



# Pertemuan 2 Sistem Version Control & Kanban Board

Tim Ajar Dasar Pemrograman 2023

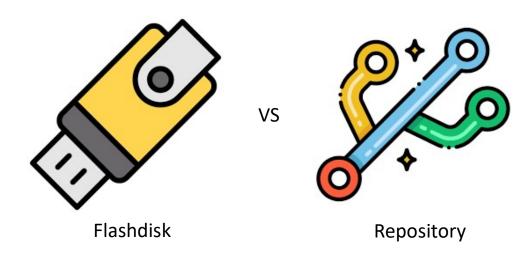


# Tujuan

- Mahasiswa memahami tentang apa itu version control, mengapa penting, dan bagaimana bekerja.
- Mahasiswa akan belajar prinsip-prinsip dasar Kanban, seperti visualisasi alur kerja, pembatasan pekerjaan yang sedang berlangsung, dan peningkatan berkelanjutan.



### Pendahuluan



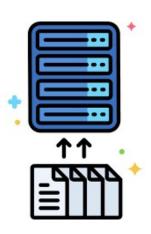
Repository memungkinkan tim pengembang untuk bekerja bersama secara terstruktur, melacak perubahan kode, dan mengelola versi berbagai komponen proyek.





# Apa itu Repositori?

Sebuah repositori (repository) dalam pengembangan perangkat lunak adalah tempat penyimpanan digital yang digunakan untuk mengelola kode sumber, berkas konfigurasi, dokumen, dan sumber daya lainnya yang terkait dengan suatu proyek.





# Komponen utama dari repositori

- Kode Sumber: berkas-berkas yang berisi instruksi-instruksi yang akan dieksekusi oleh komputer untuk menjalankan perangkat lunak.
- Dokumen dan Berkas Konfigurasi: berkas-berkas dokumentasi, petunjuk instalasi, dan berkas konfigurasi yang diperlukan untuk menjalankan perangkat lunak.
- Isu dan Tugas (Issues): menyediakan fitur untuk melaporkan isu atau masalah yang ditemukan dalam perangkat lunak.
- Branch dan Merge: memungkinkan pengembang untuk bekerja pada fitur atau perbaikan tertentu tanpa mempengaruhi kode di cabang utama. Setelah selesai, cabang tersebut dapat digabungkan (merge) kembali ke cabang utama.
- Riwayat Perubahan (Commit History): mencatat setiap perubahan yang dilakukan pada kode sumber



# Sistem Version Control (VCS)

Sistem Version Control (VCS) adalah sistem yang digunakan untuk melacak perubahan pada kode sumber dan berkas-berkas lain dalam suatu proyek perangkat lunak.



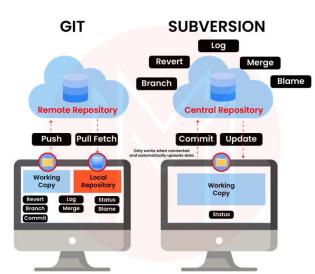


### Tipe-tipe VCS

- Sistem Version Control Tersebar (Distributed VCS): setiap anggota tim memiliki salinan penuh dari repositori. Contoh: Git.
- Sistem Version Control Sentral (Centralized VCS): repositori **pusat menyimpan semua versi kode sumber** dan berkas-berkas lainnya. Contoh: Subversion.













### **Manfaat Sistem Version Control**

- Pelacakan Perubahan: mencatat setiap perubahan yang dilakukan pada kode, memungkinkan pengembang untuk melihat sejarah perubahan dan kembali ke versi sebelumnya jika diperlukan.
- Kolaborasi Tim: memungkinkan beberapa pengembang untuk bekerja pada kode yang sama secara bersamaan, menggabungkan perubahan dengan aman.
- Manajemen Konflik: ketika dua atau lebih pengembang melakukan perubahan pada bagian yang sama dari kode, VCS membantu mendeteksi dan menyelesaikan konflik yang mungkin timbul.
- Pemulihan dan Pemecahan Masalah: Jika ada masalah dengan perubahan terbaru, memungkinkan untuk dengan mudah kembali ke versi sebelumnya yang berfungsi.
- Revisi dan Versi: memberi nomor unik pada setiap versi kode, memungkinkan pengembang untuk mengacu pada versi tertentu.
- Pengujian dan Pengujian Perangkat Lunak: memungkinkan untuk mengembangkan dalam lingkungan yang terisolasi, VCS mendukung pengujian dan eksperimen tanpa mempengaruhi kode inti.





# Peran GitHub, GitLab, dan Platform Serupa

Peran utamanya adalah menyediakan infrastruktur untuk **mengelola repositori Git dan memfasilitasi kolaborasi tim** dalam pengembangan perangkat lunak.

- Hosting Repositori; tempat untuk menyimpan repositori Git secara online.
- Kolaborasi Tim; mengizinkan tim pengembang untuk bekerja bersama dalam satu repositori.
- Manajemen Isu (Issues) dan Tugas; melacak dan mengelola isu, permintaan fitur, dan tugas dalam proyek.
- Pull Request (Permintaan Tarik); mengajukan perubahan kode untuk ditinjau oleh pengembang lain sebelum digabungkan ke repositori utama.
- Integrasi CI/CD; mendukung otomatisasi pengujian, penggabungan otomatis, dan penyebaran perangkat lunak.
- Pengaturan Akses; mengelola hak akses dan izin berdasarkan peran untuk memastikan kontrol yang tepat terhadap repositori.



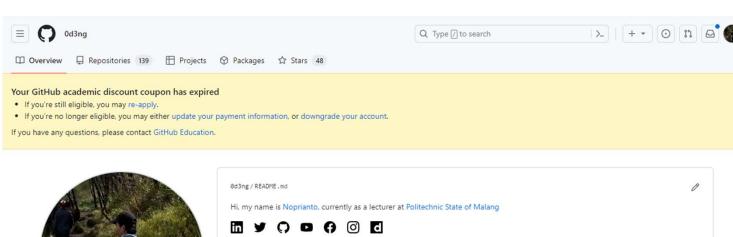
# Peran GitHub, GitLab, dan Platform Serupa (2)

- Dokumentasi Proyek; menyediakan dokumentasi proyek dan panduan bagi pengembang dan pengguna.
- Peninjauan Kode (Code Review); mendukung proses peninjauan kode dengan komentar, saran, dan tanggapan lainnya.
- Integrasi Pihak Ketiga; memiliki integrasi dengan alat-alat pengembangan lain seperti alat manajemen proyek, sistem pelacakan isu, dan alat perencanaan.
- Publikasi Open Source; pengembangan perangkat lunak sumber terbuka dengan memudahkan berbagi kode dengan komunitas.
- Versi Berbayar dan Gratis; berbayar dengan fitur tambahan, sementara yang lain memiliki opsi gratis untuk proyek open source.





- GitHub adalah platform pengelolaan repositori berbasis web yang memanfaatkan sistem kontrol versi Git.
- Salah satu platform paling populer dan digunakan secara luas oleh pengembang perangkat lunak, baik dalam proyek open source maupun proyek swasta.











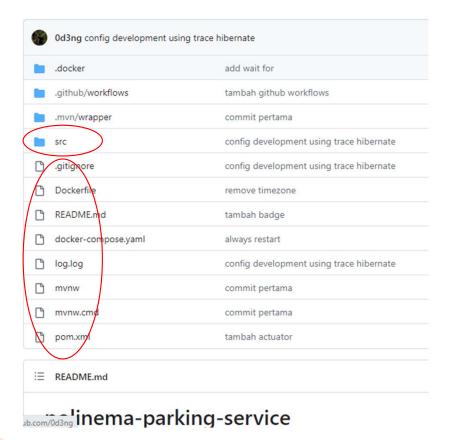


### Fitur Utama GitHub

- Repositori; membuat repositori publik atau pribadi untuk menyimpan kode sumber, berkas konfigurasi, dokumen, dan sumber daya lainnya
- Git Integration; melakukan operasi standar Git seperti commit, branch, merge, dan rebase melalui antarmuka web atau perintah baris
- Kolaborasi; dapat bekerja bersama dalam repositori yang sama
- Integrasi CI/CD; GitHub Actions adalah alat integrasi Continuous Integration (CI) dan Continuous Deployment (CD) yang terintegrasi langsung dengan repositori GitHub
- GitHub Pages; memungkinkan pengguna untuk meng-host situs web statis langsung dari repositori GitHub
- GitHub Classroom; platform pendidikan yang memfasilitasi pengajaran dan pembelajaran Git dan GitHub dalam lingkungan kelas



# GitHub - Repositori



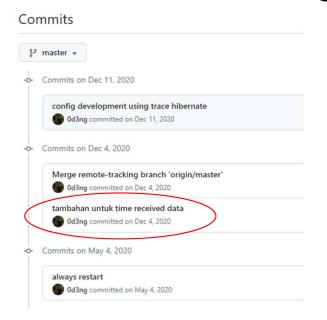


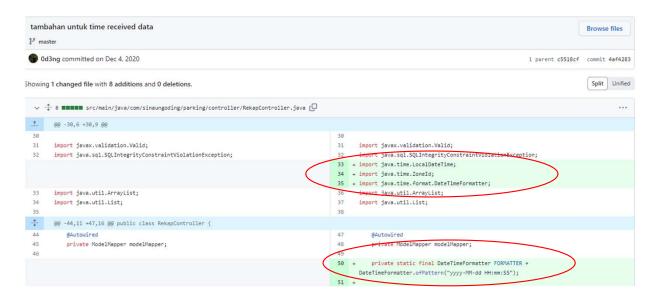
- **src**; folder yang digunakan untuk menyimpan kode program.
- File file yang lain adalah berisi tentang konfigurasi sebuah proyek. Misalkan file pom.xml adalah file konfigurasi untuk menyimpan seluruh depedensi/Pustaka ketika menggunakan Maven (build tool).





# **GitHub - Git Integration**



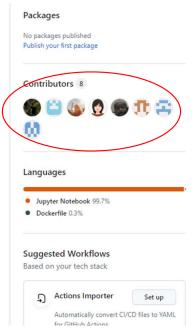


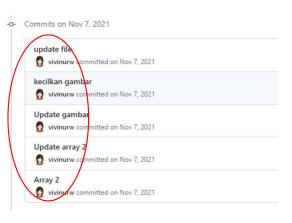
Commit adalah memastikan dan mencatat setiap penambahan kode program, setiap proses commit akan dicatat informasinya kapan, siapa, yang mana.



### GitHub - Kolaborasi



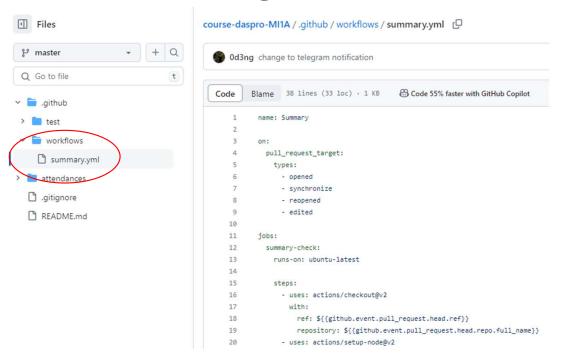


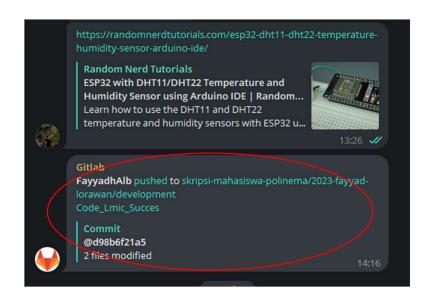


Banyak pengguna bisa memberikan kontribusi ke dalam proyek yang sama.



# GitHub - Integrasi CI/CD

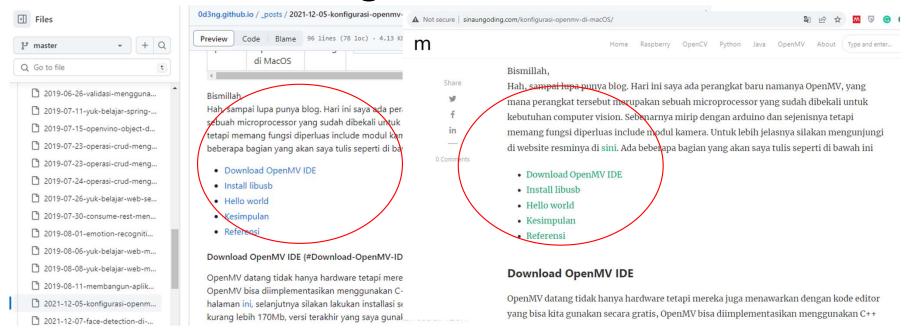




Contoh penggunaan **continue integration** yaitu ketika ada aksi pull request atau yang lain, disesuaikan kebutuhan kemudian akan melakukan merger ke master, selanjutnya memberikan notifikasi ke Telegram.



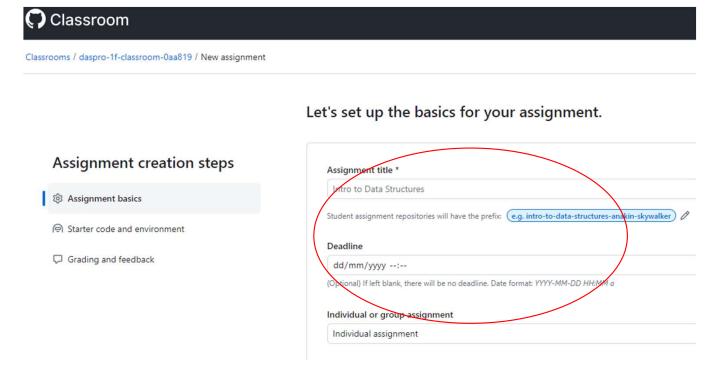
### GitHub - GitHub Pages



Kita bisa membuat sebuah **website statis** untuk kebutuhan **dokumentasi proyek, portofolio, atau halaman pribadi** di **GitHub** menggunakan **GitHub Pages**.



### GitHub - GitHub Classroom



GitHub Classroom memiliki fungsi seperti **Learning Management System (LMS)** untuk **mengelola kelas** seperti memberikan tugas, memberikan materi, melakukan penilaian, dan fungsi yang lain untuk pembelajaran.



### Kanban Board - Pendahuluan



- Ada yang pernah melihat benda seperti apa?
- Tujuannya buat apa?



### **Apa itu Kanban Board?**

- Kanban Board sebagai alat visual untuk melacak tugas dan pekerjaan.
- Kanban board memberikan cara yang sangat efektif untuk memvisualisasikan status tugas, mengatur alur kerja, dan meningkatkan produktivitas.
- Kanban board terdiri dari kolom-kolom yang mewakili tahapantahapan berbeda dalam alur kerja, serta kartu-kartu atau item yang mewakili tugas atau pekerjaan yang harus dilakukan.
- Setiap kartu dipindahkan secara visual dari satu kolom ke kolom berikutnya seiring dengan kemajuan tugas.
- Memberikan gambaran yang jelas tentang alur kerja, status tugas, dan mengidentifikasi hambatan atau masalah dengan cepat.





### Manfaat utama dari Kanban board

- Visualisasi alur kerja; memberikan gambaran visual yang jelas tentang alur kerja dan status tugas.
- Pengelolaan waktu dan prioritas; memungkinkan pengaturan prioritas dan mengontrol aliran tugas.
- Kolaborasi tim; membantu tim berkomunikasi tentang pekerjaan, bertanggung jawab, dan menyelesaikan masalah.
- Perbaikan berkelanjutan; memfasilitasi evaluasi dan perbaikan terusmenerus dalam alur kerja.
- Pengurangan pemborosan; mengurangi waktu tunggu dan kelebihan pekerjaan dalam alur kerja.
- Adaptasi terhadap perubahan; memungkinkan fleksibilitas dalam menghadapi perubahan dan prioritas baru.
- **Transparansi**; semua anggota tim dapat dengan mudah melihat status tugas dan progres.





### Struktur Kanban Board

- Backlog (To Do); berisi daftar tugas atau pekerjaan yang belum dimulai.
- In Progress; berisi tugas atau pekerjaan yang sedang aktif dikerjakan oleh tim.
- Review; tahap di mana tugas atau pekerjaan telah selesai dikerjakan dan sedang ditinjau atau direview oleh anggota tim lain atau pihak terkait.
- Testing/QA; berisi tugas atau pekerjaan yang telah melewati tahap review dan siap untuk diuji atau diverifikasi.
- **Done**; tugas atau pekerjaan yang **telah selesai** dikerjakan, diuji, dan disetujui dipindahkan ke tahap ini.

Struktur Kanban Board dapat disesuaikan dengan kebutuhan dan alur kerja spesifik tim atau proyek.







# Pengelolaan Waktu dan Prioritas – Kanban Board

- Tempatkan tugas-tugas yang memiliki prioritas tertinggi di bagian atas kolom "To Do" atau backlog.
- Tentukan batasan berapa banyak tugas yang boleh berada dalam tahap "In Progress" secara bersamaan.
- Dorong tim untuk **menyelesaikan satu tugas** sebelum memulai tugas baru.
- Prioritaskan ulang tugas berdasarkan perubahan kebutuhan atau situasi.
- Saat memilih tugas berikutnya untuk dikerjakan, pertimbangkan kompleksitas, durasi, dan prioritasnya.
- Gunakan kode warna atau label untuk menandai tugas berdasarkan prioritas atau jenis.





### Kolaborasi Tim – Kanban Board

- Gunakan Kanban Board sebagai titik pusat untuk melihat status pekerjaan seluruh tim.
- Ketika tugas ditambahkan ke papan Kanban, jelaskan **siapa** yang akan mengerjakan tugas tersebut.
- Gunakan fitur komentar pada kartu atau item Kanban untu berdiskusi tentang pekerjaan
- Jika tim Anda terlibat dalam pengembangan perangkat lunak, gunakan tahap "Review" atau kolom terpisah untuk **peninjauan kode**.
- Jika ada hambatan atau masalah yang mempengaruhi pekerjaan, anggota tim dapat menandai atau menambahkan isu terkait pada tugas.
- Pertemuan rutin tim untuk dapat melihat papan Kanban bersama dan berbicara tentang pekerjaan yang sedang dikerjakan.





### Software - Kanban Board

- Trello; menawarkan antarmuka yang intuitif dengan kartu yang dapat dipindahkan di antara kolom-kolom yang mewakili tahap alur kerja.
- **Notion**: sebagai alat pembuatan dokumen, juga memungkinkan membuat papan Kanban yang dapat diakses tim.
- Jira software; digunakan untuk pengembangan perangkat lunak, memiliki fitur Kanban Board yang kuat.



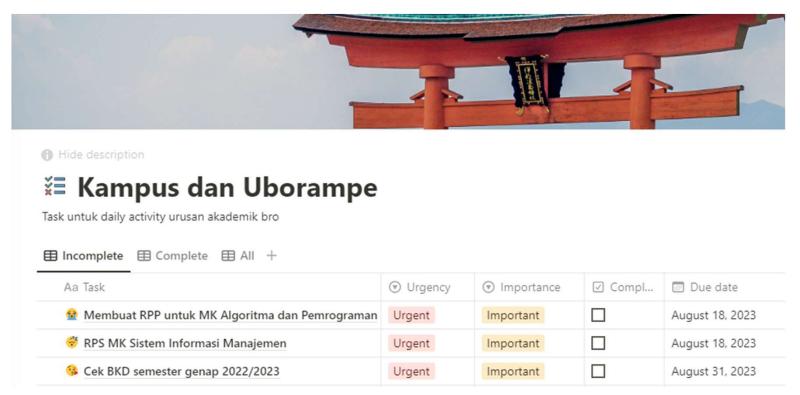


### Kanban Board - Trello





# **Kanban Board - Notion**



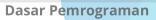
Kita dapat membuat prioritas dan batas waktu dari setiap tugas.













### Tugas di Kelas

• Silakan mencari 5 repository **open source** dengan jumlah **contributor**, **Star**, **dan commit** paling banyak!

• Buatlah **sebuah aktifitas sehari-hari** menggunakan kanban board **selama 4 minggu** (2 minggu ke belakang dan 2 minggu ke depan), minimal struktur

kanban terdiri dari To Do, In Progress, dan Done!

