



# Pengenalan Git pada Version Control System

Git adalah sistem kontrol versi terdistribusi yang populer. Ia membantu pengembang melacak perubahan kode. Git memudahkan kolaborasi dalam tim pengembangan perangkat lunak.

 by 11

# Apa itu Version Control System?

## Pelacakan Perubahan

VCS mencatat setiap perubahan pada file dalam proyek perangkat lunak.

## Kolaborasi Tim

Memungkinkan beberapa pengembang bekerja pada proyek yang sama secara bersamaan.

## Manajemen Versi

Menyimpan berbagai versi proyek, memungkinkan akses ke versi sebelumnya.





# Sejarah Singkat Git

1

2005

Linus Torvalds memulai pengembangan Git untuk Linux kernel.

2

2007

Git mulai mendapatkan popularitas di kalangan pengembang.

3

2008

GitHub diluncurkan, meningkatkan adopsi Git secara global.



## Branching dan Merging

Git memungkinkan pembuatan cabang dan penggabungan kode dengan mudah.

## 2 Distributed

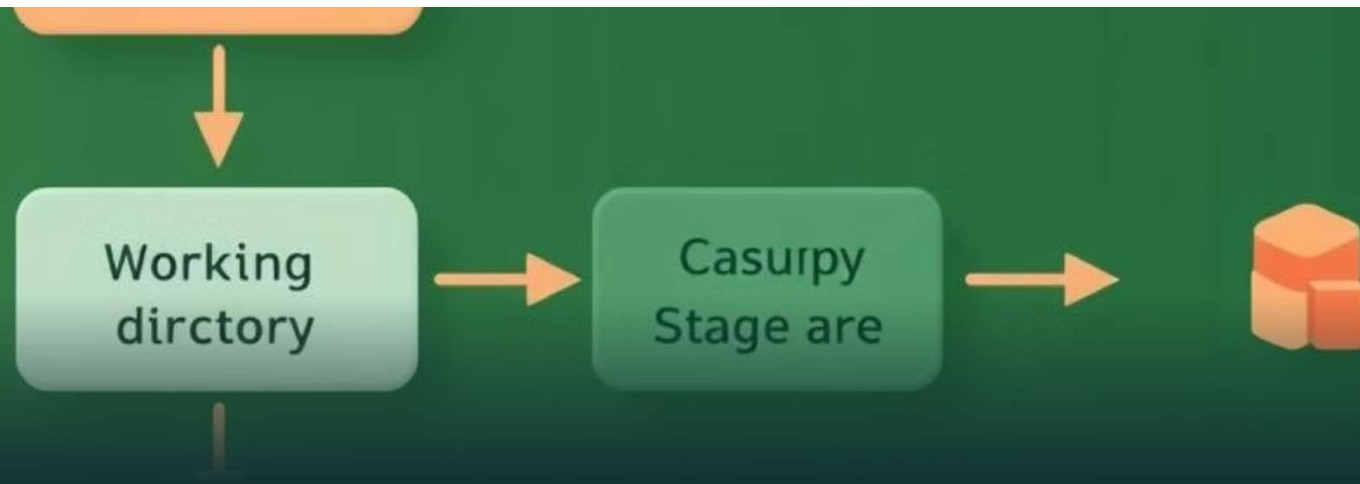
Setiap pengembang memiliki salinan lengkap repositori, termasuk riwayatnya.

### 3 Kecepatan

Git dirancang untuk kinerja yang cepat, bahkan untuk proyek besar.

## 4 Integritas Data

Git menggunakan checksums untuk memastikan integritas data.



## Konsep Dasar Git

1

### Working Directory

Area kerja lokal di mana Anda melakukan perubahan pada file.

2

### Staging Area

Area persiapan untuk perubahan yang akan di-commit.

3

### Repository

Database yang menyimpan semua versi proyek Anda.

# Perintah Dasar Git



git init

Membuat repositori Git baru

git clone

Menyalin repositori yang ada

git add

Menambahkan file ke staging area

git commit

Menyimpan perubahan ke repositori

git push

Mengirim perubahan ke repositori remote

# Branching dalam Git

## Apa itu Branch?

Branch adalah jalur pengembangan terpisah dalam proyek. Ini memungkinkan pengembangan fitur baru tanpa mengganggu kode utama.

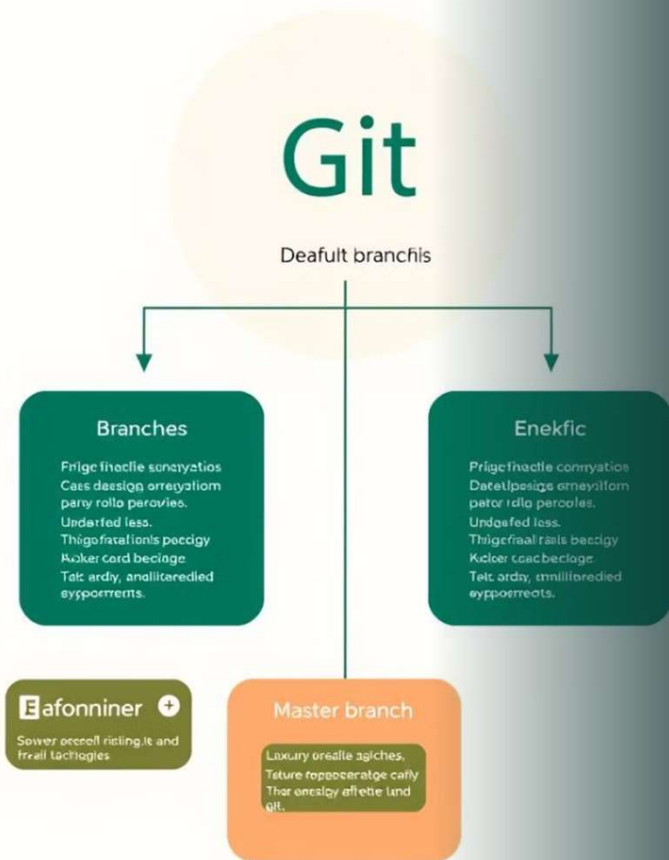
## Manfaat Branching

Memudahkan eksperimen, pengembangan fitur paralel, dan manajemen rilis. Meningkatkan kolaborasi tim dan kualitas kode.

## Perintah Branch

git branch: Membuat branch baru.  
git checkout: Beralih antar branch.  
git merge: Menggabungkan branch.

# Git Workflow



1

Buat Branch Fitur

Mulai pengembangan fitur baru di branch terpisah.

2

Commit Perubahan

Lakukan commit secara teratur selama pengembangan.

3

Pull Request

Ajukan pull request untuk review kode.

4

Merge ke Master

Setelah disetujui, gabungkan fitur ke branch utama.



# Git Hosting Platforms



GitHub

Platform hosting Git paling populer, dimiliki oleh Microsoft.



GitLab

Platform open-source dengan fitur CI/CD terintegrasi.



Bitbucket

Solusi Git hosting dari Atlassian, terintegrasi dengan Jira.

# Manfaat Menggunakan Git

## Kolaborasi Efektif

Tim dapat bekerja pada proyek yang sama tanpa konflik.

## Pelacakan Perubahan

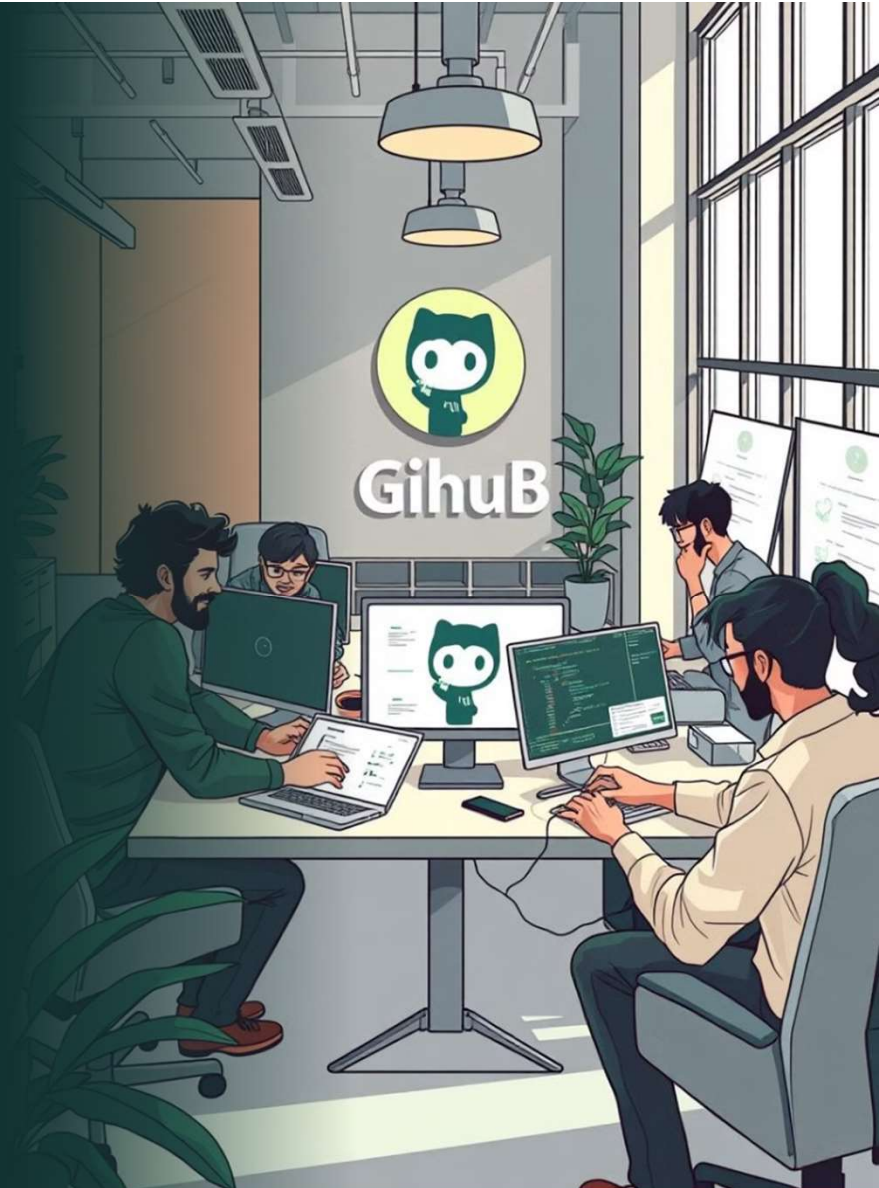
Setiap perubahan tercatat, memudahkan pemantauan dan pemulihan.

## Eksperimen Aman

Branching memungkinkan eksperimen tanpa risiko pada kode utama.

## Backup Otomatis

Repositori remote berfungsi sebagai backup proyek Anda.



# Tutorial GitHub



# Apa itu GitHub?

## 1 Platform Kontrol Versi

GitHub adalah platform hosting untuk kontrol versi dan kolaborasi menggunakan Git.

## 2 Repositori Kode

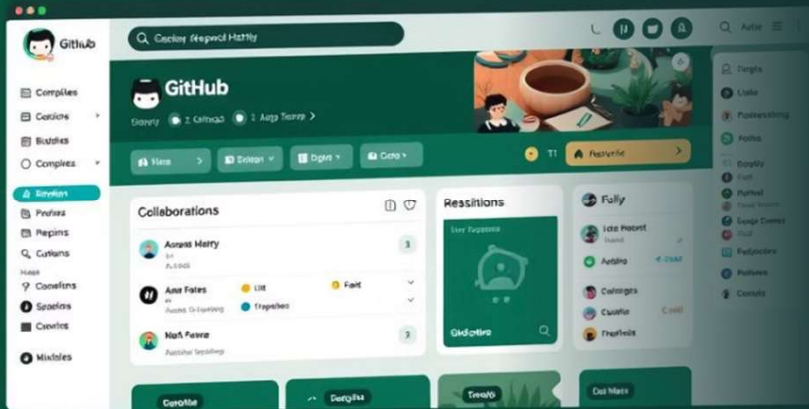
Memungkinkan pengembang menyimpan dan mengelola kode mereka secara online.

## 3 Kolaborasi Tim

Mempermudah kerja sama dalam proyek pengembangan perangkat lunak.

## 4 Jejak Perubahan

Melacak setiap perubahan kode dengan riwayat commit yang terperinci.



# Manfaat Menggunakan GitHub

## Kontrol Versi

Lacak dan kelola perubahan kode dengan mudah. Kembalikan ke versi sebelumnya jika diperlukan.

## Kolaborasi

Bekerja sama dengan tim dari mana saja. Bagi tugas dan gabungkan kontribusi dengan lancar.

## Portofolio

Tampilkan proyek Anda kepada dunia. Bangun reputasi di komunitas pengembang global.

# Membuat Akun GitHub

1

## Kunjungi Situs GitHub

Buka browser dan akses github.com. Klik tombol "Sign Up" di pojok kanan atas.

2

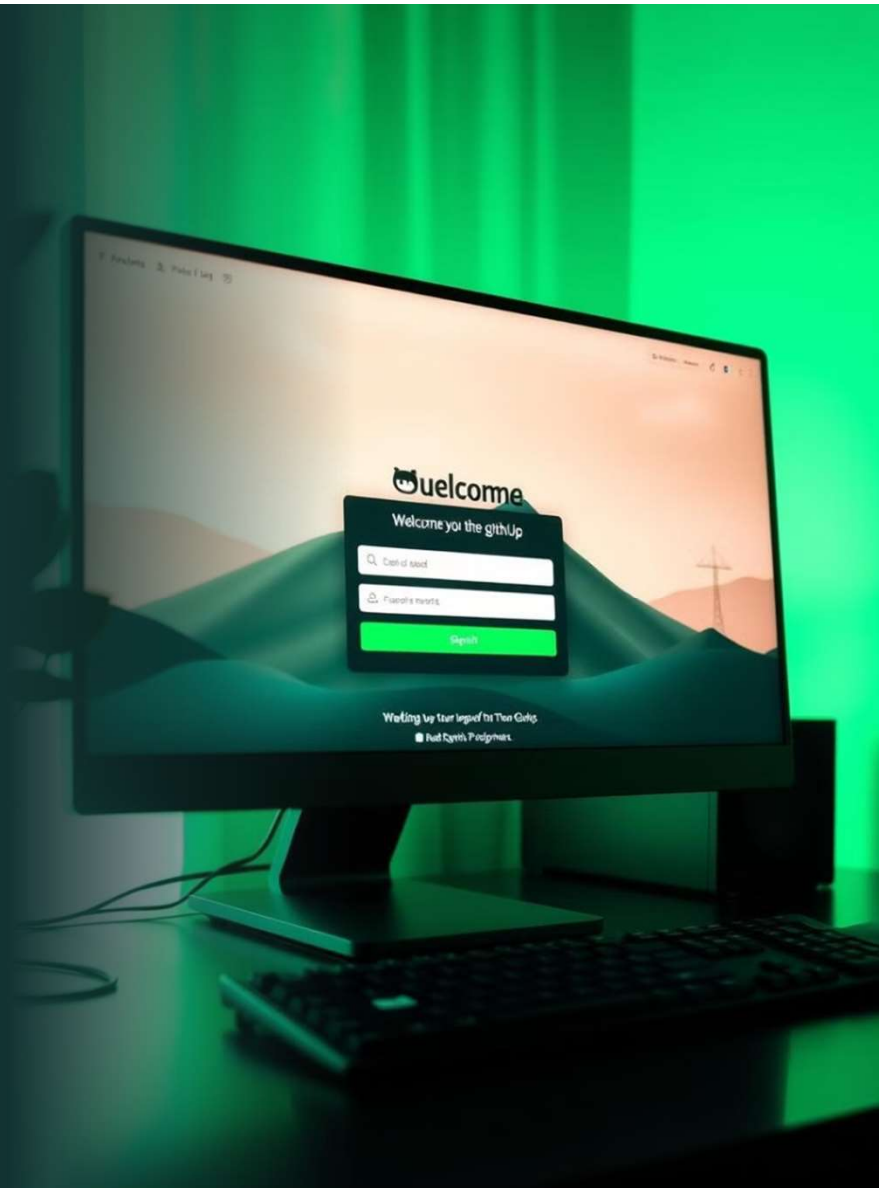
## Isi Formulir Pendaftaran

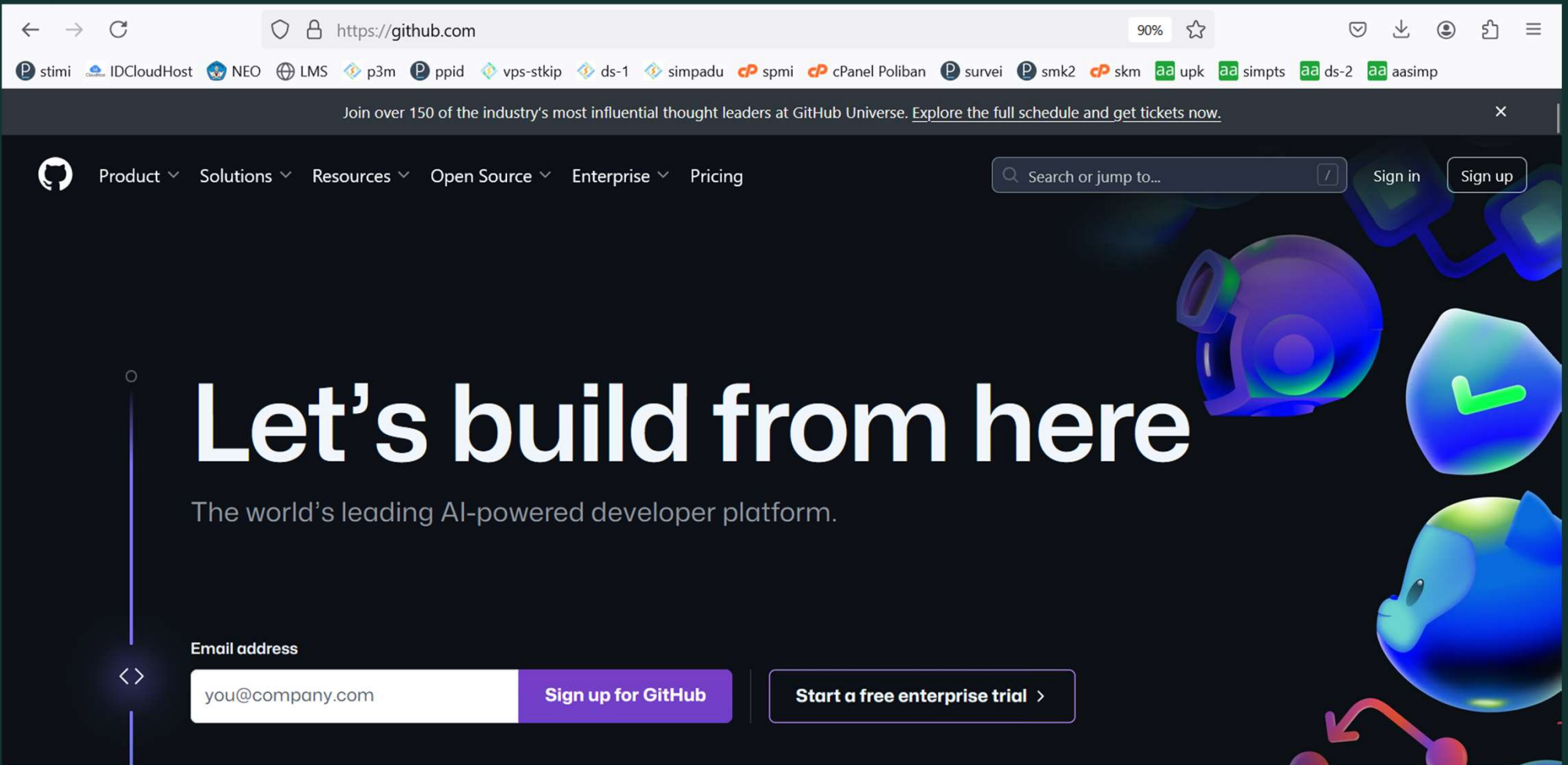
Masukkan alamat email, buat kata sandi, dan pilih nama pengguna yang unik.

3

## Verifikasi Akun

Periksa email Anda untuk tautan verifikasi. Klik tautan untuk mengaktifkan akun.







Join GitHub · GitHub

github.com/signup?ref\_cta=Sign+up&ref\_loc=header+logged+out&ref\_page=%2F&source=header-ho...

Already have an account?

Welcome to GitHub!  
Let's begin the adventure

Enter your email  
✓ email-nim-kalian@gmail.com

Create a password  
- [password] [eye icon] [Continue]

Password is strong  
\* Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.

## The tools you need to build what you want.

Soup to nuts, GitHub has it all.

### What specific features are you interested in using?

Select all that apply so we can point you to the right GitHub plan.

- ☒ Collaborative coding  
Codespaces, Pull requests, Notifications, Code review, Code review assignments, Code owners, Draft pull requests, Protected branches, and more.
- ☐ Automation and CI/CD  
Actions, Packages, APIs, GitHub Pages, GitHub Marketplace, Webhooks, Hosted runners, Self-hosted runners, Secrets management, and more.
- ☐ Security  
Private repos, 2FA, Required reviews, Required status checks.

Search or jump to...

Pull requests Issues Marketplace Explore

Create your first project  
Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.  
[Create repository] [Import repository]

Recent activity  
When you take actions across GitHub, we'll provide links to that activity here.

All activity  
Discover interesting projects and people to populate your personal news feed.  
Your news feed helps you keep up with recent activity on repositories you watch or star and people you follow.  
[Explore GitHub]

Universe 2021  
Missed any Universe sessions? All content is available on demand now so you can view on your own time.  
[Watch the sessions on demand]

🔍 The feed shows you events from people you follow and repositories you watch or star.  
🔔 Subscribe to your news feed

© 2021 GitHub, Inc. | Blog | About | All | Terms | Privacy | Status | Security | Contact | GitHub | Pricing



## Welcome to GitHub

We are glad you're here.

### How many team members will be working with you?

This will help us guide you to the tools that are best suited for your projects.

- Just me 2 - 5 5 - 10
- 10 - 20 20 - 50 50+

### Are you a student or teacher?

Student Teacher

[Continue]



## Learn to ship software like a pro.

GitHub gives students free access to the best developer tools so they can learn by doing.

### Free

- Unlimited public/private repositories
- 2,000 Actions minutes/month  
Free for public repositories
- 500MB of Packages storage  
Free for public repositories
- Community support

### Get additional student benefits

#### GitHub Pro

- Protect your branches  
Ensure that collaborators on your repository cannot make irreversible changes to branches.
- Draft pull requests
- Pages and Wikis
- 3,000 CI/CD minutes/month  
Free for public repositories
- 2GB of Packages storage  
Free for public repositories
- Web-based support

#### GitHub Student Developer Pack

- Free access to the industry's best developer tools  
Hundreds of others, including Digital Ocean, Microsoft Azure, PaaS, MongoDB, Cloudinary, Twilio, and Stripe.

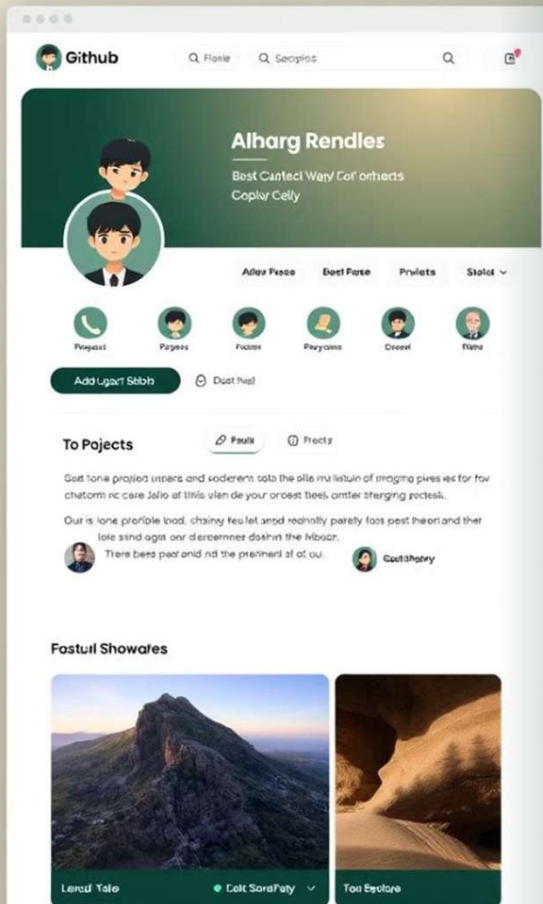
#### GitHub Campus Expert training

- Enrich your college technical community  
Learn the skills to build diverse tech communities on campus with training, mentorship, and support from GitHub.

[Apply for your GitHub student benefits]



# Mengatur Profil GitHub



## Unggah Foto Profil

Pilih avatar yang mewakili Anda. Foto profesional atau logo kreatif bisa menjadi pilihan bagus.

## Tulis Bio

Jelaskan singkat tentang diri Anda. Sebutkan keahlian dan minat dalam pengembangan perangkat lunak.

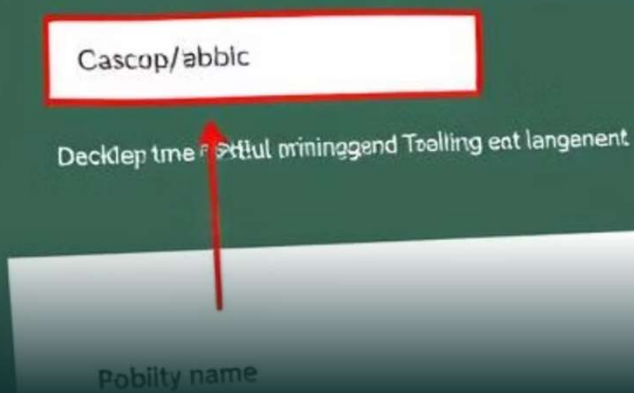
## Tambahkan Informasi Kontak

Sertakan tautan ke situs web atau profil media sosial profesional Anda.

## Pin Repositori Unggulan

Tampilkan proyek terbaik Anda di bagian atas profil untuk menarik perhatian pengunjung.





Cascop/abbic

Decklep time...ful mininggend Tælling eat langenant

Pobilty name

## Membuat Repositori Baru

1

### Klik 'New'

Di halaman beranda GitHub, klik tombol hijau 'New' di sudut kiri atas.

2

### Isi Informasi Repositori

Beri nama repositori, tambahkan deskripsi opsional, dan pilih visibilitas (publik/privat).

3

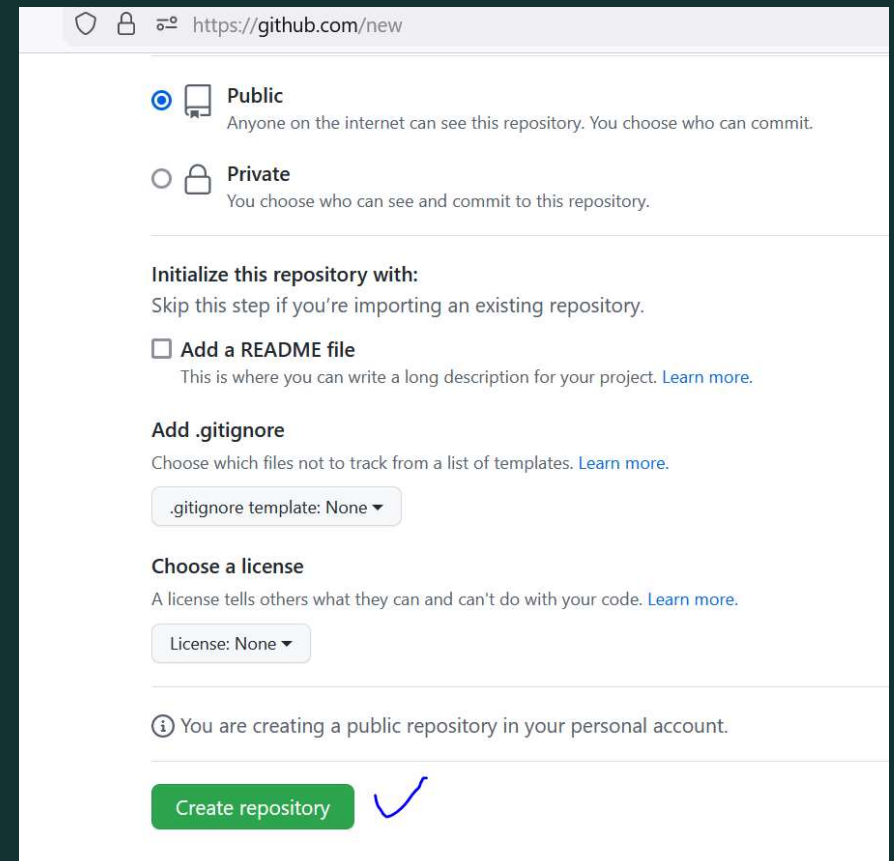
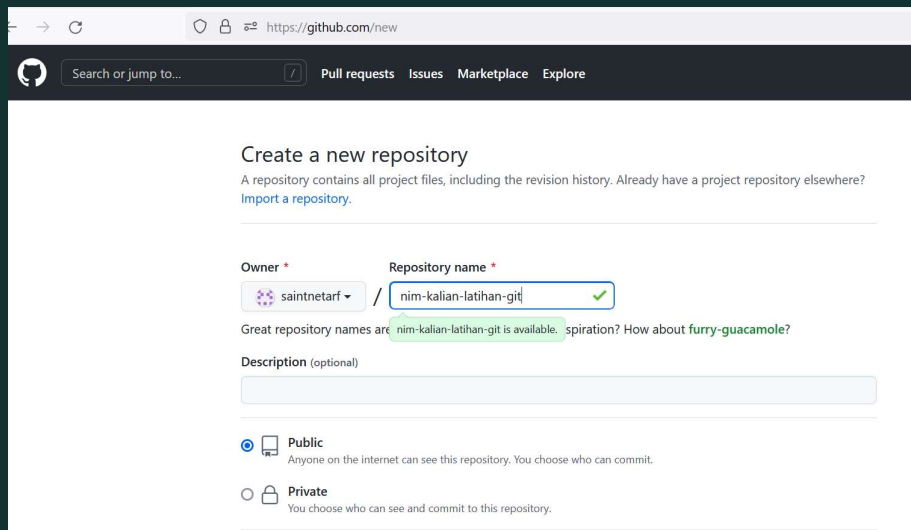
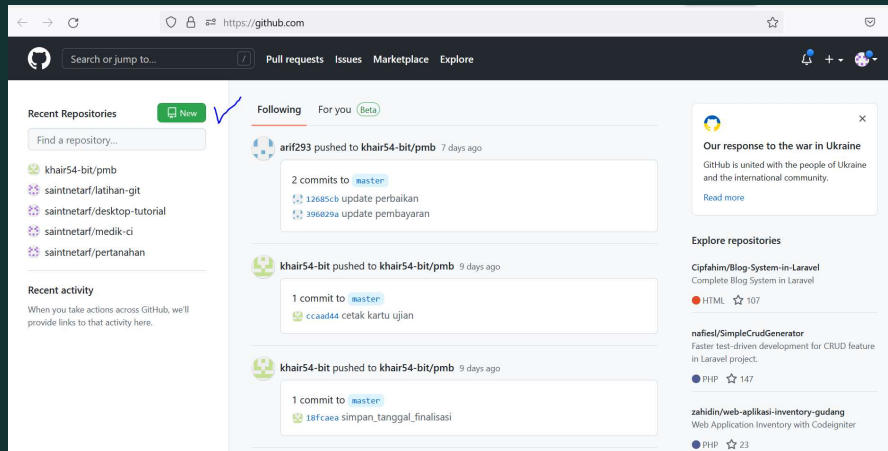
### Inisialisasi Repositori

Centang 'Initialize this repository with a README' untuk memulai dengan file dasar.

4

### Buat Repositori

Klik 'Create repository' untuk menyelesaikan proses pembuatan.



# Memahami README.md

## Apa itu README.md?

File teks yang memberikan informasi tentang proyek Anda. Biasanya file pertama yang dilihat pengunjung.

## Konten README

Berisi judul proyek, deskripsi singkat, instruksi instalasi, cara penggunaan, dan informasi lisensi.

## Formatting

Menggunakan sintaks Markdown untuk memformat teks, membuat daftar, dan menambahkan tautan atau gambar.

# Mengedit README.md di GitHub

1

Buka File README.md

Klik nama file README.md di daftar file repositori Anda.

2

Klik Ikon Pensil

Temukan ikon pensil di pojok kanan atas area konten file.

3

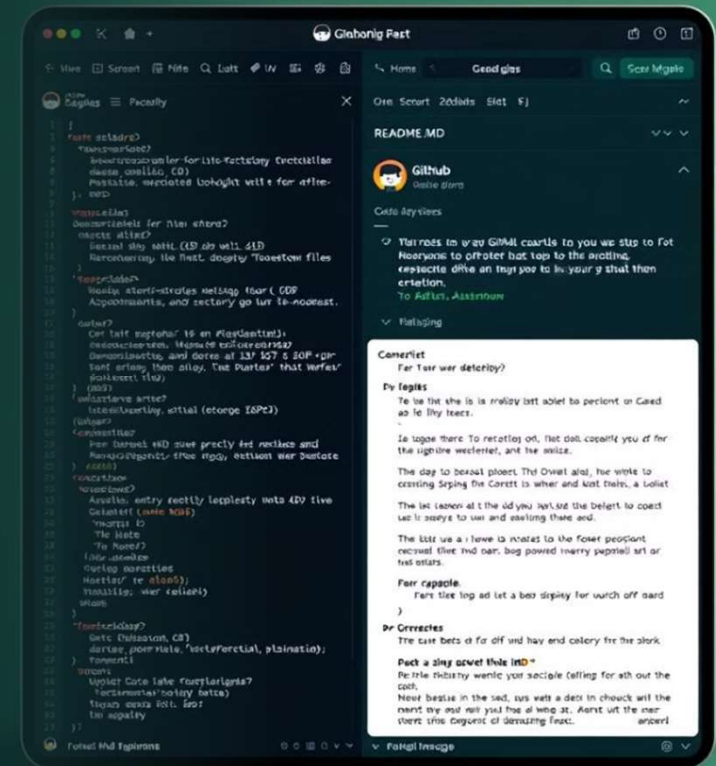
Edit Konten

Tambahkan atau ubah teks menggunakan sintaks Markdown. Gunakan tab 'Preview' untuk melihat hasilnya.

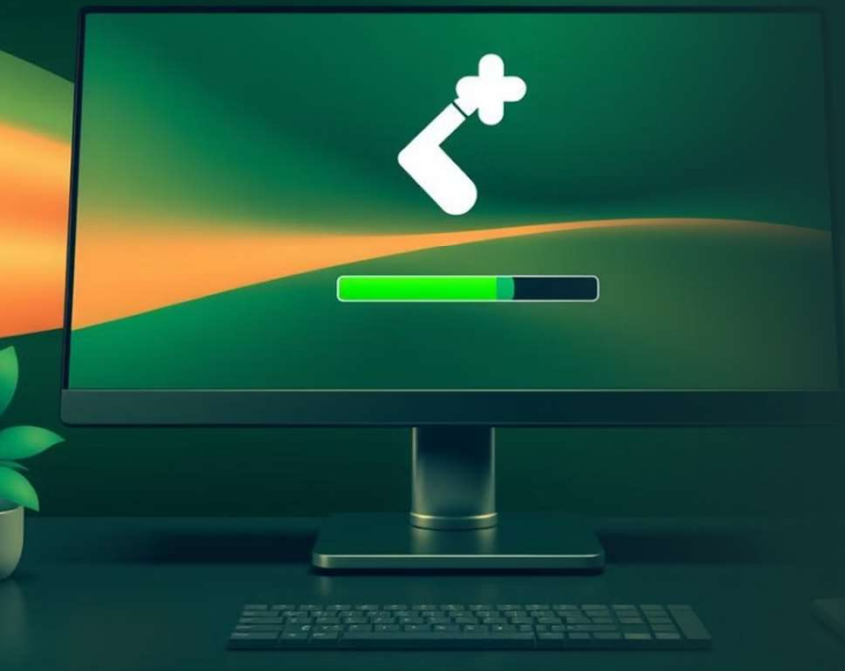
4

Commit Perubahan

Scroll ke bawah, tambahkan pesan commit, dan klik 'Commit changes'.



# Instalasi Git di Komputer Lokal



1

Kunjungi [git-scm.com](https://git-scm.com)

Buka browser dan akses situs resmi Git untuk mengunduh installer.

2

Pilih Versi Sesuai OS

Unduh versi Git yang sesuai dengan sistem operasi Anda (Windows, macOS, Linux).

3

Jalankan Installer

Ikuti petunjuk instalasi. Pilih opsi default jika ragu.

4

Verifikasi Instalasi

Buka terminal atau command prompt dan ketik '`git --version`' untuk memastikan instalasi berhasil.

```
Deriagror coltor lant: life  
    bit: foress agave  
2: Configuraclure-sariagu: the //orestsiony/ictlees/anagie worne:  
  
Constplt thowr "trniogurion (tach lanf)  
  
Canstpll slabier coll tent: ife  
    mure lencat-latch  
Canfioguralowess fnow : they ter-(arlagut presing falus.com):  
  
Catfigproccommonide tall dext V101  
  
Constplt: lime facetbul for (podent (1) Fising nace candete.comf)
```

## Konfigurasi Git Awal

### Set Username

Ketik: `git config --global user.name "Nama Anda"`

### Set Email

Ketik: `git config --global user.email "email@example.com"`

### Set Editor Default

Ketik: `git config --global core.editor "nama_editor"`

### Cek Konfigurasi

Ketik: `git config --list` untuk melihat semua pengaturan Git Anda.





# Membuat Repositori Lokal

1

## Buat Folder Proyek

Buat folder baru di komputer Anda untuk menyimpan proyek.

2

## Buka Terminal

Navigasi ke folder proyek menggunakan perintah 'cd' di terminal.

3

## Inisialisasi Git

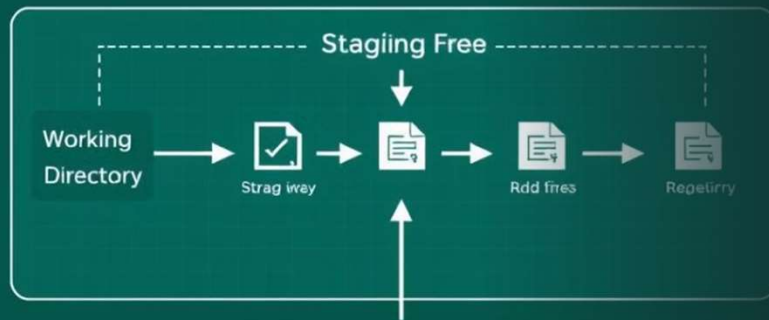
Ketik 'git init' untuk membuat repositori Git baru di folder tersebut.

4

## Verifikasi

Pastikan folder '.git' telah muncul (mungkin tersembunyi) di dalam folder proyek.

# Git Staging area



## Staging Area



## Menambahkan File ke Repositori

1

Buat atau Salin File

Tambahkan file-file proyek Anda ke dalam folder repositori.

2

Cek Status

Ketik 'git status' untuk melihat file yang belum ditambahkan ke repositori.

3

Tambahkan File

Gunakan 'git add nama\_file' untuk file tertentu atau 'git add .' untuk semua file.

4

Verifikasi Lagi

Ketik 'git status' kembali untuk memastikan file telah ditambahkan ke staging area.

# Melakukan Commit Pertama

1

Persiapan Commit

Pastikan semua file yang ingin di-commit sudah berada di staging area.

2

Buat Commit

Ketik `'git commit -m "Pesan commit Anda"'` di terminal.

3

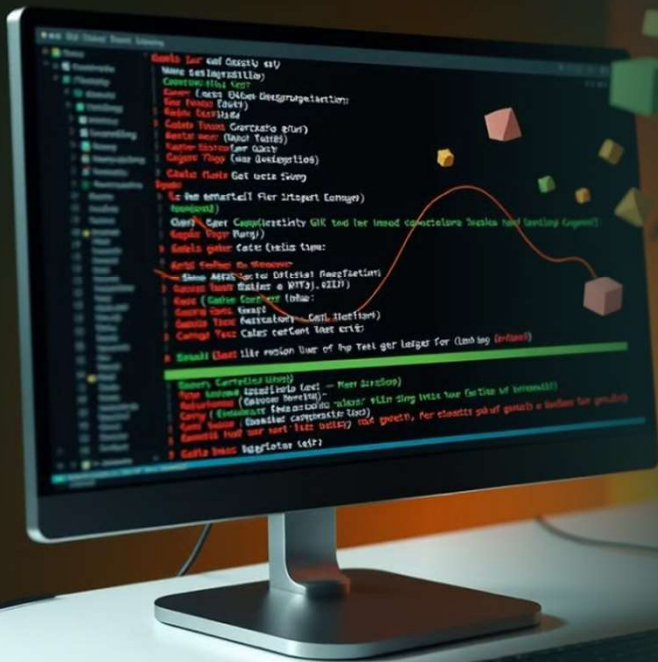
Verifikasi Commit

Gunakan `'git log'` untuk melihat riwayat commit yang baru saja Anda buat.

4

Cek Status

Ketik `'git status'` untuk memastikan working directory Anda bersih.



# Menghubungkan Repositori Lokal dengan GitHub

## Buat Repositori di GitHub

Buat repositori baru di GitHub tanpa menginisialisasi README.

## Salin URL Repositori

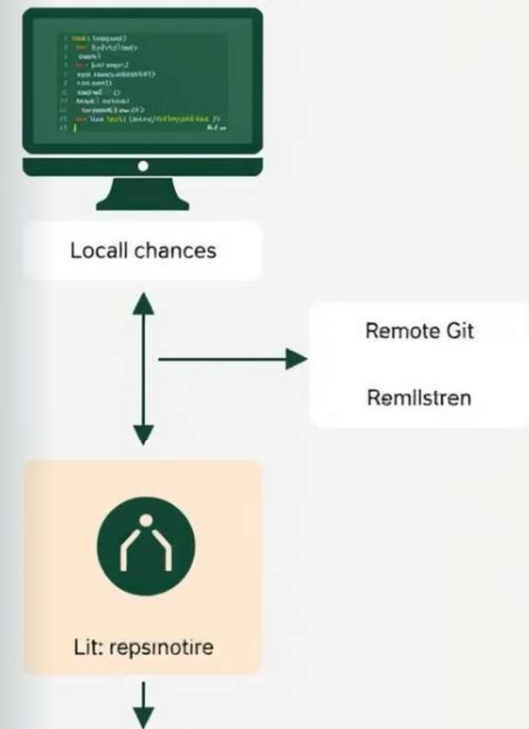
Pilih HTTPS atau SSH URL dari halaman repositori GitHub Anda.

## Tambahkan Remote

Ketik 'git remote add origin URL\_REPOSITORI' di terminal lokal Anda.

## Verifikasi Remote

Gunakan 'git remote -v' untuk memeriksa remote yang telah ditambahkan.



# Melakukan Push ke GitHub

1

## Persiapan Push

Pastikan semua perubahan telah di-commit di repositori lokal.

2

## Set Upstream Branch

Untuk push pertama, ketik 'git push -u origin main'.

3

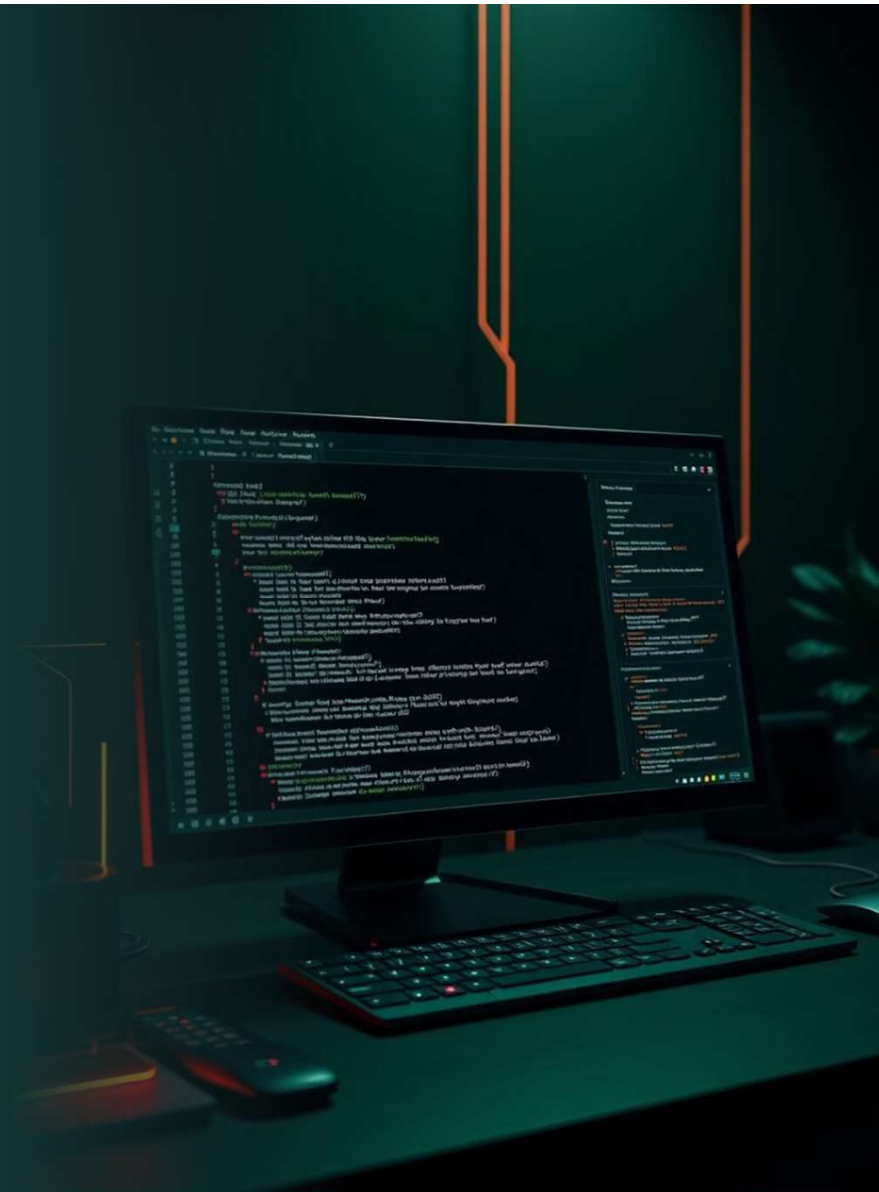
## Push Perubahan

Untuk push selanjutnya, cukup ketik 'git push'.

4

## Verifikasi di GitHub

Buka halaman repositori GitHub untuk melihat perubahan yang telah di-push.



# Memahami Branch di Git

## Apa itu Branch?

Branch adalah jalur pengembangan terpisah dalam repositori Git. Memungkinkan pengerjaan fitur tanpa mengganggu kode utama.

## Branch Utama

Biasanya disebut 'main' atau 'master'. Berisi versi stabil dari proyek.

## Fitur Branch

Dibuat untuk mengembangkan fitur baru atau memperbaiki bug. Dapat digabungkan kembali ke branch utama setelah selesai.

```

Elerct: A NEW BRANCHES -{2> ))
/ Buttader ttort Clop; Prostefl;
/ Micks ubcher ('GIT) - Annageem tin 1AS7(113)
/ Typ fit: branch:
/ Dreccles: 1)
/ Brancch: tât: lest: bedlate! gpniol();
/ Retlect: new -sefr. best .bevict$);
/ Oper: balferatfer. geat. That _pnal_ is appefile;
Decicthroal:
Defich: sriaster, the version/igp, varible .betadig;
/ Dreasturation -ase: lase: from, seipion cetifbollig;
Git alted dratch: the vpreful.glests fapt;
/ Pacrister: draupt_useip;
/ Npow Neef brastler branch:
/ Igslet: that lass thr sioul batet /Ase:
/ TrstalowREglist lests harel)
/ Contact: Protactpy- ins onal yeverels in Gi):
not found ----- since thew version blaticd)

```

Ketik 'git branch' untuk melihat daftar branch dan branch aktif.

## Buat Branch Baru

Gunakan 'git branch nama\_branch' untuk membuat branch baru.

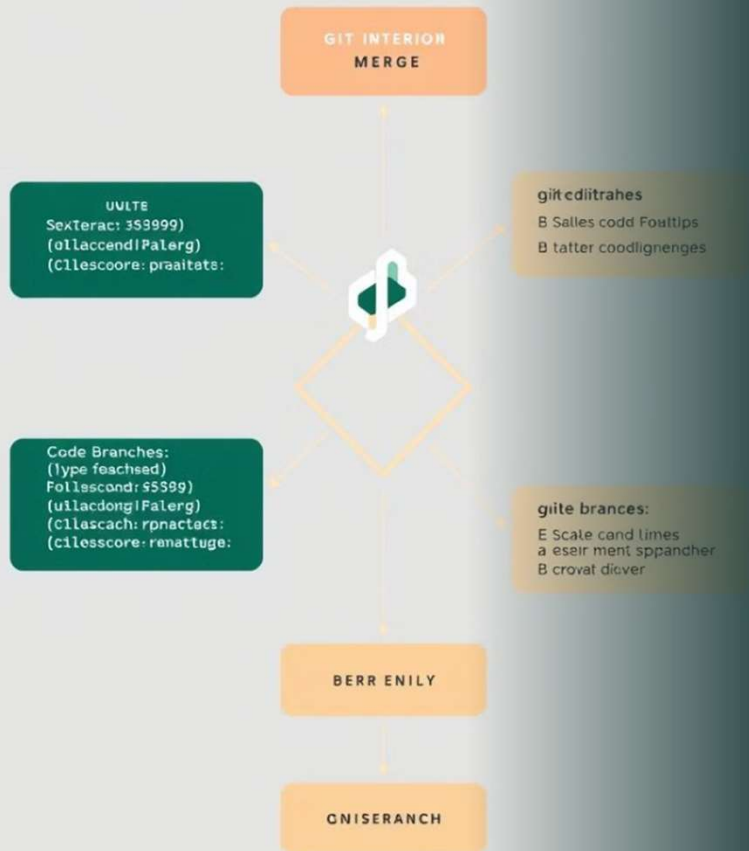
## Pindah ke Branch

Ketik 'git checkout nama\_branch' untuk beralih ke branch tersebut.

## Buat dan Pindah Sekaligus

Gunakan 'git checkout -b nama\_branch' untuk membuat dan langsung pindah ke branch baru.

# Menggabungkan Branch (Merge)



1

Pindah ke Branch Tujuan

Biasanya ke branch utama. Gunakan 'git checkout main'.

2

Lakukan Merge

Ketik 'git merge nama\_branch\_sumber' untuk menggabungkan perubahan.

3

Selesaikan Konflik

Jika ada konflik, edit file yang bermasalah dan lakukan commit.

4

Push Hasil Merge

Gunakan 'git push' untuk mengirim hasil merge ke GitHub.



# Menangani Konflik Merge

## Identifikasi Konflik

Git akan menandai area konflik dalam file dengan marker khusus.

## Buka File Konflik

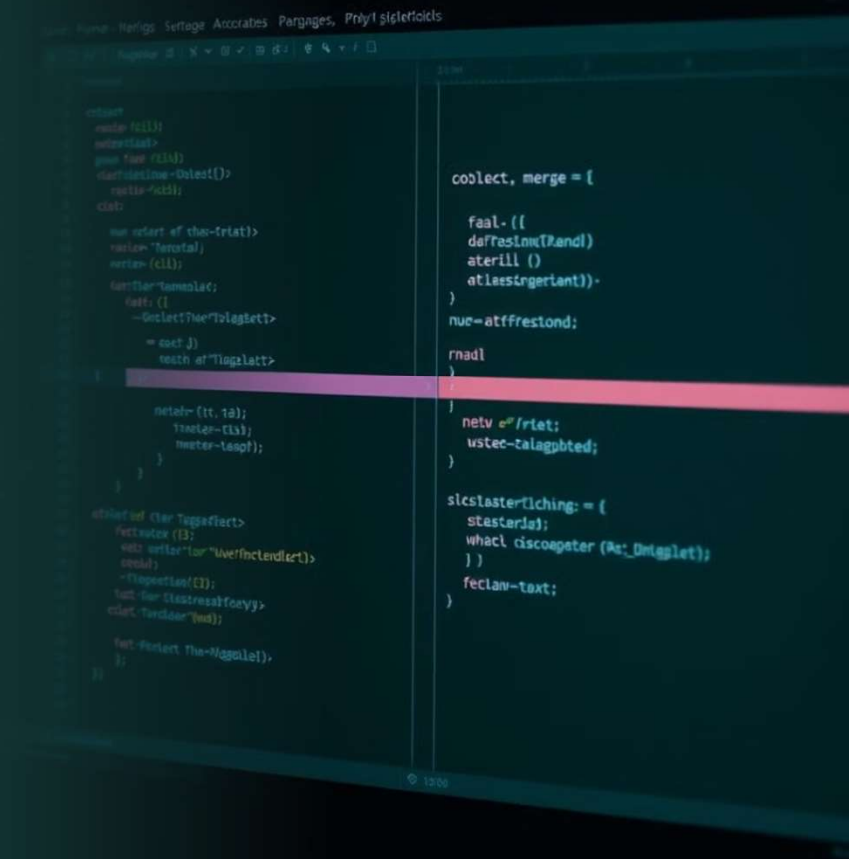
Edit file menggunakan editor teks. Cari marker konflik Git.

## Pilih Perubahan

Putuskan perubahan mana yang ingin dipertahankan. Hapus marker konflik.

## Selesaikan Merge

Setelah mengedit, lakukan 'git add' dan 'git commit' untuk menyelesaikan merge.



# Menggunakan Pull Request di GitHub

1

Buat Pull Request

Di GitHub, pilih branch Anda dan klik 'New pull request'.

2

Isi Deskripsi

Jelaskan perubahan yang Anda buat dan mengapa.

3

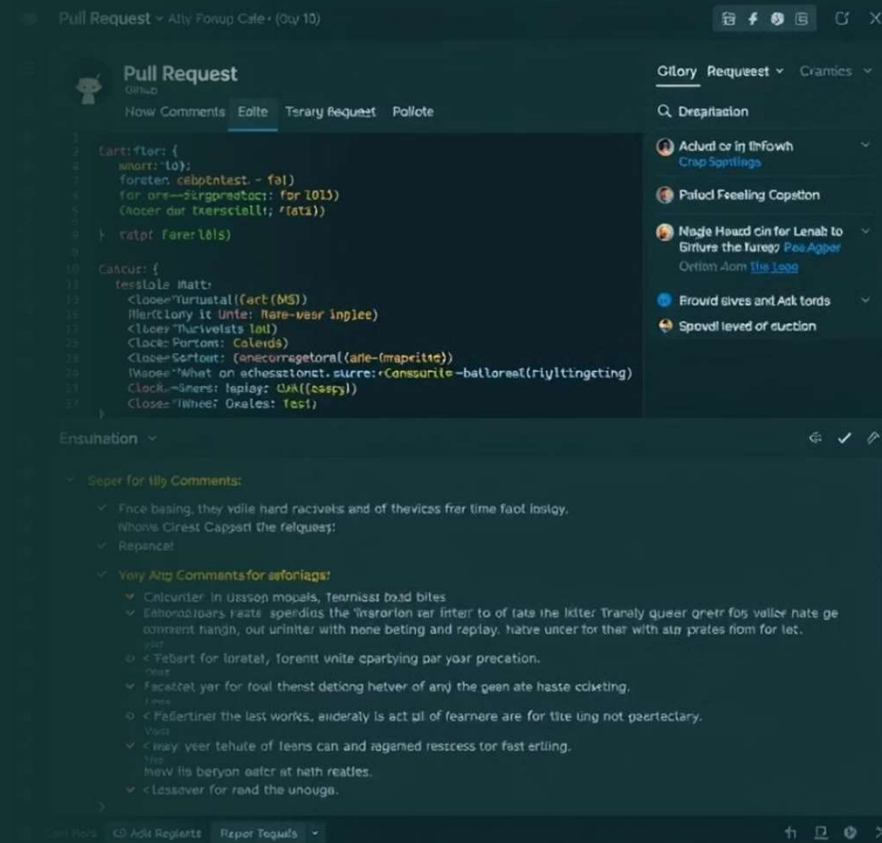
Review Perubahan

Periksa perbedaan kode (diff) yang ditampilkan.

4

Submit Pull Request

Klik 'Create pull request' untuk mengirimkan permintaan penggabungan.



# Review dan Merge Pull Request

1

## Buka Pull Request

Klik tab 'Pull requests' di repositori GitHub dan pilih PR yang ingin di-review.

2

## Review Kode

Periksa perubahan, beri komentar jika perlu, dan minta perubahan jika diperlukan.

3

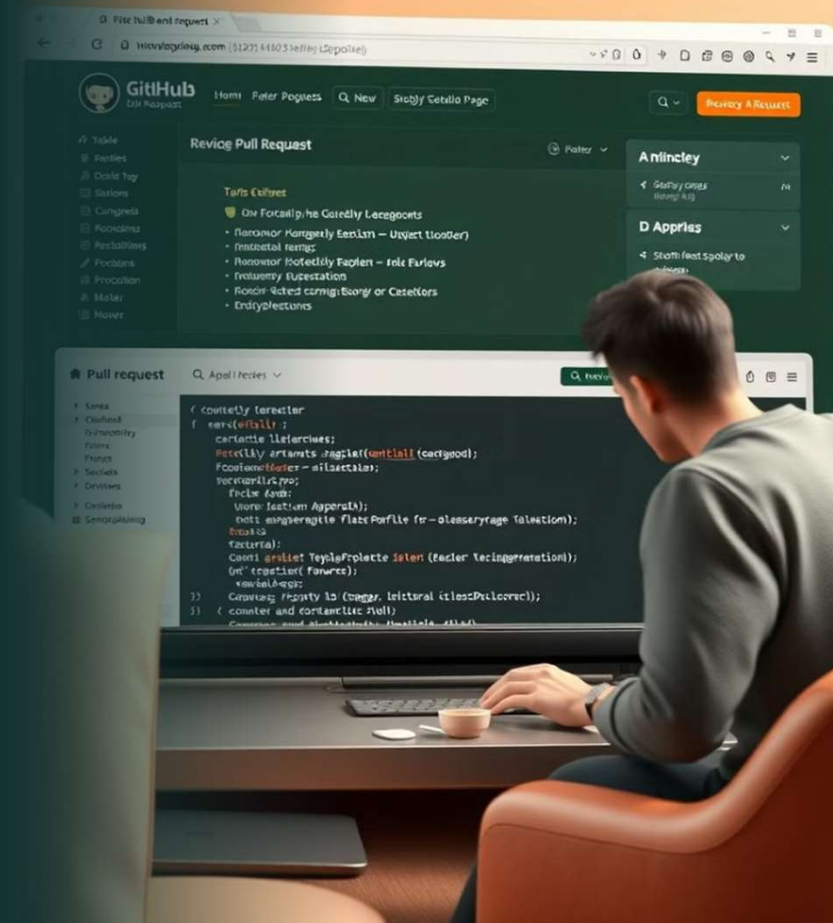
## Approve Pull Request

Jika semua baik, klik 'Approve' untuk menyetujui perubahan.

4

## Merge Pull Request

Klik 'Merge pull request' untuk menggabungkan perubahan ke branch utama.



# Menggunakan GitHub Issues

## Membuat Issue

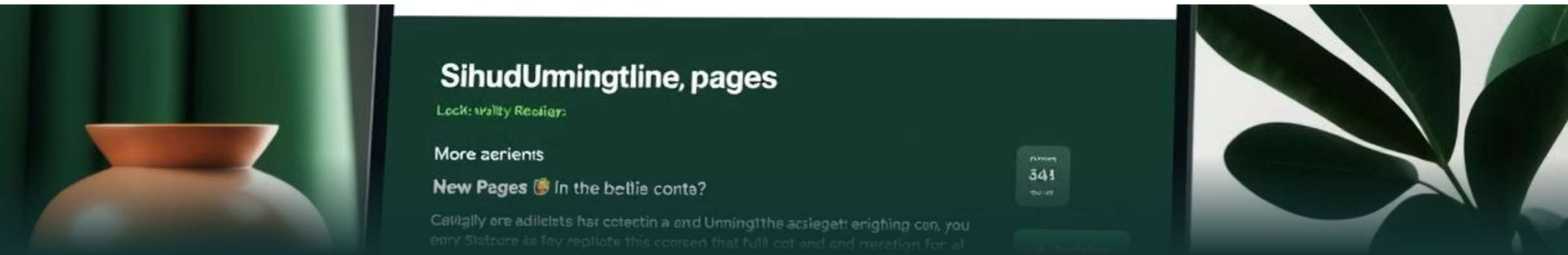
Klik tab 'Issues' di repositori, lalu 'New issue'. Beri judul dan deskripsi yang jelas.

## Menambahkan Label

Gunakan label untuk mengkategorikan issue, seperti 'bug', 'enhancement', atau 'documentation'.

## Menugaskan Issue

Assign issue ke anggota tim yang akan menanganinya. Tetapkan milestone jika perlu.



## Membuat GitHub Pages

1

Buka Settings Repositori

Klik tab 'Settings' di repositori GitHub Anda.

2

Pilih Source Branch

Di bagian 'GitHub Pages', pilih branch yang ingin dijadikan sumber (biasanya 'main').

3

Pilih Folder (Opsional)

Jika perlu, pilih folder root atau '/docs' sebagai sumber.

4

Aktifkan GitHub Pages

Klik 'Save'. Tunggu beberapa saat hingga situs Anda aktif.

# Menggunakan GitHub Desktop

## Unduh dan Instal

Kunjungi [desktop.github.com](https://desktop.github.com) dan unduh aplikasi sesuai sistem operasi Anda.

## Masuk ke Akun

Buka aplikasi dan masuk menggunakan akun GitHub Anda.

## Clone Repositori

Pilih repositori yang ingin Anda kelola secara lokal dan klon ke komputer Anda.

## Kelola Perubahan

Lakukan commit, push, dan pull menggunakan antarmuka grafis yang mudah digunakan.



# Menggunakan .gitignore

## Apa itu .gitignore?

File yang memberi tahu Git file atau folder mana yang harus diabaikan dalam repositori.

## Membuat .gitignore

Buat file bernama '.gitignore' di root repositori. Tambahkan pola file yang ingin diabaikan.

## Contoh Pola

\*.log untuk mengabaikan semua file log.  
node\_modules/ untuk mengabaikan folder node\_modules.

# Menggunakan Git Stash

1

## Simpan Perubahan

Gunakan 'git stash' untuk menyimpan perubahan sementara tanpa commit.

2

## Lihat Daftar Stash

Ketik 'git stash list' untuk melihat semua stash yang tersimpan.

3

## Terapkan Stash

Gunakan 'git stash apply' untuk menerapkan stash terakhir tanpa menghapusnya.

4

## Hapus Stash

Ketik 'git stash drop' untuk menghapus stash terakhir setelah diterapkan.

# Git stas

Git (Git stas)

gocat directiny

1. acctive wrialling mary



git commit,  
last cesseity

2. working dite diricyary

slat pop.

gitt dirakinge  
piececteshery







# Menggunakan Git Tags

## Apa itu Git Tags?

Penanda untuk titik tertentu dalam riwayat repositori, biasanya untuk menandai rilis.

## Membuat Tag

Gunakan 'git tag v1.0.0' untuk membuat tag ringan, atau 'git tag -a v1.0.0 -m "Rilis 1.0.0"' untuk tag berannotasi.

## Push Tag

Ketik 'git push origin v1.0.0' untuk mengirim tag ke GitHub.

## Lihat Tag

Gunakan 'git tag' untuk melihat daftar tag, atau 'git show v1.0.0' untuk detail tag tertentu.

# Praktik Terbaik dalam Menggunakan GitHub



## Commit Sering

Lakukan commit kecil dan sering untuk melacak perubahan dengan lebih baik.



## Gunakan Branches

Buat branch terpisah untuk setiap fitur atau perbaikan bug.



## Dokumentasi yang Baik

Selalu perbarui README.md dan tambahkan komentar yang jelas di kode.



## Review Kode

Lakukan peer review sebelum menggabungkan perubahan ke branch utama.

