slide, catatan kuliah, atau video untuk minggu pertama ke dalam folder `00_Materi_Ajar/`. Sebaiknya dalam format Quarto (`.qmd`) atau PDF.¹
2. Tantangan Pertama: Tulis deskripsi tantangan untuk minggu pertama dan letakkan di dalam folder `01_Tantangan_Treasure_Hunt/`.
3. Buat Template:
 - Buat file `journal_template.qmd` yang berisi kerangka dasar untuk “Jurnal Harian Hunter”. Ini bisa berisi judul, bagian untuk refleksi mingguan, dan contoh cara memasukkan kode atau gambar.
 - Buat file `map_template.qmd` sebagai templat untuk “Peta Pengetahuan” kelompok, mungkin dengan contoh diagram Mermaid/Graphviz sederhana.
4. Perbarui README: Edit file `README.md` di direktori utama untuk menjelaskan tujuan repositori, aturan main, dan tautan-tautan penting.

3.3 Langkah 4: Mengatur Alur Kerja dan Instruksi untuk Mahasiswa

Ini adalah bagian terpenting untuk memastikan mahasiswa memahami cara berinteraksi dengan repositori. Alur kerja ini mengajarkan mereka praktik version control standar industri.¹

1. Komunikasikan Alur Kerja: Jelaskan alur kerja Fork -> Clone -> Branch -> Commit -> Push -> Pull Request kepada mahasiswa di pertemuan pertama.
2. Langkah-langkah untuk Mahasiswa:
 - Fork: Setiap mahasiswa harus melakukan Fork pada repositori utama Sinyal-Sistem-TE-ITB ke akun GitHub pribadi mereka. Ini menciptakan salinan repositori di bawah kendali mereka.
 - Clone: Mahasiswa melakukan Clone pada repositori hasil fork mereka (bukan repositori utama) ke komputer lokal.
 - Branch: Untuk setiap tugas atau kontribusi, mereka wajib membuat branch baru dengan nama yang deskriptif (misalnya, tugas-minggu-2-NIM atau update-jurnal-minggu-2). Pekerjaan tidak boleh dilakukan langsung di branch main.
 - Bekerja & Commit: Mahasiswa mengerjakan tugas atau menulis jurnal di branch tersebut. Mereka melakukan commit secara berkala untuk menyimpan perubahan.
 - Push: Setelah selesai, mereka melakukan Push branch tersebut ke repositori forked mereka di GitHub.
 - Pull Request (PR): Dari halaman GitHub mereka, mahasiswa membuka Pull Request (PR) dari branch kerja mereka ke branch main di repositori utama. PR ini berfungsi sebagai mekanisme pengumpulan tugas formal.

3.4 Langkah 5: Proses Penilaian dan Umpan Balik

Pull Request (PR) menjadi pusat untuk interaksi dan penilaian.

1. Review PR: Asisten atau dosen dapat mereview setiap PR yang masuk. Fitur review di GitHub memungkinkan pemberian komentar pada baris kode atau teks tertentu, memberikan umpan balik yang sangat spesifik.
2. Merge PR: Setelah tugas dinilai dan dianggap selesai, PR tersebut di-merge ke repositori utama. Ini secara resmi memasukkan pekerjaan mahasiswa ke dalam arsip kuliah.
3. Menutup PR: Jika tugas memerlukan revisi, PR dapat dibiarkan terbuka dan mahasiswa dapat melakukan push commit tambahan ke branch yang sama untuk memperbaikinya.

3.5 Langkah 6 (Opsional, Sangat Direkomendasikan): Otomatisasi dengan GitHub Actions

Untuk meningkatkan efisiensi, manfaatkan GitHub Actions.¹

1. Render Quarto ke Situs Web: Buat sebuah workflow GitHub Actions yang secara otomatis merender semua file .qmd (terutama materi ajar dan jurnal mahasiswa yang sudah di-merge) menjadi halaman HTML dan menerbitkannya menggunakan GitHub Pages. Ini akan menciptakan sebuah situs web dinamis untuk mata kuliah tersebut.
2. Pemeriksaan Otomatis: Untuk tantangan level “Uang” atau “Berlian” yang melibatkan kode, Anda bisa membuat workflow yang menjalankan tes otomatis pada kode yang disubmit dalam PR untuk memeriksa fungsionalitas dasar.

Dengan mengikuti langkah-langkah ini, Anda akan berhasil membangun sebuah ekosistem pembelajaran yang modern, interaktif, dan mempersiapkan mahasiswa dengan keterampilan teknis yang relevan dengan industri.

**