**Apa Itu GitHub?**

GitHub adalah situs web dan layanan berbasis cloud yang memungkinkan pengembang untuk menyimpan dan mengelola kode open source dan dokumen mereka serta mengontrol perubahannya.   
GitHub juga merupakan salah satu repositori online terbesar untuk kolaborasi di dunia. Dengan platform ini, kita dapat merencanakan proyek yang kita buat dengan rekan kerja dari seluruh dunia. GitHub memiliki dua prinsip utama: Git dan kontrol versi

**Apa Itu Git?**

Git adalah sistem kontrol versi terdistribusi. Ini berarti bahwa seluruh basis kode dan riwayat kode open source tersedia di setiap mesin pengembang, membuat percabangan dan penggabungan menjadi mudah. Dikembangkan oleh Linus Torvalds pada tahun 2005, Git adalah inti atau jantung dari GitHub

**Apa Itu Hub?**  
Jika Git adalah jantungnya, maka hub adalah jiwanya GitHub. Sistem GitHub dirancang untuk membuat baris perintah seperti Git menjadi jaringan media sosial terbesar bagi para pengembang.  
Selain berkontribusi pada proyek tertentu, GitHub memungkinkan pengguna untuk berkomunikasi dengan orang-orang yang memiliki visi dan misi yang sama. kemudian kita dapat mengikuti mereka untuk melihat proyek yang sedang mereka kerjakan dan bahkan mencari tahu dengan siapa mereka terhubung.

**Apa Itu Version Control System?**  
  
Sistem kontrol versi adalah sistem yang merekam perubahan pada file, mencatat seluruh riwayat, dan memungkinkan kita untuk melihatnya lagi nanti. Kode open source terus diperbarui saat pengembang membuat proyek baru. Bahkan setelah proyek online, pengembang masih perlu memperbarui versi, memperbaiki bug, menambahkan fitur baru, dan banyak lagi.   
Sistem kontrol versi membantu pengembang melacak perubahan yang mereka buat pada basis kode mereka. Selain itu, sistem  mencatat siapa  yang membuat perubahan dan memulihkan kode yang  dihapus atau diubah.  
  
**Repositori**  
Repositori adalah direktori penyimpanan untuk file proyek. kita juga dapat menyimpan semua yang terkait dengan proyek yang di buat di sini. Sebagai contoh: Kode, gambar, atau file audio. Repositori itu sendiri dapat berada di repositori GitHub atau repositori lokal yang ada di dalam folder.  
  
  
**Branch**  
Branch merupakan salinan dari repositori. dapat menggunakan branch saat melakukan  pengembangan atau pengembangan individu. Pekerjaan atau tugas yang di lakukan di branch tidak  memengaruhi repositori pusat atau branch lainnya. Saat pengembangan selesai, kita dapat menarik menggabungkan branch saat ini ke branch lain dan repositori pusat.  
  
**Pull Requests**  
Pull Requests Dalam hal ini, ini memberi tahu pengguna bahwa mereka telah mendorong perubahan yang mereka buat ke cabang ke repositori master. Kolaborator repositori menerima atau menolak permintaan tarik. Saat menerima permintaan tarik, kita dapat mendiskusikan dan mengonfirmasi proyek dengan kolaborator.  
Berikut beberapa langkah untuk membuat pull requests di GitHub:  
1. Masuk ke repositori dan cari menu branch.  
2. Di menu branch, pilih branch yang menyimpan commit.  
3. Klik opsi New pull requests yang ada di samping menu branch.  
4. Masukkan judul dan deskripsi pull requests.  
5. Klik opsi Create pull requests.  
  
  
**Forking Repository**  
Forking repositori berarti membuat proyek baru berdasarkan repositori yang sudah ada. Sederhananya, forking repositori berarti  menyalin repositori yang ada, membuat perubahan yang diperlukan, kemudian menyimpan versi terbaru sebagai repositori baru dan menjadikannya proyek sendiri. Fitur ini meningkatkan pengembangan proyek. Karena proyek fork baru, tidak ada yang terjadi di repositori pusat. kita juga dapat menerapkan perubahan ke repositori master fork saat ini.  
Berikut dua langkah untuk forking repositori GitHub:  
1. Cari repositori yang ingin di forking.  
2. Klik opsi Fork.

**Fungsi Github**

Sekarang setelah mengetahui apa itu GitHub, Berikut adalah beberapa fitur GitHub yang berguna untuk pengembang:  
1. Memudahkan pengerjaan project

Pertama, fitur GitHub adalah untuk memfasilitasi kolaborasi pada proyek yang sedang berjalan. Ini karena GitHub telah mendistribusikan kontrol versi yang memungkinkan pengembang atau anggota tim untuk mengakses dan mengelola kode mereka di satu tempat.  
2. Mencegah perubahan kode yang justru dapat merusak kode asli  
  
GitHub adalah platform dengan kemampuan percabangan yang memungkinkan membuat banyak cabang dari kode utama proyek . Oleh karena itu, perubahan kode tidak mempengaruhi kode utama. Ini sangat berguna untuk pemecahan masalah atau menambahkan fitur baru.  
  
3. Bisa jadi portofolio seorang developer  
  
GitHub tidak hanya membuat proyek lebih mudah untuk dikerjakan, tetapi juga bertindak sebagai portofolio pengembang profesional. Portofolio ini dapat menjadi daya tarik tersendiri bagi klien dan bisnis yang membutuhkan jasa sebagai developer.

**Fitur-fitur Github**  
1. Pengkodean Kolaboratif  
  
GitHub memiliki kemampuan untuk membantu tim pengembangan lainnya mengerjakan proyek dengan cepat melalui lingkungan pengaturan otomatis. kita dapat menggunakan tool ini untuk melihat perubahan kode, memeriksa kode baru, atau meminta tim untuk meninjau permintaan.  
  
2. Otomatisasi dan CI/CD  
  
kita dan orang lain juga dapat menggunakan GitHub actions untuk mengotomatisasi berbagai tugas seperti CI/CD, pengujian, perencanaan, dan manajemen proyek.  
  
3. Keamanan  
  
Kita dapat menggunakan GitHub actions untuk secara otomatis memeriksa perubahan pada basis kode dan mengidentifikasi kerentanan yang ada sebelum pindah ke produksi. kita dapat menggunakan pemindaian kode untuk menemukan kerentanan dan bug dan memprioritaskan perbaikan masalah tersebut. kita juga dapat menjadwalkan pemindaian rutin untuk menghindari masalah baru.  
  
4. Manajemen Proyek  
  
GitHub memiliki alat manajemen proyek untuk membantu mengatur dan memprioritaskan tugas pengembangan perangkat lunak. Ini berisi papan proyek yang berisi masalah, permintaan tarik, dan catatan penting. Selain itu, GitHub actions menyediakan tonggak untuk menyelesaikan masalah di repositori atau melacak kemajuan permintaan tarik. Dengan berbagai fitur, GitHub adalah platform untuk memfasilitasi kolaborasi dan komunikasi antar pengembang.  
  
5. Administrasi Tim  
  
Untuk menyederhanakan pengelolaan akses dan izin antar proyek, GitHub menyediakan kemampuan manajemen tim. Fitur ini memungkinkan untuk memperbarui izin, menambahkan pengguna baru, dan memberikan izin, tergantung pada peran pengembang di perusahaan. Jika menggunakan GitHub, kita juga dapat menggunakan log audit. Ini memungkinkan administrator untuk melihat tindakan yang diambil oleh tim pengembangan organisasi.  
  
**Keuntungan Menggunakan Github**

Github tidak hanya menjadi wadah bagi para programmer untuk berkolaborasi, tetapi juga memberikan berbagai kemudahan yang bermanfaat dalam hal pengembangan perangkat lunak. Saat menggunakan platform, kita dapat langsung menggunakan setidaknya tiga manfaat atau manfaat.  
  
1. Monitoring Setiap Perubahan yang Terjadi dalam Program  
Keuntungan  pertama terkait erat dengan penggunaan kontrol versi. Ini bukan hanya tentang menjamin sistem keamanan. Menurut situs web The Balance Careers, Github memiliki sistem pencatatan perubahan yang menunjukkan aktivitas dan perubahan program.   
Lebih sederhana, dapat disamakan dengan penggunaan Google Docs dalam sebuah tim. Hal ini memudahkan untuk melihat secara real time siapa yang membuat perubahan pada aktivitas yang berbeda.  
2. Sebagai Tempat untuk Kolaborasi dengan Tim  
Pada dasarnya, untuk mengerjakan sebuah proyek yang membutuhkan kerja sama tim, maka perlu adanya tools atau teknologi khusus untuk membantu proses integrasi sistem menjadi lebih efektif. Nah, Github menjadi salah satu solusi untuk menyelesaikan permasalahan tersebut.   
Aplikasi ini sangat mampu untuk menunjang kebutuhan dalam kolaborasi antar tiap tim atau divisi. Salah satu fitur yang sering dipakai adalah terkait permintaan review. Fitur ini banyak dimanfaatkan untuk kebutuhan dalam meminta saran atau masukan (feedback) kepada atasan maupun rekan kerja.  
3. Untuk Menunjang Portofolio  
Manfaat Utama: Github adalah media yang  sangat efisien untuk membuat portofolio yang memenuhi kebutuhan karir. Lebih mudah untuk mendokumentasikan semua aktivitas terkait pemrograman, pengetahuan penulisan kode pemrograman (kode sumber), dan pembuatan repositori.

**Istilah Penting dalam Github**

Setelah mempelajari beberapa hal terkait penggunaan Github sendiri, masukkan deskripsi istilah penting untuk aplikasi.  
- Repo / Repository: Tempat atau folder untuk meletakkan project.  
- Commit: Fitur rekaman atau snapshot dari repository (menyimpan riwayat perubahan).  
- Checkout: Melakukan perpindahan ke dalam perubahan tertentu.  
- Hash: Penanda unik pada commit.  
- Merge: Menggabungkan dua komponen branch atau lebih.  
- Branch: Suatu cabang dari perubahan.  
- Remote: Sumber daya (resource) yang mempunyai repository.  
- Push: Mengirim commit menuju repository.  
- Pull: Mengambil sebuah commit dari repository.  
- Clone: Mengambil repository dari sebuah remote

**Cara Menggunakan Github**

Meskipun populer dan  penyimpanan data  terbesar di dunia, kami menemukan bahwa banyak pengembang masih tidak tahu cara  menggunakan GitHub.

**Inilah cara menggunakan GitHub**

1. Login terlebih dahulu   
  
Langkah pertama  yang perlu di lakukan adalah mendaftar di situs GitHub.   
Sebagai aturan umum, data pribadi akan diminta pada awal pendaftaran ini dan harus diunggah sepenuhnya.   
Buat ID dan kata sandi berikut untuk  login.  
  
2. Membuat repository atau tempat penyimpanan   
  
Setelah langkah pertama  dilakukan,  langkah kedua adalah membuat area yang disebut area penyimpanan, atau sering juga repositori. Untuk langsung masuk  ke akun GitHub lalu pilih menu repositori.   
Kemudian klik menu New untuk menampilkan halaman lembar kerja baru. Buat pengaturan yang  sesuai pada lembar kerja dan klik menu Create Repository untuk menyelesaikan prosesnya.  
  
3. Membuat folder pada Windows   
  
Selanjutnya, perlu membuat folder khusus  di Windows. Tujuan dari build ini adalah untuk mengintegrasikan repositori situs GitHub langsung ke perangkat sehingga dapat membukanya langsung dari komputer .   
Juga sangat mudah untuk membuka data, cukup klik kanan pada folder yang sudah di buat dan pilih Git Bash Here, dan integrasi data akan dilakukan secara otomatis  secara digital.  
  
4. Memasukkan file dalam repository   
  
Langkah selanjutnya adalah mengisi repositori tempat ia dibuat. Jangan lupa klik kanan dan pilih menu GitBash Here seperti  sebelumnya.  
  
5. Lakukan push GitHub   
  
Melakukan push ke GitHub adalah langkah terakhir yang bertujuan untuk dapat menyimpan semua proses ke situs GitHub. Jangan lupa untuk mengecek kembali apakah data yang di masukkan sudah lengkap dan masih dalam proses. Tidak hanya itu, kita juga perlu menghapus file yang tidak perlu di sistem.

**Apakah Github Hanya Bisa Dipakai oleh Developer?**

GitHub terutama ditujukan untuk digunakan oleh pengembang. Namun, pertanyaannya, apakah GitHub hanya digunakan oleh developer? Jawabannya adalah tidak. GitHub adalah platform yang dapat digunakan  siapa saja. Siapa pun dapat menggunakan GitHub untuk mengelola proyek mereka dan berkolaborasi dalam tugas.   
Jika saat ini kita sedang mengerjakan  proyek dengan rekan kerja, boleh saja menggunakan GitHub untuk manajemen proyek. Dengan platform ini, kita dapat mengerjakan proyek yang sedang di kerjakan, memperbarui proyek, dan melacak semua perubahan.

**Kesimpulan**

- Github adalah platform yang banyak digunakan untuk  kolaborasi antara pengembang individu untuk membuat perangkat lunak dengan cara yang sistematis dan fleksibel.   
- Keuntungan menggunakan platform ini adalah untuk memantau semua perubahan yang dilakukan, tidak hanya sebagai tempat untuk menyimpan hasil portofolio di bidang pengembangan perangkat lunak, tetapi juga untuk efisiensi kerja kolaborasi pengembang. Membuat akun Github   
- sangat mudah. bisa langsung masuk ke situs resminya dan mulai  mengunggah berbagai dokumen proyek yang terkait dengan kode program aplikasi.