

A picture containing icon

Description automatically generated

**TUTORIAL IMPLEMENTASI GITHUB API MENGGUNAKAN GOLANG**

Memahami cara menggunakan Github API untuk mengunggah file ke repository melalui Golang

**Rofi Nafiis Zain**

**Haris Riyoni**



**TUTORIAL IMPLEMENTASI GITHUB API MENGGUNAKAN GOLANG**

Memahami cara menggunakan Github API untuk mengunggah file ke repository melalui Golang

***Penulis :***

Rofi Nafiis Zain

Haris Riyoni

ISBN : -

***Editor :***

Rolly Awangga

***Penyunting :***

Rolly Awangga

***Desain sampul dan Tata letak :***

Haris Riyoni

***Penerbit :***

Penerbit Buku Pedia

***Redaksi :***

Athena Residence Blok. E No. 1, Desa Ciwaruga,

Kec. Parongpong, Kab. Bandung Barat 40559

Tel. 628-775-2000-300

Email : penerbit@bukupedia.co.id

***Distributor :***

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2022

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan

dengan cara apa pun tanpa ijin tertulis dari penerbit

# KATA PENGANTAR

P

uji syukur kita panjatkan ke hadirat Allah SWT, karena atas berkat dan rahmat-Nya, buku ini dapat hadir sebagai panduan yang membahas Tutorial Implementasi Github API menggunakan Golang. Dengan judul "Memahami cara menggunakan Github API untuk mengunggah file ke repository melalui Golang," buku ini membawa kita dalam sebuah perjalanan mendalam untuk memahami integrasi antara Golang dan Github API dalam pengembangan aplikasi web.

Dalam era teknologi yang terus berkembang pesat, penggunaan Github API telah menjadi landasan penting dalam kolaborasi pengembangan perangkat lunak. Melalui buku ini, pembaca akan diajak untuk memahami langkah-langkah konkret dalam menggunakan Github API, khususnya dengan penerapan Golang. Pembaca akan diberikan wawasan mendalam tentang konsep dan teknik yang diperlukan untuk mengunggah file ke repository Github dengan efisien dan efektif menggunakan bahasa pemrograman Golang.

Kami, penulis, merasa sangat bahagia dapat berbagi pengetahuan dan pengalaman kami dalam dunia pengembangan aplikasi web, terutama dalam konteks integrasi antara Golang dan Github API. Buku ini bukan hanya sekadar panduan praktis, tetapi juga merupakan upaya kami untuk memberikan pemahaman menyeluruh dan mendalam tentang topik ini.

Kami berharap bahwa buku ini akan menjadi sumber ilmu yang bermanfaat bagi pembaca, terutama bagi para pengembang web yang ingin menguasai implementasi Github API dengan menggunakan Golang. Semoga buku ini dapat memberikan kontribusi positif bagi perkembangan dunia teknologi, dan semoga Allah SWT senantiasa memberikan keberkahan pada setiap langkah kita.

Akhir kata, terima kasih kepada semua pihak yang telah mendukung dan mendorong terwujudnya buku ini. Semoga bermanfaat dan selamat membaca!

Bandung, November 2023

Penulis

# PRAKATA

B

uku ini, berjudul "Tutorial Implementasi Github API menggunakan Golang: Memahami cara menggunakan Github API untuk mengunggah file ke repository melalui Golang," hadir sebagai upaya kami untuk memberikan panduan yang komprehensif dalam integrasi antara bahasa pemrograman Golang dan Github API. Dalam dunia yang terus berubah dan berkembang, pengetahuan akan integrasi ini menjadi semakin penting untuk pengembang aplikasi web.

Pentingnya kerja sama antara Golang dan Github API semakin terasa dalam pengembangan layanan web yang handal dan efisien. Namun, literatur yang membahas implementasi spesifik ini masih terbilang terbatas. Oleh karena itu, melalui buku ini, kami berupaya untuk mengisi celah tersebut dan menyajikan panduan praktis untuk para pengembang yang ingin memahami cara menggabungkan Golang dengan Github API untuk keperluan pengunggahan file ke repository.

Buku ini dibagi menjadi beberapa bab yang secara sistematis membahas langkah-langkah implementasi, konsep dasar, dan praktik coding yang diperlukan. Harapannya, pembaca akan mendapatkan pemahaman yang mendalam tentang cara memanfaatkan Github API menggunakan Golang, serta bagaimana mengintegrasikan keduanya untuk menciptakan layanan web yang efektif.

Kami berharap buku ini dapat menjadi sumber rujukan yang bermanfaat bagi pembaca, khususnya bagi para pengembang yang tengah menjelajahi kemungkinan integrasi Golang dan Github API. Terima kasih kepada semua pihak yang telah memberikan dukungan dan inspirasi dalam proses penulisan buku ini.

Semoga buku ini memberikan manfaat, inspirasi, dan pengetahuan yang berharga. Selamat menikmati pembelajaran dan eksplorasi dalam implementasi Github API menggunakan Golang.

# DAFTAR ISI

[KATA PENGANTAR 1](#_Toc152193758)

[PRAKATA 2](#_Toc152193759)

[DAFTAR ISI 3](#_Toc152193760)

[DAFTAR GAMBAR 4](#_Toc152193761)

[PENDAHULUAN 6](#_Toc152193762)

[BAB I 8](#_Toc152193763)

[PENGANTAR GO DAN GITHUB 8](#_Toc152193764)

[8](#_Toc152193765)

[1.1 PENGENALAN 8](#_Toc152193766)

[1.1.1 TENTANG GO 8](#_Toc152193767)

[1.1.2 BENEFIT MEMPELAJARI GO 9](#_Toc152193769)

[1.1.2 KEUNGGULAN GO DENGAN BAHASA YANG LAIN 9](#_Toc152193770)

[1.1.3 KEKURANGAN DARI GOLANG 10](#_Toc152193771)

[1.1.4 DENGAN BAHASA APA SAJA GO BISA DIINTEGRASIKAN 10](#_Toc152193772)

[1.5 APA ITU GITHUB 11](#_Toc152193773)

[BAB II 13](#_Toc152193774)

[PERSIAPAN SOFTWARE DAN INSTALASI 13](#_Toc152193775)

[2.1 KEBUTUHAN-KEBUTUHAN YANG HARUS DI PENUHI 13](#_Toc152193776)

[2.1.1 INSTALASI GO 13](#_Toc152193777)

[2.1.3 AKUN GITHUB 13](#_Toc152193778)

[2.1.5 ACCESS TOKEN GITHUB 14](#_Toc152193779)

[2.1.6 AKUN GOOGLE CLOUD CONSOLE 14](#_Toc152193780)

[2.2 LOGIKA INTEGRASI 14](#_Toc152193781)

[2.2.1 GOLANG SEBAGAI BACKEND 15](#_Toc152193782)

[BAB III 16](#_Toc152193783)

[LANGKAH INTEGRASI GOLANG DAN GITHUB API 16](#_Toc152193784)

[3.1 PEMBUATAN PACKAGE UNTUK GITHUB API 16](#_Toc152193785)

[BAB IV 17](#_Toc152193786)

[Hasil Akhir Tujuan Tampilan dan Debug 17](#_Toc152193787)

[4.1 HASIL TAMPILAN FRONTEND 17](#_Toc152193788)

[4.2 DEBUG HASIL FRONTEND 18](#_Toc152193790)

[DAFTAR PUSTAKA 21](#_Toc152193791)

[GLOSARIUM 53](#_Toc152193792)

[INDEKS 58](#_Toc152193793)

# DAFTAR GAMBAR

**BAB I**

[Gambar 1. 1 Logo Golang 1](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677371)

[Gambar 1. 2 Logo JavaScript 4](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677372)

[Gambar 1. 3 Heroku Logo 7](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677373)

[Gambar 1. 4 MongoDB Logo 8](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677374)

[Gambar 1. 5 Github Logo 9](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677375)

**BAB II**

**BAB III**

[Gambar 3. 1 Struktur File Backend 14](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677376)

[Gambar 3. 2 File Struct 16](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677377)

[Gambar 3. 3 Function Backend 1 18](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677378)

[Gambar 3. 4 Function Backend 2 19](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677379)

[Gambar 3. 5 Function GetNamaUser 20](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677380)

[Gambar 3. 6 File Test Function Code 21](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677381)

[Gambar 3. 7 Test Get Nama dan Surat 21](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677382)

[Gambar 3. 8 Function Module 23](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677383)

[Gambar 3. 9 Struct Package Model 24](#_Toc140677384)

[Gambar 3. 10 Struktur Folder Iteung 24](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677385)

[Gambar 3. 11 Import di Controller 26](#_Toc140677386)

[Gambar 3. 12 Setting Database di db.go 27](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677387)

[Gambar 3. 13 Tampilan Database 27](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677388)

[Gambar 3. 14 Iteung Controller 29](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677389)

[Gambar 3. 15 File Url.go 30](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677390)

[Gambar 3. 16 Setting Cors di Cors.go 31](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677391)

[Gambar 3. 17 Deploy App Heroku 1 32](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677392)

[Gambar 3. 18 Deploy App Heroku 2 32](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677393)

[Gambar 3. 19 Add JawsDB Heroku 1 33](#_Toc140677394)

[Gambar 3. 20 Add JawsDB Heroku 2 34](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677395)

[Gambar 3. 21 Tampilan Sudah TerInstall JawsDB 34](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677396)

[Gambar 3. 22 Tampilan Jika Membuka JawsDB 35](#_Toc140677397)

[Gambar 3. 23 Tampilan Setting App Heroku 35](#_Toc140677398)

[Gambar 3. 24 Setting Config Vars 36](#_Toc140677399)

[Gambar 3. 25 Add IP Heroku Akses Database MongoDB 37](#_Toc140677400)

[Gambar 3. 26 Allow Akses MongoDB 38](#_Toc140677401)

[Gambar 3. 27 Tampilan Aplikasi dengan OpenApp 38](#_Toc140677402)

[Gambar 3. 28 Tampilan Aplikasi 38](#_Toc140677403)

[Gambar 3. 29 Struktur File Javascript 39](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677404)

[Gambar 3. 30 Get Data dari /nama 41](#_Toc140677405)

[Gambar 3. 31 Tampilan Url.js 41](#_Toc140677406)

[Gambar 3. 32 Template dan Isi untuk Data API 42](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677407)

[Gambar 3. 33 Tampilan Code Croot.js 42](#_Toc140677408)

[Gambar 3. 34 Table HTML 43](#_Toc140677409)

[Gambar 3. 35 Tampilan /Surat/Insert 44](#_Toc140677410)

[Gambar 3. 36 Tampilan Url2.js 44](#_Toc140677411)

[Gambar 3. 37 Tampilan Croot2.js 44](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677412)

[Gambar 3. 38 Form HTML 45](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677413)

[Gambar 3. 39 Deploy Frontend Github Pages 46](#_Toc140677414)

[Gambar 3. 40 Tampilan Aplikasi Front-End 46](#_Toc140677415)

[Gambar 3. 41 Struktur File Repo ke Url 47](#_Toc140677416)

[Gambar 3. 42 Tampilan Index.html Github Pages 47](#_Toc140677417)

**BAB IV**

[Gambar 4. 1 Tampilan Index Get API GitPages 48](#_Toc140677418)

[Gambar 4. 2 Tampilan Form Chat 49](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677419)

[Gambar 4. 3 Console Get API 49](#_Toc140677420)

[Gambar 4. 4 Masukan Data Form Chat 50](#_Toc140677421)

[Gambar 4. 5 Form Chat Terkirim 51](#_Toc140677422)

[Gambar 4. 6 Tampilan Hasil Send Post Form Chat 51](file:///C:\Users\imyha\Documents\penting%20buat%20buku\integrasi%20go%20dan%20js%20dalam%20pembuatan%20web%20service%20surat%20menyurat.docx#_Toc140677423)

# PENDAHULUAN

Dalam era revolusi teknologi yang terus berkembang, integrasi perangkat lunak menjadi kunci utama untuk menciptakan solusi aplikasi yang efektif dan inovatif. Salah satu aspek penting dalam pengembangan aplikasi web adalah kemampuan untuk berinteraksi dengan repositori kode, dan Github API telah menjadi fondasi yang tak tergantikan dalam hal tersebut.

Buku ini, berjudul "Tutorial Implementasi Github API menggunakan Golang: Memahami cara menggunakan Github API untuk mengunggah file ke repository melalui Golang," memasuki ranah penting ini dengan fokus pada integrasi antara bahasa pemrograman Golang dan Github API. Tujuan kami adalah memberikan panduan komprehensif bagi pembaca, terutama para pengembang web, untuk memahami dan mengimplementasikan dengan efektif Github API menggunakan Golang.

Dalam dunia pengembangan perangkat lunak, Github telah menjadi pusat kolaborasi dan manajemen kode yang sangat diperlukan. Github API, sebagai antarmuka pemrograman aplikasi dari platform tersebut, memberikan kemampuan untuk mengotomatiskan sejumlah tugas, termasuk pengunggahan file ke repository. Golang, dengan kecepatan eksekusi dan kemudahan penggunaannya, menjadi pilihan yang ideal untuk mengintegrasikan aplikasi dengan Github API.

Buku ini terdiri dari beberapa bab yang secara progresif membimbing pembaca melalui konsep-konsep dasar, langkah-langkah implementasi, dan praktek coding. Bab pertama akan mengenalkan konsep dasar Golang dan Github API, sementara bab-bab berikutnya akan membahas secara mendalam langkah-langkah praktis untuk mengunggah file ke repository melalui Golang.

Pembaca akan mendapatkan pemahaman yang kuat tentang cara menggabungkan kekuatan Golang dan Github API untuk menciptakan layanan web yang efisien dan andal. Melalui contoh kode yang jelas dan penjelasan yang terperinci, buku ini dirancang untuk memberikan wawasan praktis dan kemampuan langsung yang dapat diterapkan dalam proyek pengembangan web.

Mari bersiap untuk memasuki perjalanan menarik dalam mengimplementasikan Github API dengan menggunakan Golang, Selamat Membaca!

# BAB I

# PENGANTAR GO DAN GITHUB

### PENGENALAN

Oke kita masuk ke bagian paling awal dari buku ini yang dimana kita akan berkenalan dengan Bahasa yang akan kita gunakan pada kesempatan kali ini, yaitu Go Language.

### 1.1.1 TENTANG GO

Go, juga dikenal sebagai Golang, adalah bahasa pemrograman open source yang dikembangkan oleh Google pada tahun 2007. Go dirancang dengan tujuan menyederhanakan proses pengembangan perangkat lunak dan meningkatkan kinerja. Keunggulan Go terletak pada sintak yang mudah dipahami, kompilasi yang cepat, serta kemampuannya dalam mengelola konkurensi dan pemrograman paralel. Dengan perpustakaan standar yang kuat, Go memberikan kemudahan dalam membangun aplikasi yang efisien dan andal.



### 

Gambar 1. Logo Golang

### 1.1.2 BENEFIT MEMPELAJARI GO

Mempelajari Go memiliki beberapa manfaat yang signifikan adalah sebagai berikut:

* Pertama, Go menawarkan sintak yang sederhana, mempercepat proses pembelajaran dan membantu pengembang menghasilkan kode yang lebih bersih dan efisien.
* Kedua, Go menyediakan dukungan bawaan untuk konkurensi, memudahkan pengembang dalam menangani proses paralel. Fitur ini sangat berguna dalam pengembangan aplikasi berskala besar. Selain itu, Go juga memiliki performa yang tinggi dan penggunaan memori yang efisien, menjadikannya pilihan yang baik untuk aplikasi yang responsif dan scalable.

### 1.1.2 KEUNGGULAN GO DENGAN BAHASA YANG LAIN

Go memiliki beberapa keunggulan dibandingkan dengan bahasa pemrograman lainnya. Pertama, Go memiliki kinerja yang sangat baik, dengan waktu eksekusi yang cepat dan penggunaan memori yang efisien. Go juga memiliki dukungan bawaan untuk konkurensi, sehingga mudah untuk mengembangkan aplikasi yang berjalan secara paralel.

Selain itu, Go memiliki sistem manajemen memori otomatis (garbage collector) yang canggih, yang mengurangi beban pengembang dalam mengelola alokasi memori. Go juga dikenal dengan stabilitasnya, dengan versi bahasa yang kompatibel secara mundur dan dukungan yang kuat dari komunitas pengembang.

### 1.1.3 KEKURANGAN DARI GOLANG

Golang atau Go adalah bahasa pemrograman yang telah mendapatkan popularitas dalam beberapa tahun terakhir. Namun, seperti halnya bahasa pemrograman lainnya, Golang juga memiliki beberapa kekurangan yang harus dipertimbangkan sebelum memilihnya sebagai bahasa utama dalam pengembangan perangkat lunak. Salah satu kekurangan utama Golang adalah kurangnya dukungan untuk tipe generic. Tipe generic memungkinkan pengembang menulis kode yang bekerja secara generik dengan berbagai jenis tipe data tanpa perlu mengulangi kode yang serupa. Ketidakhadiran tipe generic di Golang dapat mengakibatkan perulangan kode yang dapat mengurangi efisiensi dan kejelasan kode.

Selain itu, beberapa pengembang juga menganggap ekosistem pihak ketiga untuk Golang masih kurang berkembang dibandingkan bahasa pemrograman populer lainnya seperti Python atau JavaScript. Meskipun ekosistem Golang terus berkembang dan semakin beragam, namun terkadang sulit untuk menemukan atau memperoleh pustaka pihak ketiga yang dibutuhkan dalam proyek tertentu. Keterbatasan ini dapat menjadi tantangan bagi pengembang yang bergantung pada pustaka pihak ketiga untuk mempercepat proses pengembangan dan menghadapi kebutuhan khusus.

### 1.1.4 DENGAN BAHASA APA SAJA GO BISA DIINTEGRASIKAN

Salah satu keunggulan Golang adalah kemampuannya untuk diintegrasikan dengan bahasa pemrograman lain. Dengan antarmuka C, Golang dapat berkomunikasi dengan bahasa pemrograman lain yang mendukung antarmuka C, seperti C++ dan Rust. Integrasi ini memungkinkan pengembang menggunakan kode dari bahasa-bahasa tersebut sambil tetap memanfaatkan keunggulan kinerja tinggi dari Golang.

Selain itu, Golang juga dapat diintegrasikan dengan bahasa pemrograman Python menggunakan paket Cgo. Dengan dukungan ini, pengembang dapat dengan mudah berkolaborasi antara Golang dan Python, dua bahasa pemrograman yang sering digunakan dalam pengembangan perangkat lunak.

Lebih lanjut, Golang juga berfungsi sebagai antarmuka untuk memanggil kode JavaScript melalui WebAssembly. Fitur ini membuka peluang integrasi yang lebih erat dengan teknologi web modern, sehingga memungkinkan pengembang menggabungkan kekuatan Golang dengan ekosistem JavaScript yang luas dalam pengembangan aplikasi web kompleks dan efisien.

### 1.5 APA ITU GITHUB

GitHub adalah platform hosting yang populer untuk pengembangan perangkat lunak berbasis Git. Ini memungkinkan para pengembang untuk menyimpan, mengelola, dan berbagi kode sumber proyek secara terpusat. GitHub memainkan peran penting dalam kolaborasi dan kerjasama tim dalam pengembangan perangkat lunak dengan menyediakan alat kolaborasi, sistem kontrol versi yang kuat, dan integrasi dengan berbagai layanan pengembangan lainnya.

Gambar 1. Github Logo

Kelebihan GitHub:

Kolaborasi dan Sharing: GitHub memungkinkan pengembang untuk berkolaborasi dalam proyek secara terpusat. Dengan adanya sistem kontrol versi Git, tim dapat bekerja bersama-sama dalam mengembangkan kode, melakukan perubahan, dan menggabungkan kode secara mudah.

Sistem Kontrol Versi yang Kuat: Git adalah sistem kontrol versi yang populer dan kuat, dan GitHub menyediakan antarmuka pengguna yang ramah untuk mengelola repositori Git. Pengembang dapat dengan mudah melakukan perubahan, membuat cabang, melakukan penggabungan, dan mengatasi konflik dengan cepat dan efisien.

Integrasi dengan Layanan Pengembangan Lainnya: GitHub memiliki integrasi yang luas dengan berbagai layanan pengembangan seperti CI/CD, layanan penerapan otomatis, alat manajemen proyek, dan banyak lagi. Hal ini memungkinkan pengembang untuk membangun alur kerja pengembangan yang otomatis dan efisien.

Masyarakat Open Source: GitHub memiliki komunitas yang besar dan aktif dari pengembang dan proyek open source. Ini memudahkan pengembang untuk menemukan proyek open source yang menarik, berkontribusi ke proyek tersebut, atau menggunakan perangkat lunak open source dalam proyek mereka.

Top of Form

# BAB II

# PERSIAPAN SOFTWARE DAN INSTALASI

### 2.1 KEBUTUHAN-KEBUTUHAN YANG HARUS DI PENUHI

Oke Kita masuk ke bagian selanjutnya dari buku ini yaitu, mempersiapkan kebutuhan software dan cara instalasi software dan Bahasa yang dibutuhkan. Kebutuhan dasar yang dibutuhkan tentu adalah aplikasi IDE untuk Code, dan akun github.

### 2.1.1 INSTALASI GO

Tentu selanjutnya yang kita install adalah Golang, untuk instalasi kalian bisa melalui link golang resmi bisa melalui link ini [The Go Programming Language](https://go.dev/). Untuk tambahan informasi Bahasa go ini dibuat Oleh Google dan Go adalah Bahasa yang digadangi akan populer untuk kedepannya. Dan jika kamu sudah menguasai dasar dari GO ini akan sangat mempermudah kamu dalam mengikuti buku ini.

### 2.1.3 AKUN GITHUB

Membuat akun Github, Github memiliki peran yang penting karena pada kasus apapun dalam membuat project, kamu dapat menyimpan project kamu di github dan juga github banyak platform deploy untuk memudahkan developer atau programmer untuk deploy Back-End atau Front-End dengan Project kita yang tersimpan di Github kita.

Contohnya Github sendiri memiliki deploy Frontend yaitu bernama Github-Pages fitur ini akan digunakan kita kedepannya karena github memberikan fitur ini secara gratis. Platform seperti 000Webhost, Netlify, Heroku dst. Adalah contoh platform yang terintegrasi dengan Github, Netlify untuk deploy atau mempublikasi Web Statis dan Heroku untuk Publikasi Website Dinamis.

Untuk Pendaftaran kamu bisa mengunjungi langsung Website mereka di official resminya atau melalui link ini [GitHub](https://github.com/). Dan diharapkan kamu sudah menguasai git.

### 2.1.5 ACCESS TOKEN GITHUB

Untuk menggunakan API dari Github kita harus memiliki access token agar kita dapat bebas mebuat client dan melakukan aktivitas menggunakan API sesuai dengan Dokumentasi yang disediakan oleh GitHub.

### 2.1.6 AKUN GOOGLE CLOUD CONSOLE (OPSIONAL)

Selanjutnya, jika sudah memiliki akun GitHub dan sudah menginstall Go. Hal yang harus dilakukan adalah membuat akun Google Cloud Console. Pada Google Cloud Console kita akan memanfaatkan fitur Cloud Functions untuk mendeploy backen kita nanti. Cara Pembuatan akun dan project bisa melalui link berikut [Google Cloud Platform](https://cloud.google.com/?hl=en).

### LOGIKA INTEGRASI

Oke Kita sudah masuk kedalam Logika dari Aplikasi yang kita akan buat disini aku mau memberi sedikit pemahaman tentang ke-2 bahasa yaitu Golang dan Javascript mereka digunakan untuk apa aja sih?

### 2.2.1 GOLANG SEBAGAI BACKEND

Pertama Golang, Bahasa Pemrograman Golang ini kita akan gunakan untuk sebagai Backend untuk Aplikasi kita yang dimana tentunya Golang adalah Bahasa dalam Backend terkuat dan tercepat dibanding dengan Bahasa yang lain karena golang meng-convert code program kita menjadi biner yang dimana ini adalah keunggulan Go dari pada Bahasa yang lain.

Pada Buku ini kita akan membuat backend menggunakan Golang akan tetapi dengan cara yang berbeda dengan menggunakan Package yang kita buat sendiri.

# BAB III

# LANGKAH INTEGRASI GOLANG DAN GITHUB API

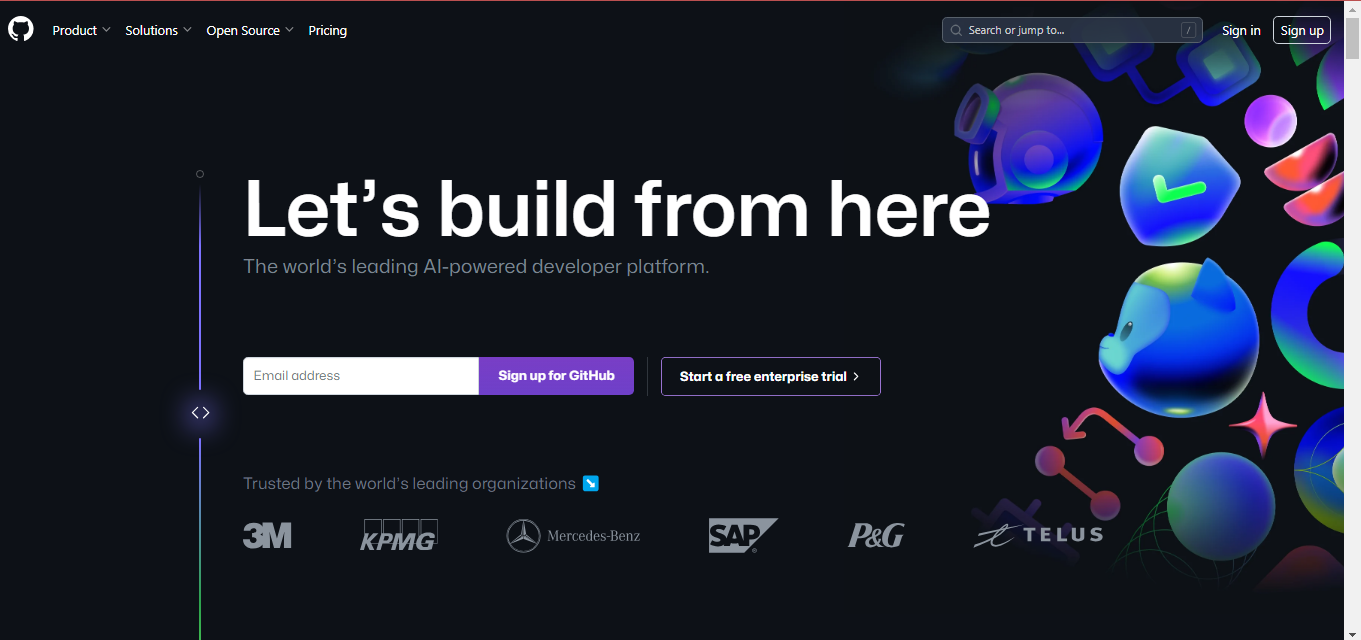
### 3.1 LANGKAH INTEGRASI GITHUB API

Pada bab ini kita akan mulai membuat program untuk mengimplementasikan Github API dengan menggunakan Golang. Terdapat beberapa langkah yang harus diikuti untuk melakukan praktik pada bab ini.

Pada Sub-bab ini yang harus dilakukan adalah memastikan akun GitHub yang dimiliki telah aktif, melakukan generate access token GitHub, dan membaca dokumentasi dari Library yang akan kita gunakan.

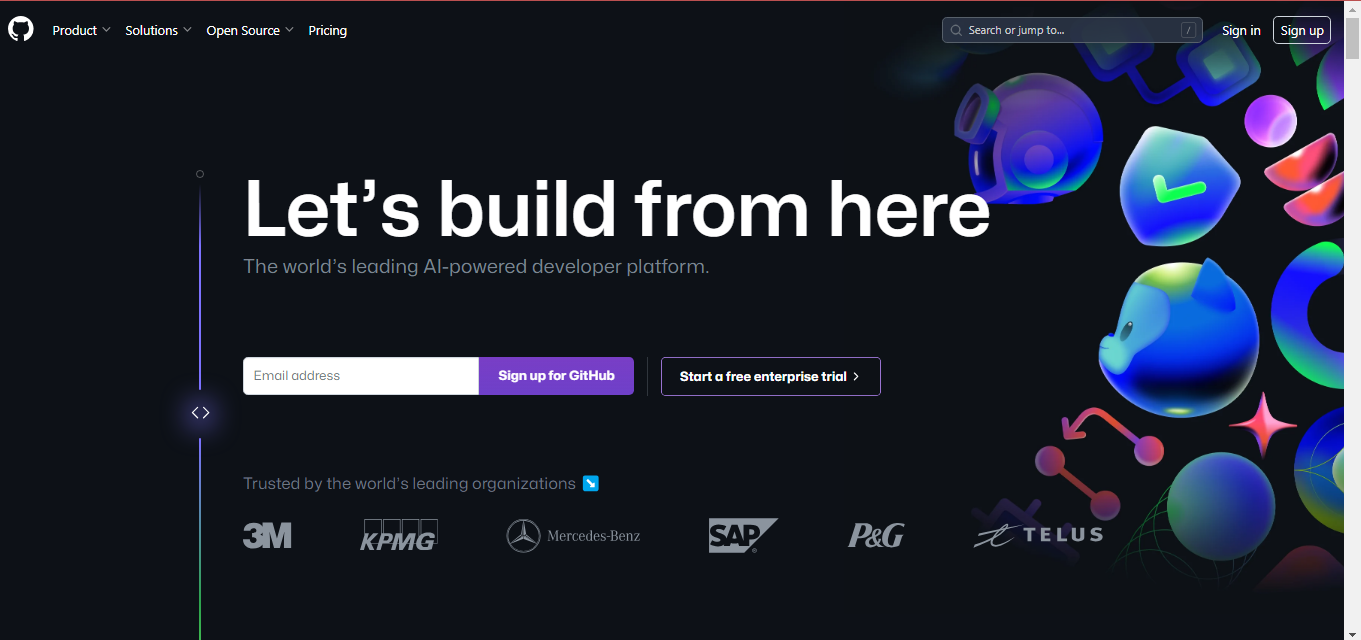
### 3.1.1 LOGIN AKUN GITHUB

Langkah pertama yaitu, lakukan login ke akun GitHub dengan mengakses link berikut: [Github](https://github.com/). Maka tampilan laman utamanya seperti terdapat pada Gambar 3.1.



**Gambar 3. 1 Laman Utama Github**

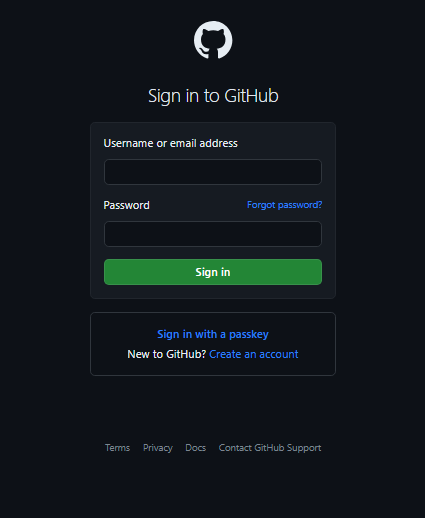
Kemudian, Klik Button Sign In yang ada pada bagian kanan atas layar anda, atau dapat dilihat pada Gambar 3.2



**Gambar 3. 2 Button SIgn in**

Jika anda tidak memiliki akun GitHub sebelumnya, lakukan Sign Up terlebih dahulu agar bisa melanjutkan ke langkah selanjutnya. Untuk Sign Up dapat mengunjungi link [Sign Up GitHub](https://github.com/signup).

Selanjutnya, pastikan anda sudah berada di laman login GitHub, masukan alamat email atau username dan password akun GitHub. Lalu Klik tombol Sign In. Seperti pada Gambar 3.3.



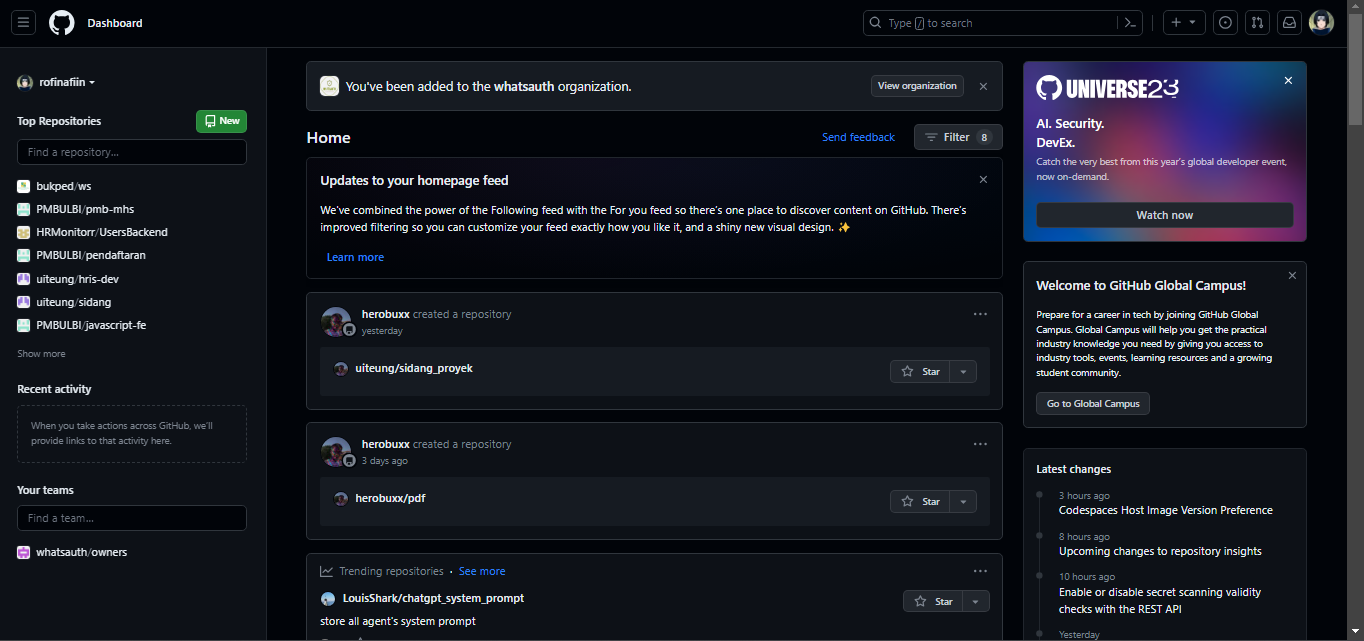
**Gambar 3. 3 Laman Sign In Github**

Pastikan Username dan password yang dimasukan adalah username yang benar, agar dapat melanjutkan ke langkah selanjutnya.

### 3.1.2 GENERATE ACCESS TOKEN

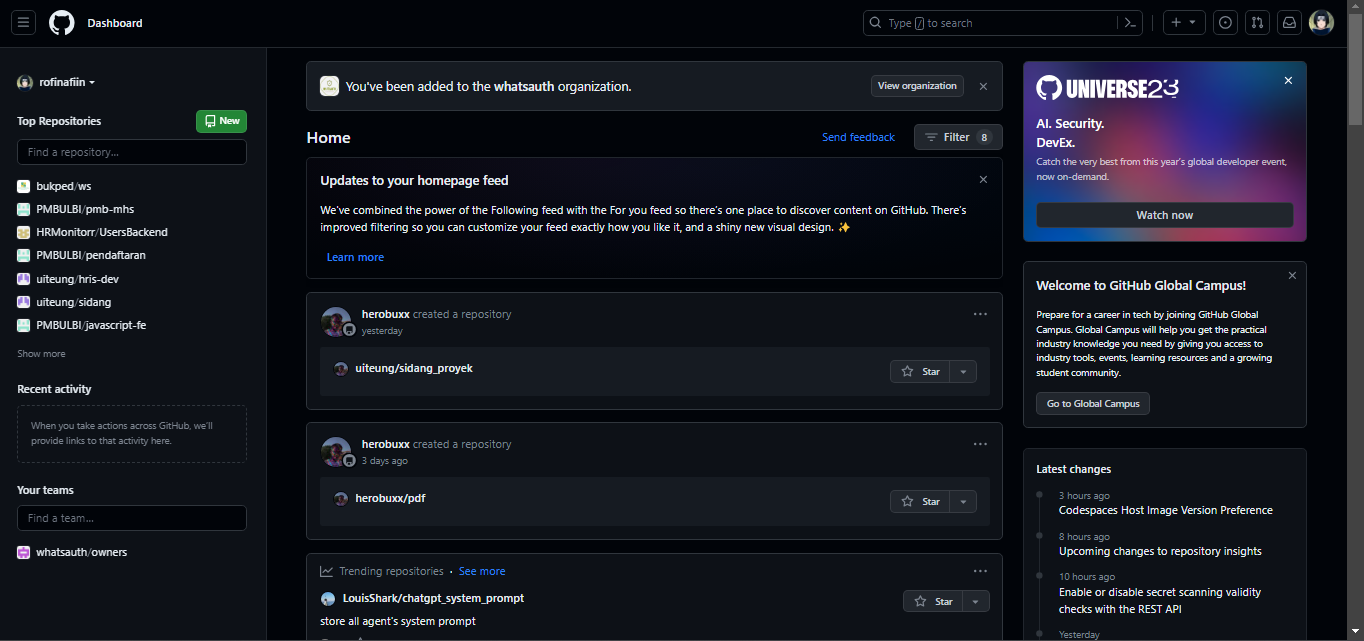
Langkah selanjutnya adalah mengambil access token untuk akun GitHub anda. Ikuti langkah-langkah berikut ini:

1. Pastikan anda berada pada laman utama GitHub setelah login, seperti pada Gambar 3.4.



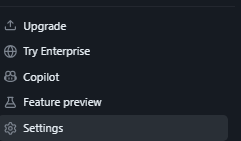
Gambar 3. Laman Utama GitHub

1. Kemudian klik profile akun yang ada pada bagian kanan atas layar anda.



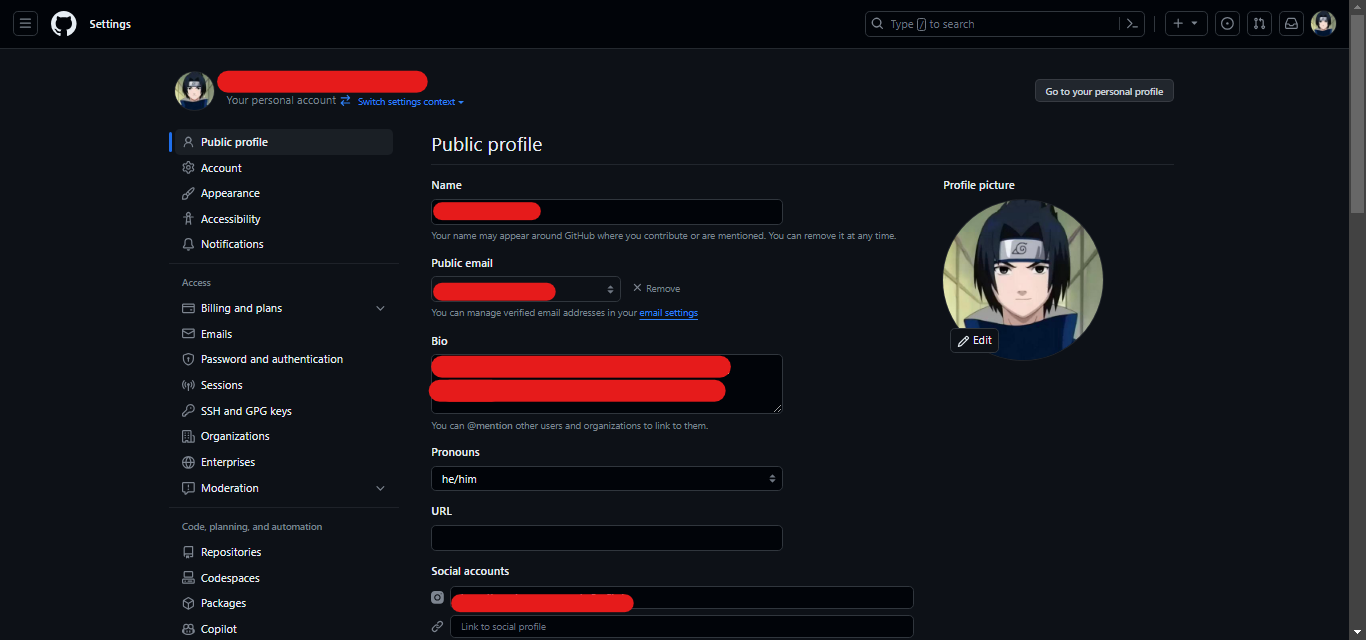
Gambar 3. Bagian Profile GitHub

1. Masuk ke menu settings.



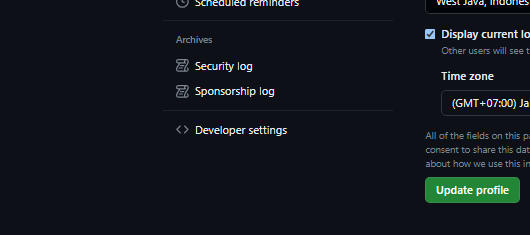
Gambar 3. Menu Settings

1. Setelah masuk ke menu settings maka tampilannya akan seperti ini



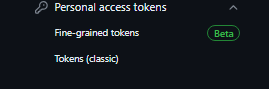
Gambar 3. Laman Settings Profile

1. Kemudian masuk ke menu Developer Settings yang ada pada bagian kiri bawah layar anda.



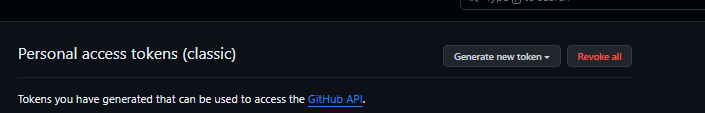
Gambar 3. Menu Developer Settings

1. Lalu masuk ke menu Personal Access Token – Tokens (Classic)



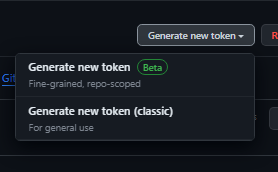
Gambar 3. Personal Access Tokens

1. Kemudian klik Generate new Token



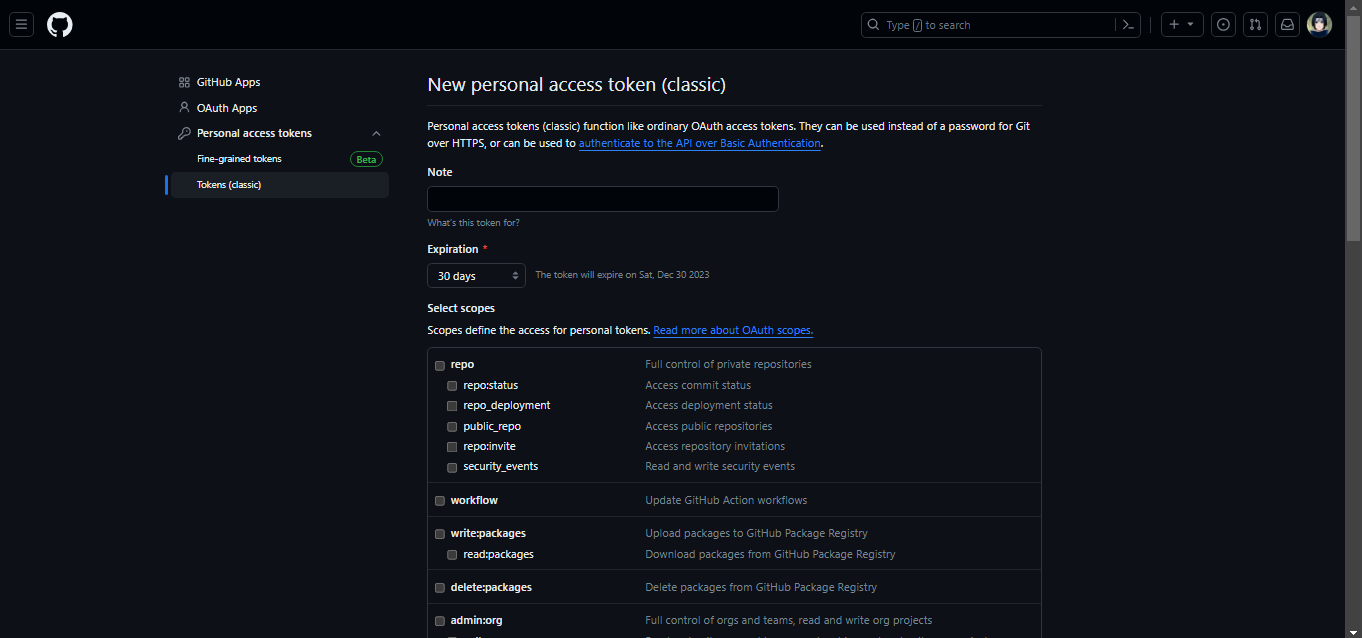
Gambar 3. Generate new token

1. Pilih Generate new Token Classic



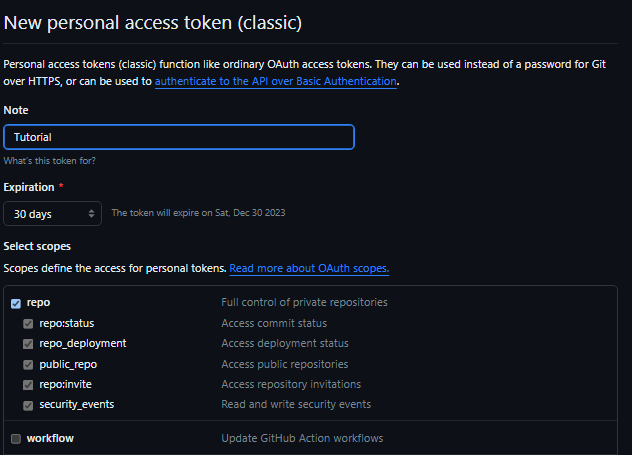
Gambar 3. Generate new token (classic)

1. Maka tampilan menu akan menjadi seperti berikut ini



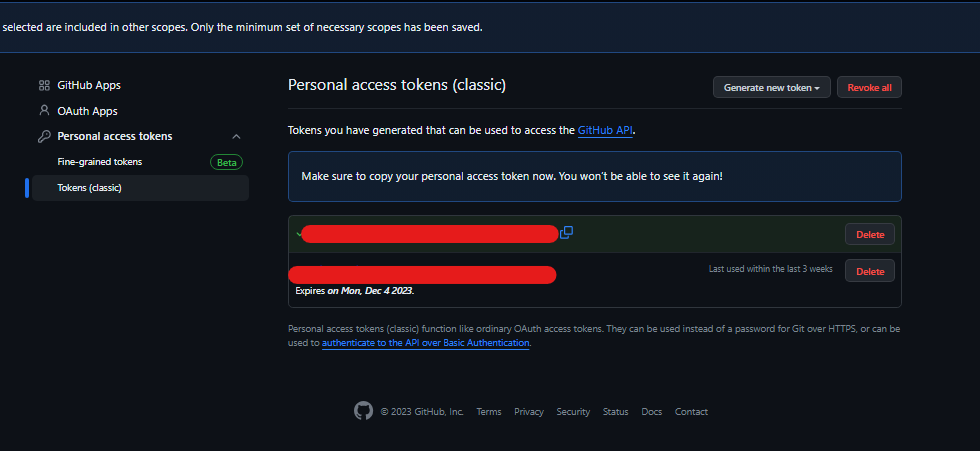
Gambar 3. Halaman Personal Access Token

1. Isi Kolom Note dan centang bagian repo untuk mendapatkan full access repositories.



Gambar 3. Mengisi Note dan Grant access

1. Kemudian, klik generate new token, setelah itu copy tokennya dan anda bisa simpan pada file json.



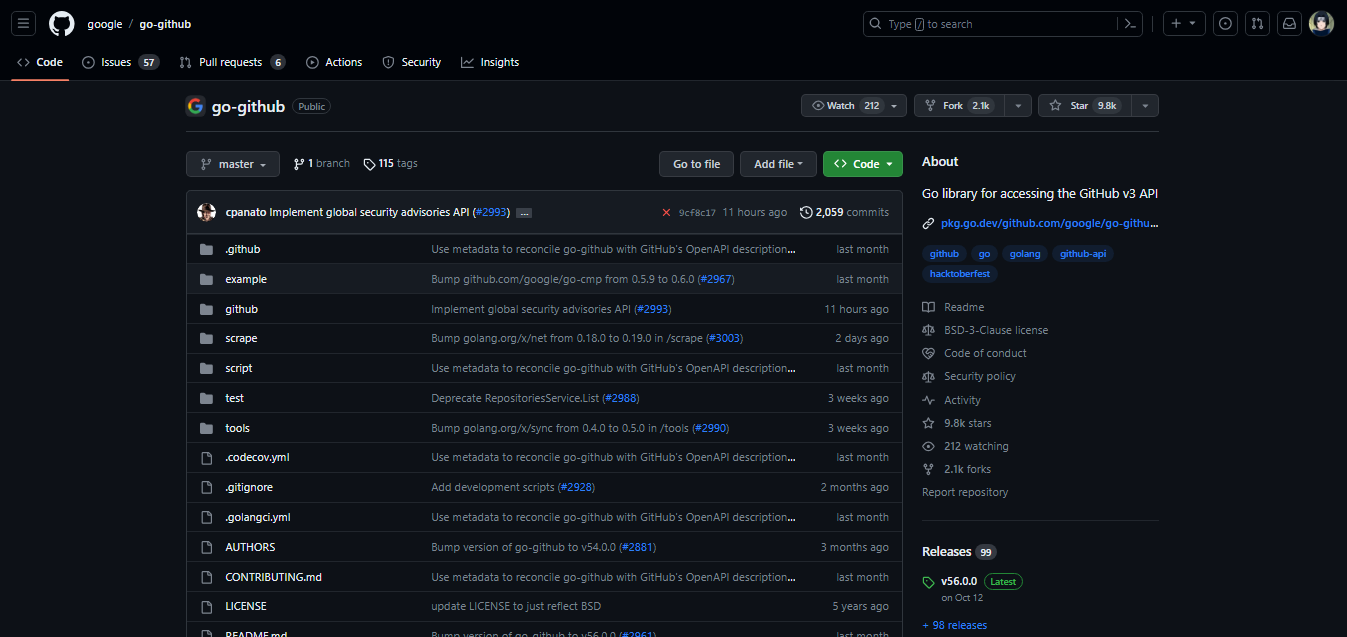
Gambar 3. New Token

Token yang didapatkan akan berlaku sesuai dengan batas waktu yang ditentukan pada saat generate token, anda dapat melakukan kustomisasi batas waktu pada saat generate token atau dapat dilihat pada laman berikut laman [expiration and revocation](https://docs.github.com/en/enterprise-server@3.6/authentication/keeping-your-account-and-data-secure/token-expiration-and-revocation).

### 3.2 MEMBUAT PACKAGE GOLANG

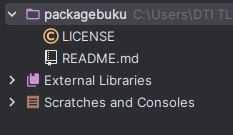
Langkah selanjutnya kita akan membuat package dengan menggunakan Bahasa pemrograman Go, untuk full access package yang dibuat dapat diakses pada url berikut [GithubWrapper](https://github.com/hrmonitorr/githubwrapper). Ikuti langkah – langkah berikut ini.

1. Langkah pertama kita akan mencari library Go yang akan kita gunakan pada praktik kali ini.
2. Library yang akan digunakan adalah library Go-Github yang tersedia pada url ini [go-github](https://github.com/google/go-github)



Gambar 3. Laman Github go-github

1. Kemudian buat Golang project menggunakan aplikasi IDE yang biasa anda gunakan (pada kesempatan ini kami menggunakan Goland)



Gambar 3. Folder project

1. Kemudian init project dengan syntax *go mod init <urlrepo>*



Gambar 3. Init project

1. Setelah itu download package go-github
2. Kemudian buat file bernama github.go, struct.go, dan github\_test.go
3. Sebelum membuat functions, buatlah struct pada struct.go seperti yang ada pada gambar berikut.
4. Setelah itu buatlah functions seperti pada gambar dibawah ini.
5. Jika anda ragu apakah functions berjalan atau tidak, lakukan test pada github\_test.go, seperti pada gambar berikut ini.
6. Jika merasa sudah berhasil dan file terdapat pada repository tujuan lakukan publish untuk package Go.

### 3.2.1 PUBLISH PACKAGE GO

Pada langkah sebelumnya kita telah membuat sebuah package untuk melakukan push ke repository Github dengan menggunakan Go. Nah, pada sub-bab kali ini kita akan melakukan publish package menuju pkg.go.dev dengan mengikuti langkah berikut ini.

1. Pastikan semua library yang digunakan sudah terdownload pada go.mod dan go.sum.
2. Untuk memastikan hal tersebut anda dapat menggunakan perintah go mod tidy.
3. Selanjutnya, Cek remote dari project anda dengan menggunakan perintah git remote –v
4. Jika remote sudah benar lanjutkan dengan melakukan commit terhadap file yang akan dipublish.
5. Lakukan commit seperti pada gambar berikut ini.
6. Kemudian lakukan tag
7. Kemudian push terlebih dahulu tag yang telah di set
8. Lakukan list untuk package yang akan dipublish
9. Langkah terakhir lakukan push ke repository tujuan.

Untuk mengetahui apakah package kita sudah publish atau belum, kita dapat melihatnya di pkg.go.dev/namapackage.

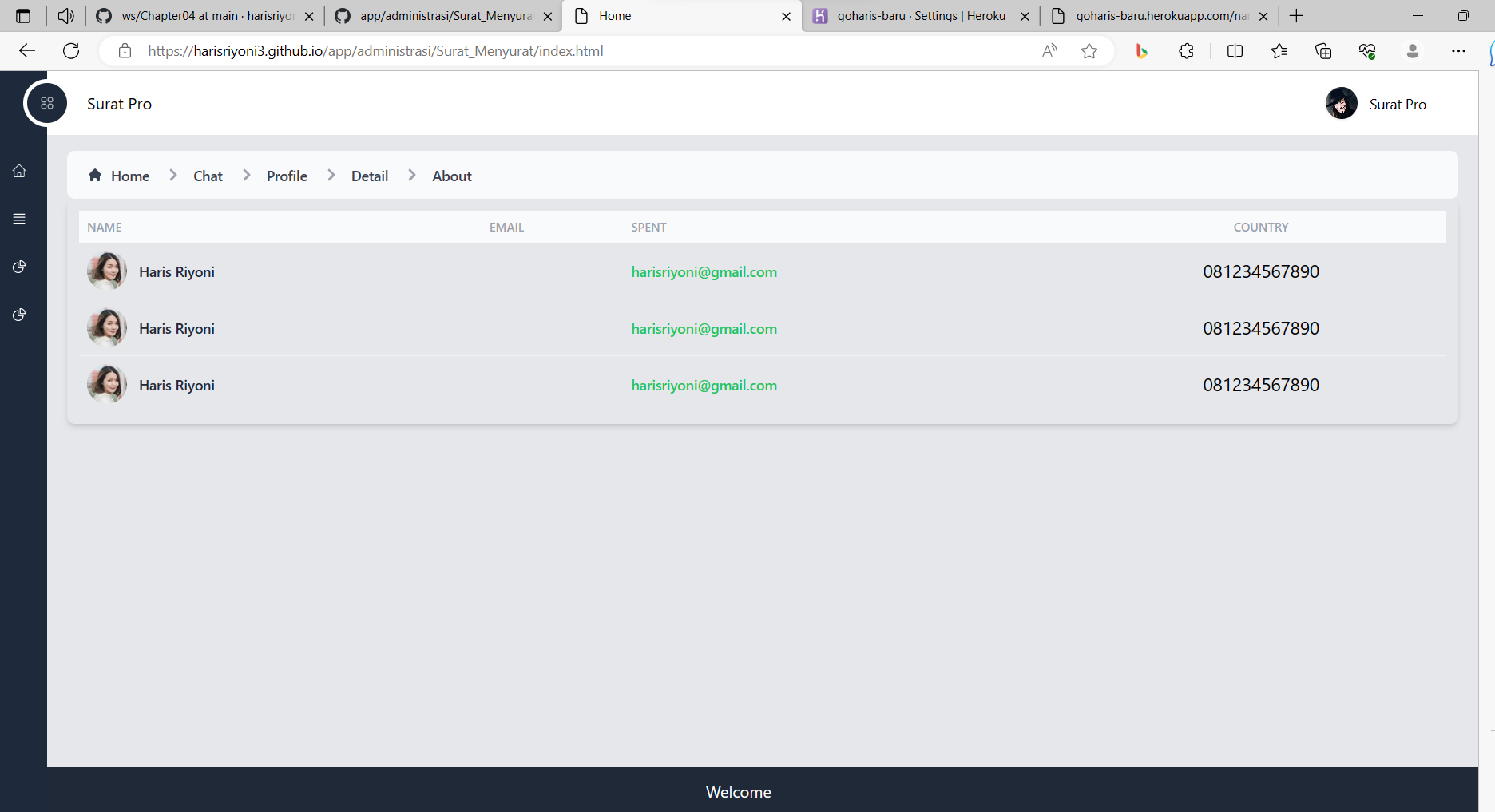
# BAB IV

# Hasil Akhir Tujuan Tampilan dan Debug

### 4.1 HASIL TAMPILAN FRONTEND

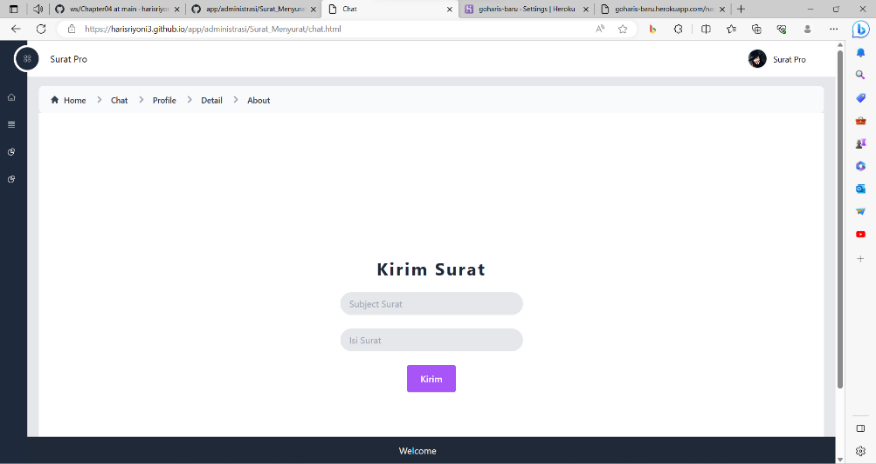
Pada Bab ini kita akan melihat hasil dari Frontend kita mulai dari Index.html dan Chat.html

* Untuk index sebagai berikut:



Gambar 4. Tampilan Index Get API GitPages

* Untuk Chat.html sebagai berikut:



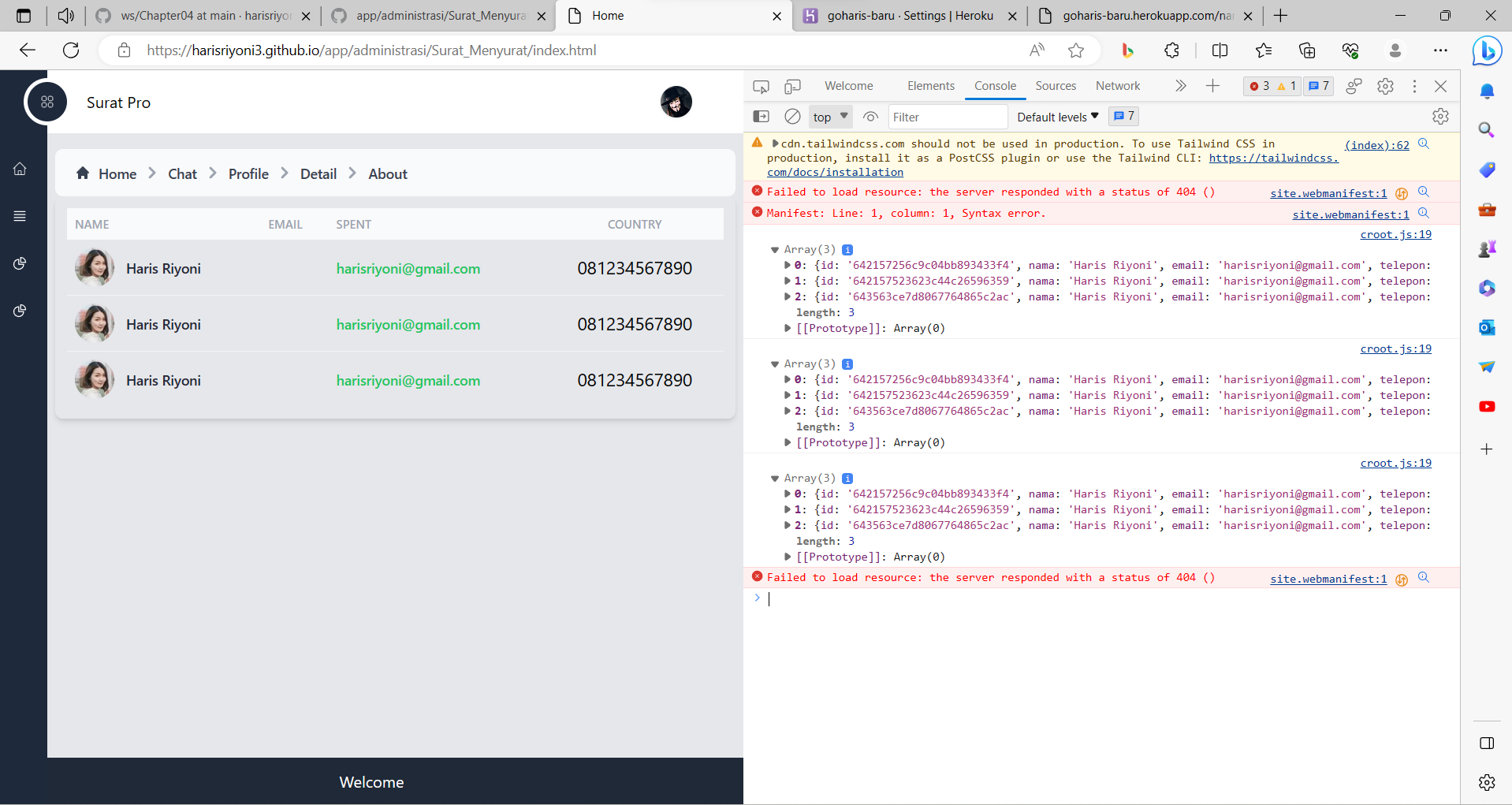
### 

Gambar 4. Tampilan Form Chat

### 4.2 DEBUG HASIL FRONTEND

Kita masuk ke bagian debug console untuk index.html Muncul dalam Console data-data yang telah di get oleh Js kita sehingga muncul di Console juga muncul di index.html.

* Untuk Index.html:

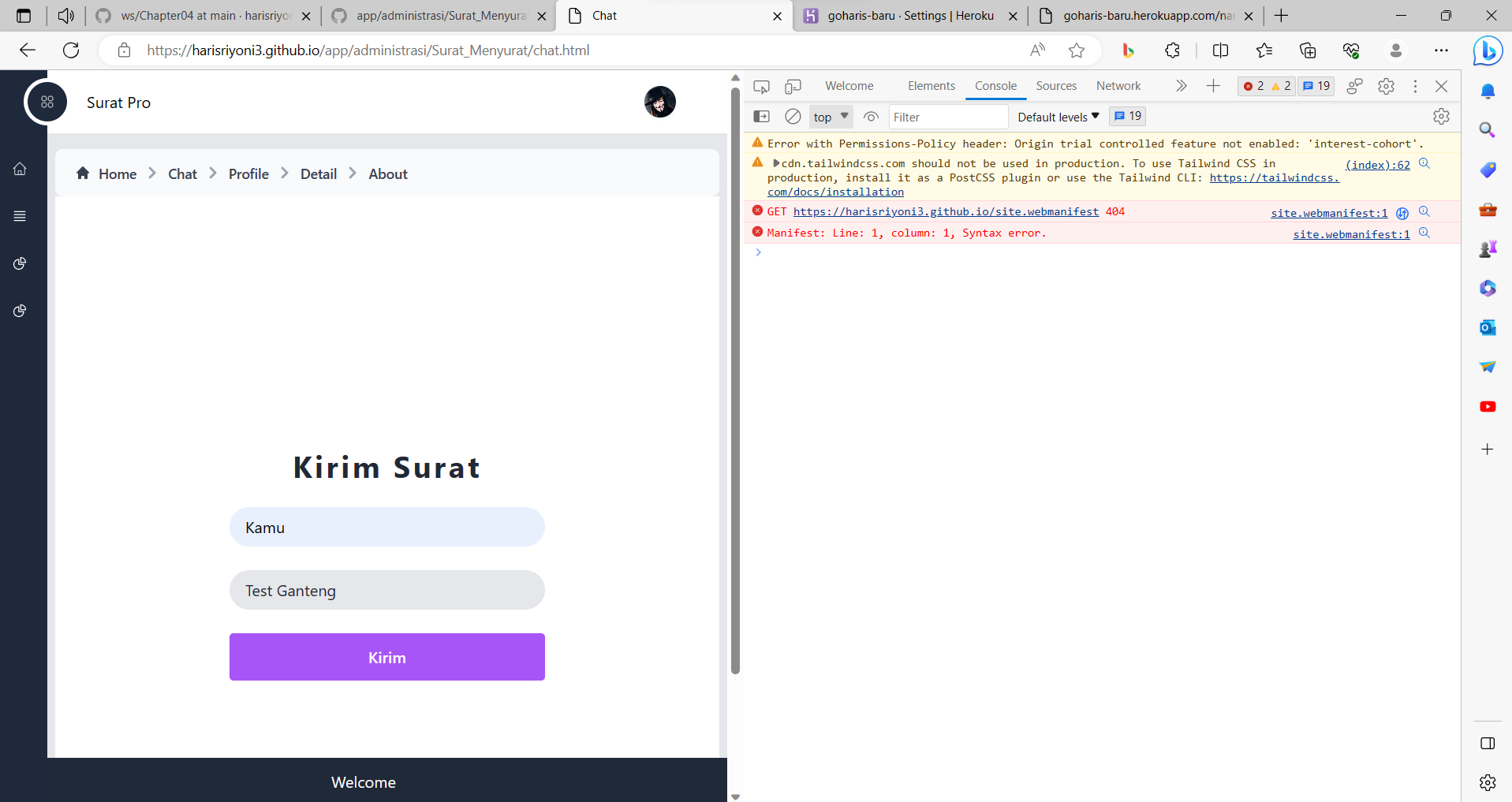


Gambar 4. Console Get API

* Untuk Chat.html

Kita masuk ke bagian debug console untuk chat.html Muncul dalam Console data-data yang telah di Post oleh Js kita sehingga muncul di Database juga muncul dalam MongoDB teman-teman juga dapat memunculkan data-data real yang baru dipost kedalam API dan ini masuk kedalam Get data API ini untuk latihan teman-teman untuk memunculkan data yang telah dipost.

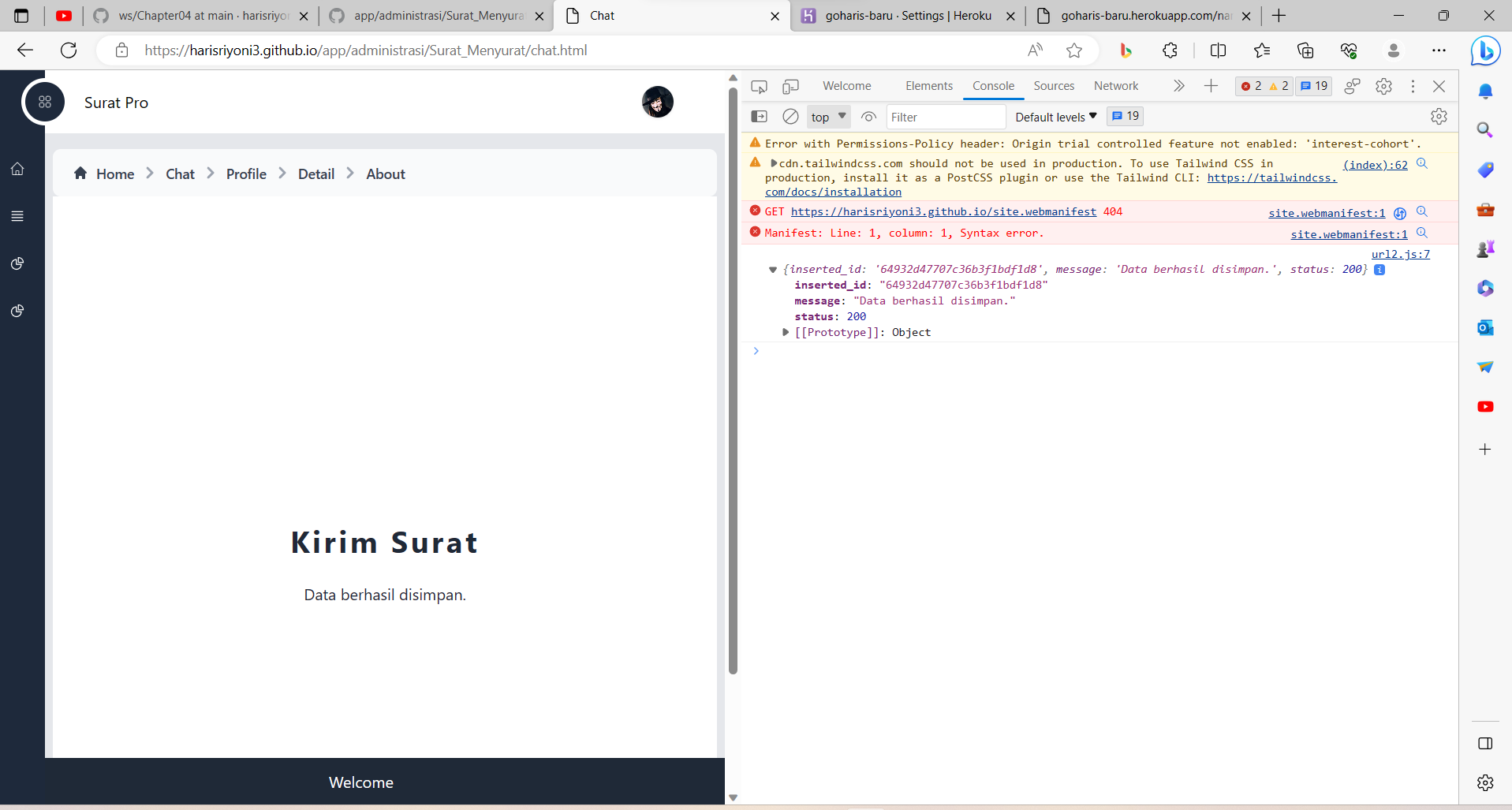
* Memasukan Field:



Gambar 4. Masukan Data Form Chat

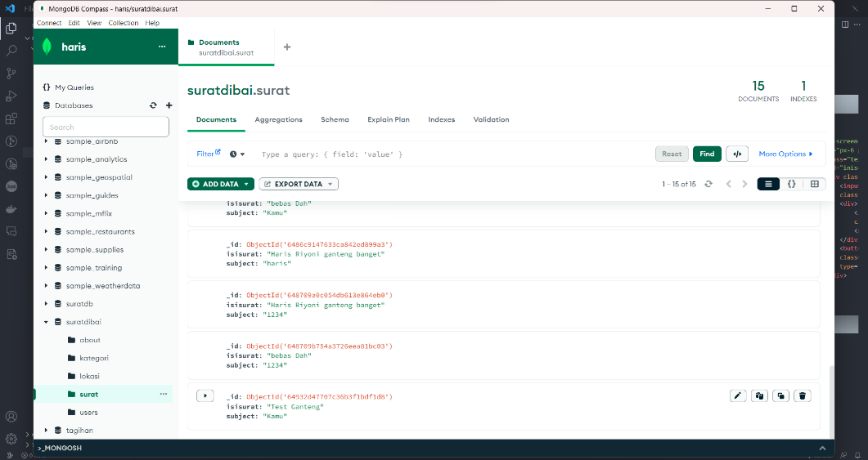
* Send Post

Terlihat di Console data Inserted ini berarti data berhasil di Post atau dikirim ke Database, kita akan cek didatabase:



Gambar 4. Form Chat Terkirim

Cek Database, dan terlihat pada gamabr dibawah ini datanya muncul “Test Ganteng” dan “Kamu”.



Gambar 4. Tampilan Hasil Send Post Form Chat

Selamat untuk teman-teman yang telah mengikuti buku ini sampai akhir dan saya ucapkan terimakasih banyak, tetap semangat dalam mempelajari dunia programmer dan sedikit Quote “Coding terasa sulit karna tak pernah dicicil karena jika dicicil pasti akan selesai sesulit apapun sulitnya Erorr”.

# DAFTAR PUSTAKA

Chabbi & Ramanathan, (2022) A study of real-world data races in Golang.

Kristanto et al., (2020) Implementasi Golang dan New Simple Queue pada Sistem Sandbox Pihak Ketiga Berbasis REST API.

Susanto, (2021) Riset Pengembangan API Generator Menggunakan Golang.

Kristanto, Harjoseputro, Samodra, et al., (2020) Golang and New Simple Queue Implementation on Third Party Sandbox System Based on REST API.

Ramadhanu & Priandika, (2021) Rancang Bangun Web Service Api Aplikasi Sentralisasi Produk Umkm Pada Uptd Plut Kumkm Provinsi Lampung.

Ardiansyah & Fatwanto, (2022) Comparison of Memory usage between REST API in Javascript and Golang.

Nascimento et al., (2021) JavaScript API Deprecation Landscape: A Survey and Mining Study.

Guthals, (2023) GitHub For Dummies.

Irawan, (2021) Implementasi Arsitektur Microservice Untuk Input Nilai Praktikum Mahasiswa Stmik Akakom Yogyakarta Menggunakan Restful Api.

Cahyono & others, (2022) Pengembangan Backend Pada Startup Sajiloka.

Sari & Hidayat, (2022) Designing website vaccine booking system using golang programming language and framework react JS.

Tsoukalos, (2019) Mastering Go: Create Golang production applications using network libraries, concurrency, machine learning, and advanced data structures.

# GLOSARIUM

A

**Table**: Bisa dari struct entity atau bisa dari database.

**Attribute**: Kolom yang berada dalam suatu table atau entity dari struct atau database.

B

**Button**: Tombol.

**Backend**: Bagian belankang dari aplikasi seperti code interaksi dengan database dll.

C

**Cloud**: Suatu Penyampaian, penyimpanan dan lain lain melalui layanan melalui internet.

**Commit**: Yang mendekteksi perubahan yg terjadi.

**Cache**: Proses penyimpanan sementara data.

**Command**: Sebuah perintah.

**Code editor**: Tempat menyusun, mengedit, menghapus code.

D

**Deploy:** Sama halnya seperti hosting.

**Drag**: Seret.

**Drop:** Lepas.

**Database**: Kumpulan data yang dikelola berdasarkan ketentuan tertentu, saling berhubungan sehingga mudah dalam pengelolaannya.

**Debug**: Mengetest atau melihat hasil dari aplikasi berhasil atau ada kendala dalam aplikasi.

**Dependensi**: Merapikan Struktur code project file dan folder dalam kasus ini kita untuk merapikan file dan folder dalam Bahasa Golang.

E

**Error**: Code atau Dependensi Bermasalah.

**ENV**: Environment.

F

**Frontend**: Bagian depan dari aplikasi seperti tampilan aplikasi.

**Function**: Fungsi-fungsi code yang telah dibuat.

**Field**: Attribute atau kolom.

G

**Get**: Mendapatkan.

H

**Hosting**: Mempublikasikan web atau aplikasi ke server internet atau jaringan online.

I

**Integrasi**: Menyatukan.

**Import**: Menggunakan, bisa seperti menggunakan library yang telah disediakan oleh bahasa tersebut atau dapat dari orang lain.

J

K

 L

**Library**: Sebuah kerangka code yang bisa kita manfaatkan untuk aplikasi kita.

M

**Merge**: Menggabungkan.

**Master:** Sebuah cabang di Git.

**Main:** Sebuah cabang di Git.

N

O

**Open source:** Suatu software atau kode pemrograman komputer yang dipublikasikan secara umum pada orang-orang.

P

**Push:** Mendorong atau mengirim sebuah project ke suatu repository.

**Programmer**: Sebuah profesi.

**Pull request:** Suatu permintaan untuk menggabungkan (merge) kode yang kita modifikasi dengan repositori utama atau repositori lain.

**Post**: Mengunggah.

Q

R

**Repository**: Tempat penyimpanan di Github.

S

**Source code**: Merupakan deretan kode.

**Skill:** Kemampuan.

**Sintak**: penulisan codenya.

**Solve**: Memperbaiki Code yang bermasalah.

T

U

**Upload**: Mengunggah.

V

**Verifikasi**: Menyetujui.

W

X

Y

Z

# INDEKS

**A**

Apa Itu Go? ,1

Apa Itu JS? ,2

Akun Gihtub, 4

Akun Heroku, 5

Akun dan MongoDB Instalasi, 5

**B**

**C**

**D**

Deploy Front-End dengan Github Pages, 33

Debug Hasil Front-End, 37

**E**

**F**

Fitur Pada GitHub ,13

**G**

Golang sebagai Back-End, 5

**H**

Hasil Akhir Tujuan Tampilan dan Debug, 36

Hasil Tampilan Front-End, 36

**I**

Instalasi Go, 4

**J**

Javascript tidak Memerlukan Instalasi, 4

Javascript sebagai Front-End, 6

**K**

keunggulan JS dengan Bahasa yang lain, 3

Kebutuhan Dan Logika Dalam Integrasi Go Dan Js, 4

Kebutuhan-Kebutuhan Yang Harus Di Penuhi & Tujuan, 4

**L**

**M**

Manfaat Mempelajari GO, 2

Manfaat Mempelajari JS, 3

Membuat Struct, 8

Membuat Function, 9

Membuat Test Code Function, 12

Membuat Get Data API dan Html, 29

Membuat Post Data API dan Html, 32

**N**

**O**

**P**

Pengenalan Dengan Go Dan Js, 1

Pengenalan, 1

Pembuatan Folder dan File untuk Package Backend, 7

Push Github Back-End dan Membuat Tag untuk Package Back-End, 13

Penggunaan Library Iteung, 16

Pembuatan Controller, 19

Pembuatan URL , 20

Push Github dan Deploy Heroku Iteung, 21

Pembuatan Folder, File dan Integrasi Front-End, 28

**Q**

**R**

**S**

Struktur File Backend Golang, 7

Struktur Folder Iteung, 16

Setting DB, 17

Setting CORS, 20

Setting Network Acces MongoDB Atlas Website, 26

Struktur Folder Front-End Javascript, 28

**T**

Tahap Praktek Coding Untuk Back-End Dan Front-End, 7