PEMANFAATAN GITHUB DALAM PENGEMBANGAN WEBSITE MEDIA PEMBELAJARAN *PIAN* UNTUK MENDORONG MINAT ANAK DALAM DUNIA PEMROGRAMAN

Disusun Guna Memenuhi Tugas Mata Kuliah Pemrograman Dasar 1

Dosen Pengampu:

Dr. Harja Santanapurba, M.Kom

Novan Alkaf B. S., S.Kom., M.T

Ihdalhubbi Maulida, M.Kom

Disusun Oleh:

Kelompok 7

Muhammad Iqbal Ramadhan	(2410131210025)
Muhammad Syahbana	(2410131110002)
Nur Laily Azkiya	(2410131220033)

PEMROGRAMAN WEB 1/ABKC6205/A2

PROGRAM STUDI PENDIDIKAN KOMPUTER

FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN

UNIVERSITAS LAMBUNG MANGKURAT

BANJARMASIN

2025

DAFTAR ISI

PEMBAHASAN

A. Pemahaman Dasar Git

Git adalah salah satu tool penting yang sering digunakan dalam proyek pengembangan perangkat lunak, dan wajib dipahami oleh setiap programmer. Git merupakan sistem kontrol versi (Version Control System) yang diciptakan oleh Linus Torvalds, pendiri kernel Linux. Sebagai sistem kontrol versi, Git bertugas mencatat setiap perubahan yang terjadi pada file proyek, baik yang dikerjakan secara individu maupun berkelompok.

Git dikenal sebagai sistem distributed revision control (kontrol versi terdistribusi), yang berarti penyimpanan database Git tidak hanya berada di satu tempat, melainkan didistribusikan ke seluruh pengguna. Dengan demikian, setiap pengguna memiliki salinan lengkap dari seluruh riwayat versi proyek, yang membuat Git sangat andal dalam kolaborasi tim dan pemulihan data.

1) Konsep Umum Tentang Git

a. Repository

Repository Git adalah struktur penyimpanan yang memuat seluruh file proyek beserta riwayat perubahan atau versi dari file-file tersebut. Repository dapat bersifat lokal (disimpan di dalam komputer pengguna) maupun jarak jauh (di hosting pada platform seperti GitHub). Di dalam repository inilah semua proses manajemen versi, kolaborasi, dan pengembangan berlangsung.

b. Commit

Commit merupakan rekaman atau snapshot dari kondisi repository pada titik waktu tertentu. Setiap commit mencerminkan serangkaian perubahan yang telah dilakukan terhadap file di dalam repository, dan disertai dengan *commit message* yang berfungsi sebagai dokumentasi perubahan. Commit memudahkan pengembang untuk melacak perkembangan fitur atau perbaikan yang dilakukan.

c. Branch

Branch (Cabang) dalam git adalah jalur pengembangan independen yang memungkinkan pengembang untuk mengerjakan fitur baru,

perbaikan bug, datau eksperimen tanpa memengaruhi cabang utama (main atau master). Dengan memanfaatkan branch, proses pengembangan dapat dilakukan secara paralel dan lebih terorganisir. Setelah pekerjaan di cabang selesai, hasilnya dapat digabungkan kembali ke cabang utama.

d. Merge

Marge (Penggabungan) adalah proses integrasi perubahan dari satu cabang ke cabang lainnya. Biasanya dilakukan setelah pengembangan fitur atau perbaikan selesai pada cabang tertentu, dan ingin dimasukkan ke dalam cabang utama. Git secara otomatis akan menggabungkan perubahan jika tidak ada konflik namun, jika terjadi konflik (perbedaan isi file yang saling bertentangan), pengguna harus menyelesaikannya secara manual.

e. Clone

Kloning adalah proses menyalin repository yang ada (biasanya repository jarak jauh) ke dalam komputer lokal. Dengan melakukan kloning, pengguna mendapatkan salinan lengkap dari semua file proyek beserta riwayat versinya. Perintah *git clone* memungkinkan pengembang untuk mulai bekerja pada proyek yang sudah ada dengan cepat, tanpa perlu mengatur ulang struktur proyek dari awal.

2) Perintah Git Yang Sering Digunakan

a. Konfigurasi

- *git config –global user.name "Nama Anda"* : Menetapkan nama Anda untuk komitmen Git
- *git config –global user.email "yourname@explane.com"* : Mengatur email Anda untuk komitmen Git.

b. Inisialisasi dan Kloning Repository

- *git init* : Menginisialisasi repository Git baru.
- *git clone <repository_url>* : Menyalin (mengkloning) repository yang ada ke komputer lokal.

c. Alur Kerja Dasar

- *git add* : Menambahkan file yang dimodifikasi ke dalam *area staging*, yaitu tahap persiapan sebelum commit.
- *git commit -m "Pesan commit"* : Menyimpan perubahan dari *staging area* kedalam repository dengan pesan deskriptif.
- *git push <remote> <branch>* : Mengirimkan perubahan yang telah di commit dari repository lokal ke GirHub.
- *git pull <remote> <branch>* : Mengambil (menarik) pembaruan dari repository GitHub (online) dan menggabungkan ke cabang lokal saat ini.

d. Percabangan dan Penggabungan

- *git branch <branch_name>* : Membuat cabang baru.
- *git checkout <branch_name>* : beralih ke cabang yang ditentukan

e. Repository Jarak Jauh (online)

- *git remote add origin <url_repository>* : Menambahkan repository online

f. Berkolaborasi

- *git clone <forked_repository_url>* : Mengkloning repository bercabang ke lokal.

B. Pengenalan Platform GitHub

GitHub adalah platform berbasis web yang digunakan untuk mengelola proyek perangkat lunak menggunakan Git. Selain sebagai sistem versioning, GitHub juga berfungsi sebagai alat kolaborasi tim dan jejaring sosial bagi para pengembang dari seluruh dunia. Melalui GitHub, pengguna dapat menyimpan kode, melacak perubahan, bekerja sama dalam satu proyek, serta berkontribusi pada proyek lain melalui fitur seperti *repository*, *pull request*, dan *issue*. GitHub memudahkan pengelolaan proyek secara terbuka, terstruktur, dan kolaboratif. Fitur utama Github dalam pengembangan perangkat lunak:

1. Repository adalah tempat menyimpan seluruh file proyek dan riwayat perubahan.

2. Branch fungsinya untuk memungkinkan pengembangan fitur secara terpisah tanpa mengganggu kode utama.
3. Pull Request (PR) digunakan untuk mengusulkan perubahan dari satu branch ke branch lain dan mendiskusikannya sebelum digabung.
4. Fork untuk menyalin repository milik orang lain ke akun sendiri untuk dikembangkan lebih lanjut.
5. Issues adalah sistem pelaporan masalah (bug), permintaan fitur, atau diskusi terkait proyek.
6. Actions (GitHub Actions) merupakan alat otomatisasi seperti pengujian kode, build, dan deployment secara otomatis.
7. Projects adalah Papan manajemen tugas (mirip Trello) untuk merencanakan dan memantau perkembangan proyek.
8. Wiki yaitu dokumentasi proyek dalam bentuk halaman wiki yang bisa diperbarui secara kolaboratif.
9. GitHub Pages adalah layanan hosting gratis untuk membuat dan menampilkan halaman web langsung dari repository.
10. Security & Insights menyediakan analisis kontribusi, pelacakan kerentanan keamanan, dan laporan riwayat aktivitas proyek.

Git dan GitHub merupakan dua entitas berbeda yang saling melengkapi. Git bekerja secara lokal untuk mengelola versi proyek, sedangkan GitHub berperan sebagai platform daring untuk menyimpan, mengelola, dan berkolaborasi dalam proyek.

C. Panduan Kolaborasi Proyek Menggunakan GitHub

Dalam pengembangan perangkat lunak berbasis tim, kemampuan untuk bekerja secara kolaboratif merupakan faktor penting yang menentukan keberhasilan sebuah proyek. GitHub, sebagai platform manajemen kode berbasis Git, menyediakan berbagai fitur yang memfasilitasi kerja sama antaranggota tim secara efisien dan terstruktur. Kolaborasi proyek melalui GitHub memungkinkan setiap anggota tim untuk berkontribusi terhadap kode sumber, melacak perubahan, mendiskusikan ide, serta mengelola tugas secara sistematis.

1) Langkah awal dalam kolaborasi proyek adalah membuat repository utama oleh salah satu anggota tim, biasanya ketua kelompok. Repository ini berfungsi sebagai pusat pengembangan proyek. Setelah repository berhasil dibuat,

anggota lain dapat ditambahkan sebagai kolaborator melalui pengaturan yang tersedia pada menu Settings di GitHub. Dengan ditambahkan sebagai kolaborator, anggota memiliki hak akses untuk membaca dan menulis terhadap isi repository.

2) Setiap anggota tim kemudian mengkloning repository tersebut ke komputer lokal mereka menggunakan perintah git clone. Proses ini memungkinkan anggota untuk bekerja secara lokal tanpa memengaruhi file utama yang berada di GitHub. Untuk menghindari konflik dalam penulisan kode, setiap anggota dianjurkan untuk membuat *branch* (cabang) baru yang merepresentasikan fitur atau tugas yang sedang dikerjakan. Dengan menggunakan branch, proses pengembangan dapat dilakukan secara paralel tanpa mengganggu cabang utama (*main* atau *master*).
3) Setelah melakukan perubahan kode di dalam branch masing-masing, anggota tim dapat menyimpan perubahan tersebut menggunakan perintah git commit disertai dengan pesan yang menjelaskan apa saja yang telah diperbarui. Kemudian, branch yang telah diubah tersebut di-*push* ke repository GitHub. Langkah selanjutnya adalah membuat *pull request* (PR), yaitu usulan penggabungan perubahan dari satu branch ke branch utama. PR ini dapat ditinjau oleh anggota tim lain, yang berfungsi sebagai proses kontrol kualitas dan diskusi terhadap perubahan yang diusulkan.
4) Jika pull request disetujui, maka branch tersebut akan digabungkan ke branch utama. Namun, dalam beberapa kasus, penggabungan ini dapat menimbulkan konflik apabila terdapat perubahan yang bertabrakan di bagian kode yang sama. GitHub akan menandai konflik tersebut, dan anggota yang bersangkutan harus menyelesaikannya secara manual sebelum proses *merge* dapat dilanjutkan.
5) Untuk menjaga agar seluruh anggota tim tetap sinkron dengan perkembangan proyek, mereka harus secara rutin menarik pembaruan terbaru dari branch utama menggunakan perintah git pull. Hal ini penting dilakukan agar tidak terjadi ketidaksesuaian versi yang dapat menyebabkan *error* dalam pengembangan.
6) Selama proses kolaborasi, anggota tim juga dapat memanfaatkan fitur-fitur tambahan yang disediakan oleh GitHub, seperti *Issues* untuk pelaporan bug dan permintaan fitur, serta *Projects* untuk pengelolaan tugas secara visual.

Dengan demikian, GitHub tidak hanya berfungsi sebagai tempat penyimpanan kode, tetapi juga sebagai alat manajemen proyek yang mendukung pengembangan perangkat lunak secara kolaboratif, transparan, dan terstruktur.

KESIMPULAN

Pemanfaatan GitHub dalam pengembangan website media pembelajaran PIAN *(Platform Interaktif Anak Ngoding)* terbukti memberikan kontribusi positif dalam mendukung kolaborasi tim dan pengelolaan kode sumber secara efektif. Melalui penggunaan Git sebagai sistem kontrol versi, setiap perubahan pada proyek dapat dicatat, dikelola, dan dilacak dengan baik, sehingga mempermudah proses pengembangan dan pemeliharaan kode. Sementara itu, GitHub sebagai platform berbasis web menyediakan berbagai fitur yang mendukung kerja sama tim, seperti repository jarak jauh, branch, pull request, dan issue tracker, yang membantu menjaga kualitas kode sekaligus mempercepat penyelesaian tugas secara paralel.

Dalam proyek ini, GitHub berperan penting sebagai alat manajemen proyek yang mendorong terciptanya kolaboratif, transparan, dan terstruktur. Penggunaan GitHub juga mendorong mahasiswa untuk lebih memahami praktik kerja tim dalam pengembangan perangkat lunak modern, sekaligus menumbuhkan minat anak-anak dalam dunia pemrograman melalui media pembelajaran PIAN. Dengan demikian, penerapan Git dan GitHub tidak hanya meningkatkan efisiensi pengembangan, tetapi juga menjadi sarana edukatif untuk memperkenalkan teknologi pengelolaan proyek perangkat lunak kepada generasi muda.

DAFTAR PUSTAKA

Hosting, R. J. (2025, February 25). Apa Itu Git? Ini Fungsi, Fitur, dan Manfaatnya. *Blog Jagoan Hosting*. https://www.jagoanhosting.com/blog/git-adalah/

Mega. (2022, September 14). *GIT: Pengertian, Fitur dan Manfaatnya*. Bikin.Website. https://bikin.website/blog/pengertian-git/

Verma, A. (2023, August 17). Git & GitHub Concepts - Abhijeet Verma - Medium. *Medium*. https://medium.com/@abhijeetv007/git-github-concepts-e233adf17dba