

PANDUAN LENGKAP CRUD DENGAN GOLANG, MONGODB, DAN GRAPHQL: MEMBANGUN SISTEM EFISIEN DAN SCALABLE DENGAN TRIO BERKEKUATAN TINGGI



RAUL MAHYA KOMARAN
MAULANA IMANULHAQ
RONI ANDARSYAH

PANDUAN LENGKAP CRUD DENGAN GOLANG, MONGODB, DAN GRAPHQL

Membangun Sistem Efisien Dan Scalable Dengan Trio Berkekuatan
Tinggi

**RAUL MAHYA KOMARAN
MAULANA IMANULHAQ NURDIANA
RONI ANDARSYAH**



PANDUAN LENGKAP CRUD DENGAN GOLANG, MONGODB, DAN GRAPHQL

Membangun Sistem Efisien Dan Scalable Dengan Trio
Berkekuatan Tinggi

Penulis:

Raul Mahya Komaran
Maulana Imanulhaq Nurdiana
Roni Andarsyah

ISBN :

Editor:

Roni Andarsyah

Penyunting:

Roni Andarsyah

Desain sampul dan Tata letak:

Raul Mahya Komaran
Maulana Imanulhaq Nurdiana
Roni Andarsyah

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 CenterJl. Sari Asih No. 54 Bandung 40151
Email : irc@poltekpos.ac.id

Cetakan Pertama, 2023

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk dan
dengan cara apapun tanpa izin tertulis dari penerbit

KATA PENGANTAR

Buku Langkah Mudah Membuat Crud Golang Dengan Menggunakan GraphQL Menggunakan Mongodb ini disusun untuk memenuhi kebutuhan data dan information tentang Golang khususnya di bagian tutorial pembuatan dashboard menggunakan GraphQL.

Dalam buku ini, kami akan memandu Anda melalui langkah-langkah praktis dalam membangun aplikasi menggunakan bahasa Golang dan antarmuka GraphQL. Anda akan mempelajari konsep dasar GraphQL, seperti skema, tipe data, dan resolver, serta bagaimana mengimplementasikan operasi CRUD dengan MongoDB.

Sehingga buku ini dapat memenuhi pembaca untuk membuat dashboard secara cepat simple dan sederhana. Berikut link github ini berisi tentang Langkah Mudah Membuat Crud Golang Dengan Menggunakan GraphQL Menggunakan Mongodb secara singkat padat dan jelas pada (<https://github.com/bukped/LANGKAH-MUDAH-MEMBUAT-CRUD-GOLANG-DENGAN-MENGGUNAKAN-GRAPHQL-MENGGUNAKAN-MONGO-DB>)

Bandung, 8 Juni 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
BAB 1.....	1
APA ITU GOLANG?	1
1. Definisi Golang.....	1
2. Fungsi golang.....	1
3. Kelebihan golang	2
BAB II.....	3
SEJARAH GOLANG.....	3
BAB III.....	5
GOLANG VS RUST VS PYTHON	5
Apa Itu Python?	5
1. Sejarah Python:.....	5
2. Kelebihan Python:.....	5
3. Kekurangan Python:.....	6
4. Fungsi Python:	6
Apa Itu Rust?.....	7
1. Sejarah Rust:	7
2. Kelebihan Rust:	7
3. Kekurangan Rust:	8
4. Fungsi Rust:.....	8
Apa Itu Golang?	8

1. Sejarah Go:.....	9
2. Kelebihan Go:.....	9
3. Kekurangan Go:.....	9
4. Fungsi Go:	10
BAB IV	11
APA ITU VUE JS?	11
Apa Itu VUE JS?	11
5+ Kelebihan Vue JS	11
1. Ukurannya Ringan	11
2. Dokumentasi yang Lengkap	11
3. Mudah Diintegrasikan	11
4. Mudah Dipahami	12
5. Bisa Berinteraksi Dua Arah.....	12
6. Fleksibel	12
7. Banyak Plugin	12
8. Banyak Tema yang Menarik	13
Fitur Vue JS	14
1. Computed Properties.....	14
2. Event Handling	14
3. Components.....	14
4. Data Binding.....	14
5. Virtual DOM	14
6. Lightweight	14
7. Routing.....	14
8. Watchers	15
9. Directives	15
10. Templates.....	15

BAB V	16
CARA INSTALL VUE JS DI WINDOWS	16
CARA INSTALASI	16
Langkah 1: Download dan Install Node JS	16
Langkah 2: Cek Instalasi Node JS	17
Langkah 3: Install Vue JS	18
Langkah 4: Buat Project Vue JS	19
BAB VI	22
CARA INSTALL GOLANG	22
Apa Itu Golang?	22
1. Kelebihan Go.....	22
2. INSTALASI GOLANG (STABLE & UNSTABLE)	22
B. INSTALASI GO DI MACOS	23
C. INSTALASI GO DI LINUX.....	24
3. VARIABEL GOROOT	24
4. INSTALASI GO UNSTABLE/DEVELOPMENT	24
5. GO MODULES.....	24
6. PENJELASAN.....	25
7. INISIALISASI PROJECT MENGGUNAKAN GO MODULES	25
8. Environtment Variable	26
9. GOPATH DAN WORKSPACE	27
10. VARIABEL GOPATH	27
11. SETUP WORKSPACE.....	28
BAB VII BERIKUT ADALAH PERBANDINGAN KOMPREHENSIF ANTARA TIGA BAHASA PEMROGRAMAN POPULER: RUST, GOLANG (GO), DAN PYTHON.	30
1. Kinerja.....	30
2. Keselamatan	30

3. Kemudahan Pengembangan	31
4. Ekosistem	31
5. Penggunaan	32
6. Sumber Daya	32
7. Modul Sintaks	33
8. Kecepatan	33
BAB VIII	35
EXTENSION YANG HARUS DI SIAPKAN	35
1. Live Server	35
2. Quokka.js	35
3. Code Spell Checker	36
4. GitLens	37
5. Prettier	37
6. VSCode Icons	38
7. Debugger for Chrome	38
8. ESLint	39
9. Live Saas Compiler	39
10. Browser Preview	40
11. Path Intellisense	40
12. Auto Close Tag	41
13. Auto Rename Tag	41
14. Open in Browser	42
15. Highlight Matching Tag	42
Kesimpulan	43
BAB IX MODUL GOLANG	44
Modul Golang Yang Akan Digunakan	44

BAB X PERUSAHAAN BESAR YANG MENGGUNAKAN BAHASA PEMPROGRAMAN GOLANG VUE	47
BAB XI.....	48
MENGAPA HARUS MENGGUNAKAN GRAPHQL	48
BAB XII.....	52
CARA MEMBUAT CRUD.....	52
LANGKAH MUDAH MEMBUAT CRUD + FRONT END	52

DAFTAR GAMBAR

Gambar 4. 1 Tema Yang Menarik.....	13
Gambar 5. 1 Download dan Install Node JS.....	16
Gambar 5. 2 Cek Instalasi Node JS.....	17
Gambar 5. 3 Cek Version Node JS.....	18
Gambar 5. 4 Bersihkan Cache.....	18
Gambar 5. 5 Proses Install	18
Gambar 5. 6 Buat Project Vue JS pada disk D.....	19
Gambar 5. 7 Create Vue JS Folder	19
Gambar 5. 8 Vue JS Telah Sukses Dibuat	20
Gambar 5. 9 npm run serve.....	20
Gambar 5. 10 Tampilan Front End	21
Gambar 6. 1 Environtment Variable	26
Gambar 6. 2 Skema Penulisan Command.....	27
Gambar 8. 1 Live Server.....	35
Gambar 8. 2 Quokka.js	35
Gambar 8. 3 Code Spell Checker.....	36
Gambar 8. 4 GitLens	37
Gambar 8. 5 Prettier	37
Gambar 8. 6 VSCode Icons.....	38
Gambar 8. 7 ESLint.....	39
Gambar 8. 8 Live Saas Compiler	39
Gambar 8. 9 Browser Preview	40
Gambar 8. 10 Auto Close Tag.....	41
Gambar 8. 11 Auto Rename Tag	41
Gambar 8. 12 Open In Browser	42
Gambar 8. 13 Highlight Matching Tag	42
Gambar 12. 1 Folder Tugas Besar	52
Gambar 12. 2 Masuk Ke Visual Studio Code.....	52
Gambar 12. 3 Buka Terminal di VS Code	52
Gambar 12. 4 Menginput Go Mod Init	53
Gambar 12. 5 Hasil Go Mod Init	53
Gambar 12. 6 Menginput Go get github.com/99designs/gqlgen.....	53
Gambar 12. 7 Hasil Go Get	54

Gambar 12. 8 Melakukan command pada terminal	55
Gambar 12. 9 memasukan go get github.com/99designs/gqlgen.....	55
Gambar 12. 10 Melakukan go mod tidy	56
Gambar 12. 11 Melakukan ls	56
Gambar 12. 12 Menginputkan cat tools.go	57
Gambar 12. 13 Mengetikan go run github.com/99designs/gqlgen init	57
Gambar 12. 14 Masuk Ke VS Code	58
Gambar 12. 15 Hapus codingan pada file generate.....	63
Gambar 12. 16 Hasil Tampilan Pada Halaman Local Host	66

DAFTAR TABEL

Tabel 7. 1 Kinerja (Rust vs Golang vs Python)	30
Tabel 7. 2 Keselamatan (Rust vs Golang vs Python)	30
Tabel 7. 3 Kemudahan dan Pengembangan.....	31
Tabel 7. 4 Ekosistem (Rust vs Golang vs Python)	31
Tabel 7. 5 Penggunaan (Rust vs Golang vs Python)	32
Tabel 7. 6 Sumber Daya (Rust vs Golang vs Python).....	32
Tabel 7. 7 Modul Sintaks (Rust vs Golang vs Python)	33
Tabel 7. 8 Kecepatan (Rust vs Golang vs Python)	33

BAB 1

APA ITU GOLANG?

1. Definisi Golang

Golang adalah bahasa pemrograman yang dikembangkan oleh Google yang bertujuan untuk meningkatkan bahasa pemrograman yang sudah ada seperti Python, C, dll. Ini menjadikan Golang pilihan yang tepat untuk mengembangkan aplikasi baru.

Golang adalah bahasa pemrograman yang dikembangkan oleh Google yang bertujuan untuk meningkatkan bahasa pemrograman yang sudah ada seperti Python, C, dll. Ini menjadikan Golang pilihan yang tepat untuk mengembangkan aplikasi baru.

Golang adalah bahasa pemrograman sumber terbuka yang dikembangkan oleh insinyur Google Robert Griesemer, Rob Pike, dan Ken Thompson. Mudah dipelajari, sederhana, kuat, dan diketik secara statis. Pertama kali dirilis pada 10 November 2009, Golang berupaya mengumpulkan bahasa pemrograman sebelumnya untuk membuat bahasa pemrograman baru yang lebih "lengkap" untuk bekerja di situs web dan aplikasi. Golang mengintegrasikan "fitur bagus" dari kerangka kerja lama seperti Python, C, dan lainnya agar lebih praktis digunakan dengan satu bahasa pemrograman. Fitur gabungan Golang termasuk pengetikan statis dan runtime untuk bahasa seperti C#, dan sintaksis sederhana untuk bahasa seperti Python dan JavaScript.

2. Fungsi golang

Golang adalah bahasa pemrograman sumber terbuka yang dikembangkan oleh insinyur Google Robert Griesemer, Rob Pike, dan Ken Thompson. Mudah dipelajari, sederhana, kuat, dan diketik secara statis. Pertama kali dirilis pada 10 November 2009, Golang berupaya mengumpulkan bahasa pemrograman sebelumnya untuk membuat bahasa pemrograman baru yang lebih "lengkap" untuk bekerja di situs web dan aplikasi. Golang mengintegrasikan "fitur bagus" dari kerangka kerja lama seperti Python, C, dan lainnya agar lebih praktis digunakan dengan satu bahasa pemrograman. Fitur gabungan Golang termasuk pengetikan statis dan runtime untuk bahasa seperti C#, dan sintaksis sederhana untuk bahasa seperti Python dan JavaScript.

3. Kelebihan golang

1. Performa Luar Biasa Golang adalah bahasa pemrograman yang dinilai sangat lancar dan menawarkan kinerja yang lebih baik jika dibandingkan dengan bahasa pemrograman lain, seperti Python dan Java. Salah satu alasan kinerja yang lebih baik pada Golang adalah karena bahasa pemrograman satu ini diketik secara statis, sama seperti C dan C++.
2. Mudah dipelajari kesederhanaan bahasa pemrograman ini membuatnya sangat mudah dipelajari, tidak peduli pengalaman yang dimiliki seseorang dalam bidang web development. Sintaksnya bahkan lebih mudah untuk diadopsi, jika Anda memiliki pengalaman dalam menggunakan Java dan C. Tidak hanya itu, bahasa pemrograman ini juga memiliki komunitas yang besar yang akan memudahkan Anda untuk mendapatkan solusi dari forum yang berpusat pada Go/Golang.
3. Multitasking Golang akan membagi program besar menjadi tugas yang lebih kecil dan menjalankannya secara bersamaan melalui goroutine dan channels. Dengan demikian, komputer akan menjalankan tugas-tugas secara lebih cepat dan menghabiskan lebih sedikit waktu idle. Fitur ini juga menjadikan Golang ideal untuk membuat aplikasi yang dapat diskalakan dari waktu ke waktu.
4. Tidak Perlu Framework Meskipun memiliki beberapa framework, Anda dapat aplikasi maupun website menggunakan Golang tanpa bergantung pada kerangka kerja apapun! Golang adalah bahasa pemrograman yang memiliki library standar sangat baik, yang memungkinkan Anda untuk membuat aplikasi langsung di dalamnya tanpa menggunakan framework, seperti kebanyakan bahasa pemrograman. Oleh karena itu, Golang adalah bahasa pemrograman yang sangat tepat bagi Anda para developer yang bukan penggemar framework, karena library standar Golang cukup baik untuk membantu Anda dalam development.
5. Library Standar yang Cerdas Seperti yang sudah dijelaskan sebelumnya, library standar Golang sudah cukup baik dalam mendukung Anda dalam membangun website maupun aplikasi. Developer yang menggunakan bahasa pemrograman Golang dapat melakukan hampir semua hal tanpa perlu mengimport library lainnya.
6. Otorisasi Google Secara Langsung Kelebihan terakhir dari Golang adalah dukungan Google yang diberikan secara langsung. Golang adalah bahasa pemrograman yang dibuat oleh Google. Tentu saja hal ini berarti Golang mendapatkan otorisasi langsung dari Google, sehingga Anda tidak perlu mengkhawatirkan masa depan programnya.

BAB II

SEJARAH GOLANG

Go, juga dikenal sebagai Golang, adalah bahasa pemrograman sumber terbuka yang dikembangkan oleh Google pada tahun 2007 dan dirilis secara resmi pada tahun 2009. Go dikembangkan oleh Robert Griesemer, Rob Pike, dan Ken Thompson dengan tujuan menyediakan bahasa yang efisien dan mudah digunakan. dan cara yang andal untuk mengembangkan perangkat lunak besar. Berikut kisah Golang dalam 3000 kata:

Go dimulai sebagai proyek internal Google pada tahun 2007. Motivasi utama pengembangan bahasa baru ini adalah ketidakpuasan tim pengembangan dengan bahasa yang ada saat itu seperti C++ dan Java. Mereka menganggap bahasanya terlalu rumit, lambat untuk diterjemahkan, dan sulit dipertahankan. Tim pengembangan Go, yang terdiri dari Robert Griesemer, Rob Pike, dan Ken Thompson, bertujuan untuk menciptakan bahasa yang lebih sederhana dengan penekanan pada efisiensi, kecepatan kompilasi, dan kemudahan pemeliharaan. Mereka ingin membuat bahasa yang akan membantu pengembang Google melakukan tugas sehari-hari dengan lebih efisien.

Pengembangan Go dimulai dengan mempelajari bahasa yang sudah ada, termasuk C, Pascal, dan bahasa modern seperti Python dan Ruby. Tim memilih sintaks C sebagai dasar untuk Go, tetapi menyederhanakan beberapa aspeknya. Mereka juga mengadopsi konsep pengelompokan modul dari bahasa Pascal. Salah satu fitur utama yang membedakan Go dari bahasa lain adalah sistem pengumpulan sampahnya yang kuat. Tim pengembangan Go ingin menghapus kerumitan yang diasosiasikan dengan manajemen memori manual sehingga pengembang dapat fokus pada logika bisnis tanpa mengkhawatirkan pengalokasian dan pengosongan memori.

Saat mengembangkan Go, tim juga menekankan kinerja tinggi. Mereka memeriksa berbagai teknik pengoptimalan seperti penjadwalan Goroutine, penjadwalan multiprosesor yang efisien, dan pengoptimalan terjemahan.

Go juga memberikan dukungan untuk konkurensi melalui kemampuan Goroutine-nya. Goroutine adalah unit eksekusi ringan yang dikelola oleh runtime Go dan dapat berjalan bersamaan dengan goroutine lainnya. Goroutine memungkinkan pengembang menulis kode bersamaan dengan mudah dan efisien.

Pada tahun 2009, Google secara resmi merilis Go sebagai proyek sumber terbuka. Dalam beberapa tahun pertama setelah dirilis, Go mendapat banyak perhatian dari komunitas pemrograman. Banyak orang tertarik dengan desain Go yang sederhana, efisiensi, dan fitur yang bersamaan.

Sejak rilis awal, Go terus berkembang dan diperbarui dengan peningkatan fitur dan perbaikan bug. Versi stabil pertama Go, Go 1, dirilis pada Maret 2012. Saat ini, Go sudah digunakan di berbagai proyek di dalam dan di luar Google.

Go semakin populer dengan berbagai perusahaan teknologi dan komunitas pengembang yang mengadopsinya. Banyak perusahaan seperti Dropbox, Docker, SoundCloud, dan lainnya menggunakan Go untuk perangkat lunak mereka. Go juga digunakan dalam proyek open source populer seperti Kubernetes dan lain-lain. Pada tahun 2018, Google Go merilis 1.11, versi pertama yang mendukung modul. Fitur modul memungkinkan pengembang mengelola dependensi dengan lebih baik dan mempermudah penggunaan pustaka eksternal dalam proyek Go.

Go melanjutkan pertumbuhan pesatnya dengan merilis versi baru yang menghadirkan peningkatan kinerja, fitur baru, dan perbaikan keamanan. Komunitas pemrograman Go juga terus berkembang berkat forum diskusi, konferensi, dan banyak sumber belajar.

Pada saat penulisan, Go 1.17 adalah versi terbaru Go yang dirilis pada Agustus 2021. Organisasi dan individu di seluruh dunia terus menggunakan dan mengembangkan Go untuk semua jenis proyek, mulai dari aplikasi web hingga sistem besar. Dalam sejarah singkatnya, Go telah mencapai tujuannya untuk menjadi bahasa pemrograman yang kuat, mudah digunakan, dan andal. Dengan desainnya yang sederhana, dukungan konkurensi yang kuat, dan performa tinggi, Go tetap menjadi pilihan populer bagi banyak pengembang yang ingin membangun perangkat lunak yang kuat dan andal.

GOLANG VS RUST VS PYTHON

Apa Itu Python?

Python adalah bahasa pemrograman tingkat tinggi yang diciptakan oleh Guido van Rossum pada akhir tahun 1980-an. Berikut adalah sejarah, kelebihan, kekurangan, dan fungsi Python secara lengkap:

1. Sejarah Python:

Python pertama kali dikembangkan oleh Guido van Rossum di Belanda pada akhir 1980-an. Nama Python terinspirasi dari acara televisi favorit Guido, Monty Python's Flying Circus. Versi pertama Python, Python 0.9.0, dirilis pada Februari 1991. Python 2, serangkaian versi yang berkelanjutan, dirilis pada tahun 2000 dan sangat populer. Python 3, versi yang membawa perubahan terbesar, dirilis pada tahun 2008. Pembaruan ini mencakup peningkatan desain, peningkatan kinerja, dan peningkatan bahasa. Python 3 saat ini merupakan versi aktif terbesar dan diperbarui secara berkala.

2. Kelebihan Python:

- 2.1. Mudah dipelajari dan dibaca: Python memiliki sintaks yang mudah dibaca dan lebih sederhana dibandingkan dengan bahasa lain. Ini membuatnya ideal untuk pemula dan mempercepat proses pengembangan. Bahasa multi-paradigma: Python mendukung pemrograman berorientasi objek, pemrograman prosedural, dan paradigma pemrograman fungsional. Ini memungkinkan pengembang untuk memilih gaya pemrograman yang paling sesuai dengan kebutuhan mereka.
- 2.2. Dukungan multi-platform: Python berjalan di beberapa sistem operasi seperti Windows, macOS, Linux dan UNIX. Ini memungkinkan portabilitas kode yang tinggi dan memudahkan pengembang untuk bekerja di berbagai lingkungan. Ekosistem yang kuat:
- 2.3. Python memiliki perpustakaan standar yang luas dan komunitas pengembang yang besar. Ribuan modul dan pustaka pihak ketiga tersedia untuk memperluas fungsionalitas Python dan mempercepat aplikasi.
- 2.4. Fokus pada produktivitas: Python kurang menekankan pada penulisan kode dengan sintaks sederhana. Ini memungkinkan pengembang untuk lebih fokus pada pemecahan masalah daripada masalah teknis.

3. Kekurangan Python:

Kekuatan relatif: Python adalah bahasa yang ditafsirkan, yang berarti bahwa kode Python dijalankan oleh juru bahasa Python. Meskipun Python telah dioptimalkan dari waktu ke waktu, itu mungkin tidak berjalan secepat bahasa yang dikompilasi seperti C atau C++.

GIL (Kunci Penerjemah Global): GIL adalah fitur juru bahasa Python yang membatasi banyak utas Python untuk berjalan secara bersamaan. Ini dapat memengaruhi kinerja aplikasi yang sangat bergantung pada konkurensi.

4. Fungsi Python:

Pengembangan web: Python memiliki kerangka kerja web yang populer seperti Django dan Flask, yang digunakan untuk membangun situs web dan aplikasi web yang kuat dan efisien. **Ilmu data dan kecerdasan buatan:** Python memiliki pustaka yang kuat seperti NumPy, Pandas, dan TensorFlow yang memudahkan analisis data, pembelajaran mesin, dan pengembangan kecerdasan buatan.

Otomatisasi tugas: Dengan mudahnya pembuatan skrip dan penggunaan pustaka seperti Selenium, Python sering digunakan untuk otomatisasi tugas-tugas rutin seperti pengujian, pemrosesan data, dan administrasi sistem. **Pengembangan perangkat lunak umum:** Python digunakan untuk membangun berbagai jenis perangkat lunak, mulai dari aplikasi desktop hingga perangkat lunak berbasis jaringan.

Prototyping dan scripting: Python sering digunakan sebagai bahasa pemrograman untuk prototyping cepat dan scripting karena sintaksnya yang mudah, dukungan modul yang melimpah, dan pemrosesan string yang kuat. Secara keseluruhan, Python adalah bahasa pemrograman yang populer dan kuat, dengan banyak kelebihan dan fungsionalitas yang luas. Kekurangan Python dapat diatasi dengan cara memilih solusi yang tepat dan mengoptimalkan kode saat diperlukan.

Apa Itu Rust?

Rust adalah bahasa pemrograman sistem yang dikembangkan oleh Mozilla Research. Berikut adalah sejarah, kelebihan, kekurangan, dan fungsi Rust secara lengkap:

1. Sejarah Rust:

Rust dikembangkan oleh Graydon Hoare pada tahun 2006 sebagai proyek sampingan di Mozilla Research. Pada tahun 2010, Rust mulai mendapatkan perhatian yang lebih besar di komunitas pengembang setelah Mozilla mempersembahkannya di konferensi "Mozilla Summit".

Pengembangan Rust berlanjut dan pada tahun 2012, Rust 0.1 dirilis sebagai versi awal yang eksperimental. Rust mengalami iterasi dan perbaikan yang signifikan, dan pada tahun 2015, Rust 1.0 dirilis sebagai versi stabil pertama.

Sejak itu, Rust terus berkembang dengan dukungan aktif dari komunitas pengembang dan penerimaan yang semakin luas di industri perangkat lunak.

2. Kelebihan Rust:

Keamanan yang tinggi: Salah satu keunggulan utama Rust adalah keselamatan pada tingkat kompilasi. Rust menerapkan sistem pinjaman dan kepemilikan yang unik, yang memungkinkan pengawasan ketat terhadap penggunaan memori dan mencegah kesalahan umum seperti akses memori yang tidak valid dan data race. **Kinerja yang tinggi:** Rust dirancang untuk memberikan kinerja yang tinggi dan efisiensi yang serupa dengan bahasa pemrograman sistem seperti C dan C++. Rust memiliki sistem pengelolaan memori yang aman tanpa overhead dari garbage collector.

Kesederhanaan dan ekspresivitas: Rust menawarkan sintaks yang bersih dan sederhana, sehingga mudah dipelajari dan dipahami. Dukungan untuk pola pemrograman fungsional dan pemrograman berorientasi objek memungkinkan penulisan kode yang lebih ekspresif dan mudah dibaca. **Paralelisme yang aman:** Rust menyediakan fitur konkurensi yang aman melalui mekanisme pemrograman asinkron dan pemrograman berbasis aktor. Ini memungkinkan pengembangan aplikasi yang efisien secara paralel tanpa masalah yang umum terkait dengan konkurensi seperti data race.

Ekosistem yang berkembang: Meskipun Rust adalah bahasa yang relatif baru, ekosistem Rust terus berkembang dengan cepat. Terdapat perpustakaan yang kaya dan alat bantu yang banyak, yang memperluas fungsionalitas dan mempermudah pengembangan dengan Rust.

3. Kekurangan Rust:

Kurva belajar yang curam: Ada konsep yang cukup kompleks di Rust, terutama yang terkait dengan sistem kredit dan properti. Hal ini membuat pembelajaran dan pemahaman awal tentang Rust menjadi sulit bagi beberapa developer, terutama yang baru dalam pemrograman sistem.

Kurangnya pustaka dan dukungan yang tersedia: Meskipun ekosistem Rust berkembang pesat, perpustakaan dan dukungan masih lebih terbatas dibandingkan dengan bahasa pemrograman yang lebih mapan seperti C++.

Kompatibilitas dengan bahasa lain: Meskipun Rust memiliki antarmuka C yang bagus dan dapat digunakan untuk memanggil kode C yang ada, kompatibilitas dengan bahasa lain mungkin tidak selengkap dengan bahasa seperti C++.

4. Fungsi Rust:

Pengembangan perangkat lunak sistem: Rust banyak digunakan dalam perangkat lunak sistem seperti sistem operasi, pemrosesan paralel, pemrograman tingkat rendah, dan infrastruktur jaringan. Pengembangan perangkat lunak terintegrasi: Rust dapat digunakan untuk membangun perangkat lunak terintegrasi seperti driver perangkat keras, firmware, dan perangkat lunak yang menggunakan perangkat keras tertanam.

Pengembangan aplikasi pada tingkat tinggi: Meskipun Rust adalah bahasa pemrograman sistem dengan dukungan pustaka yang berkembang, ia juga digunakan untuk mengembangkan aplikasi tingkat lanjut seperti aplikasi batch, alat pengembangan, dan layanan backend. Pengembangan kecerdasan buatan: Karena kinerjanya yang tinggi dan operasi paralel yang aman, Rust digunakan dalam pengembangan sistem kecerdasan buatan, khususnya dalam pemrosesan data dan pembelajaran mesin.

Secara keseluruhan, Rust adalah bahasa pemrograman sistem berkinerja tinggi yang aman dengan ekosistem yang berkembang pesat. Berkat keamanan, efisiensi, dan kesederhanaannya, Rust menawarkan alternatif yang menarik untuk perangkat lunak sistem kritis. Terlepas dari beberapa kekurangannya, Rust terus mendapatkan popularitas dan komunitas pengembang terus berkembang.

Apa Itu Golang?

Go, juga dikenal sebagai Golang, adalah bahasa pemrograman yang dikembangkan oleh Robert Griesemer, Rob Pike, dan Ken Thompson di Google. Berikut adalah sejarah, kelebihan, kekurangan, dan fungsi Go secara lengkap:

1. Sejarah Go:

Pengembangan Go dimulai pada tahun 2007 di Google oleh tim yang dipimpin oleh Robert Griesemer, Rob Pike, dan Ken Thompson. Tujuan utama pengembangan Go adalah menciptakan bahasa pemrograman yang efisien, mudah digunakan, dan sesuai untuk pengembangan perangkat lunak skala besar. Go dirilis sebagai proyek open source pada tahun 2009, dengan versi 1.0 yang stabil dirilis pada Maret 2012. Sejak dirilis, Go telah mengalami perkembangan yang signifikan dengan peningkatan kinerja, penambahan fitur, dan pembaruan rutin oleh tim pengembang dan komunitas.

2. Kelebihan Go:

Produktivitas tinggi: Go dirancang untuk meningkatkan produktivitas pengembang dengan sintaks sederhana, dukungan konkurensi bawaan, dan waktu kompilasi yang cepat. Ini memungkinkan pengembang untuk mengembangkan perangkat lunak dengan cepat dan efisien. **Penanganan konkurensi yang mudah:** Go menawarkan konkurensi yang kuat dengan rutinitas dan saluran go. Ini memungkinkan pengembang untuk dengan mudah menulis kode bersamaan, mengatasi masalah konkurensi, dan mengaktifkan aplikasi bersamaan yang efisien.

Pertunjukan: Go menghasilkan kode yang ringan dan efisien dalam hal penggunaan memori dan performa. Gossa memiliki pengumpul sampah yang kuat yang secara otomatis mengelola alokasi dan dealokasi memori. **Dukungan Lingkungan Pengembangan:** Go menyertakan alat pengembangan yang canggih, termasuk kompiler, pemformatan kode, linter kode, dan alat manajemen ketergantungan (modul Go). Ini memudahkan pengembang untuk mengelola proyek dan memastikan kualitas kode. **Skalabilitas:** Go dirancang untuk memecahkan masalah besar. Dukungan konkurensi yang kuat memungkinkan pengembang membangun aplikasi yang dapat berjalan secara paralel dan mendukung beban kerja yang berat.

3. Kekurangan Go:

Kurangnya pustaka khusus: Meskipun Go memiliki perpustakaan standar yang lengkap, ada beberapa pustaka khusus yang mungkin kurang tersedia dibandingkan dengan bahasa pemrograman lain yang lebih mapan. Namun, ekosistem Go terus berkembang dengan banyak kontribusi dari komunitas. **Kurva pembelajaran:** Penggunaan beberapa konsep baru dalam Go, seperti goroutine dan saluran, dapat menimbulkan tantangan bagi pengembang yang tidak terbiasa dengan pemrograman konkurensi. Dibutuhkan waktu dan upaya untuk memahami dengan baik dan menggunakan konsep-konsep ini dengan benar.

4. Fungsi Go:

4.1. Pengembangan Perangkat Lunak Web:

Go banyak digunakan dalam perangkat lunak jaringan seperti server web, mikrokontroler, dan layanan back-end. Kecepatan terjemahan yang cepat, dukungan bersamaan, dan efisiensi menjadikannya pilihan populer untuk aplikasi web. Pemrograman Sistem:

Go juga cocok untuk mengembangkan program sistem seperti sistem operasi, perangkat lunak berbasis kernel, dan alat sistem. Kemampuan manajemen memori dan kinerjanya yang tinggi menjadikannya bahasa yang ideal untuk pemrograman sistem.

4.2. Pengembangan aplikasi yang cepat:

Dirancang untuk produktivitas dan kecepatan, Go sering digunakan untuk mengembangkan aplikasi yang harus dijalankan dengan cepat dan efisien, seperti B. Alat lintas platform, aplikasi cepat, dan perangkat lunak berbasis cloud. Pemrograman paralel:

Go menawarkan fitur konkurensi yang canggih seperti rutinitas go dan saluran yang memudahkan pengembangan aplikasi bersamaan dan paralel. Ini menjadikannya bahasa populer untuk pemrograman paralel, pemrosesan paralel, dan sistem terdistribusi. Secara keseluruhan, Go adalah bahasa pemrograman yang kuat, mudah digunakan, dan kaya akan fitur konkurensi. Dibangun untuk produktivitas, efisiensi, dan skalabilitas, Go adalah pilihan populer untuk berbagai jenis pengembangan perangkat lunak.

BAB IV

APA ITU VUE JS?

Apa Itu VUE JS?

Vue JS adalah library JavaScript yang digunakan untuk membuat tampilan interface sebuah website yang interaktif. Pada dasarnya, fitur utama dari Vue JS framework adalah rendering dan komposisi element. Namun, untuk membuat aplikasi yang lebih kompleks, Anda akan membutuhkan routing, state manajemen, template, build-tool, dan lain-lain.

Library Vue JS hanya fokus pada view layer. Framework ini sangat mudah diimplementasikan dan diintegrasikan dengan library lain atau dengan project yang sudah Anda buat sebelumnya. Vue JS menggunakan id, name atau class dari suatu component html.

Sebagai sebuah framework, bisa dikatakan perkembangan Vue.js cukup pesat. Bahkan, saat ini framework ini sudah digunakan banyak perusahaan besar seperti **Google** dan **Adobe**. Padahal, framework ini sendiri masih cukup baru.

5+ Kelebihan Vue JS

1. Ukurannya Ringan

Ukuran Vue.js adalah lebih ringan dibanding framework lain, hanya 33KB. Artinya, Anda tak perlu waktu lama mendownloadnya. Dengan ukuran tersebut, proses pembuatan website tentu akan lebih efisien. Hasilnya, website dapat diselesaikan dalam waktu cepat, tapi tetap dengan performa yang baik.

2. Dokumentasi yang Lengkap

Kelebihan Vue.js yaitu memiliki dokumentasi yang sangat lengkap. Hal ini akan memudahkan programmer untuk mencoba menggunakannya. Sebab, jika menemui kendala, mereka cukup membuka halaman dokumentasi online yang disediakan. Bisa dikatakan informasi yang diberikan cukup lengkap, mulai dari cara install Vue.js hingga integrasi dengan berbagai plugin. Dengan berbagai kode yang disediakan, Anda cukup copas untuk mencoba berbagai coding dengan framework ini.

3. Mudah Diintegrasikan

Kemudahan integrasi dengan aplikasi lain tentu menjadi pertimbangan saat menggunakan framework bukan? Nah, inilah kelebihan Vue.js yang lain. Karena dibangun dengan dasar JavaScript, Anda akan lebih mudah ketika melakukan

integrasi dengan aplikasi JavaScript lain. Hal ini diperlukan ketika Anda membangun website yang cukup kompleks seperti toko online. Sebagai contoh, Anda bisa mengintegrasikan website dengan sistem pembayaran otomatis (payment gateway). Jadi, ketika pembeli telah berhasil melakukan pembayaran, sistem akan melakukan konfirmasi otomatis. Dengan begitu, pembeli tidak perlu melakukan konfirmasi lagi.

4. Mudah Dipahami

Salah satu kelebihan Vue.js adalah mudah untuk dipahami. Tak heran, framework tersebut direkomendasikan untuk pemula yang ingin belajar framework JavaScript. Selain coding yang lebih singkat, strukturnya juga lebih sederhana dibandingkan framework JavaScript yang lain. Dengan demikian, lebih mudah bagi Anda untuk menguasai framework ini. Nantinya, proses pembuatan website yang Anda lakukan bisa lebih cepat.

5. Bisa Berinteraksi Dua Arah

Kelebihan Vue.js adalah kemampuan berinteraksi dua arah. Maksudnya, Anda bisa menggunakan Vue.js untuk berkomunikasi dengan kode HTML dengan mudah. Begitu pun sebaliknya. Hal ini akan membuat performa website menjadi lebih optimal karena komunikasi berjalan dengan baik. Kenapa bisa demikian? Sebab, Vue.js menggunakan konsep MVVM (Model View View Model) yang memungkinkan tampilan berkomunikasi dengan bagian proses.

Sebagai contoh, pada metode data binding. Ketika HTML melakukan pengiriman data kepada JavaScript, akan muncul respon yang diberikan. Nantinya, respon tersebut akan ditampilkan oleh HTML kembali dan dilihat oleh user melalui browser.

6. Fleksibel

Tidak bisa dipungkiri bahwa Vue.js adalah framework yang fleksibel. Anda bisa menuliskan kode HTML, CSS, JavaScript pada satu file. Namun, bukan hanya itu saja. Dengan framework ini, Anda bisa membuat template untuk digunakan berkali-kali. Dengan fleksibilitas ini juga, Vue.js cukup mumpuni untuk membangun website dengan skala kecil maupun besar.

7. Banyak Plugin

Vue.js memiliki berbagai plugin yang dapat membantu dalam pembangunan website. Mulai dari plugin untuk meningkatkan kecepatan website, meningkatkan UX, maupun untuk meningkatkan kecepatan website.

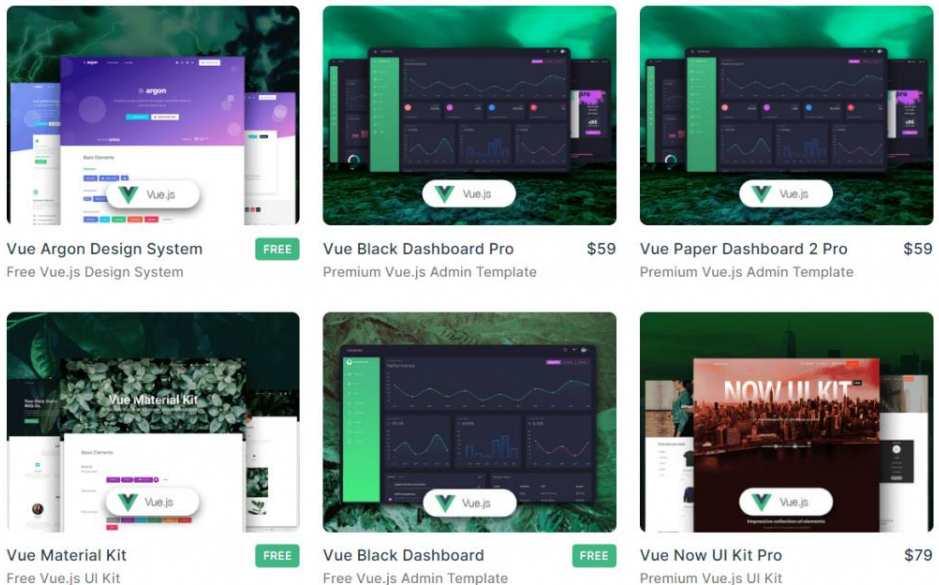
Beberapa contoh plugin yang bisa Anda manfaatkan adalah Vuetify dan Vue Router.

Vuetify adalah plugin yang menawarkan komponen yang menarik. Dengan plugin ini, Anda akan lebih mudah membuat tampilan website yang menarik meskipun tanpa skill desain yang mumpuni. Belum lagi, adanya beberapa fitur seperti dukungan RTL (right to left) dan Tree Shaking akan membuat website Anda lebih fungsional.

Sementara itu, Vue Router adalah plugin untuk membuat website Single Page Application dengan mudah. Dibekali dengan fitur seperti **view transition effects** dan **customizable scroll behavior**, website Anda akan mampu tampil unik dan menarik.

8. Banyak Tema yang Menarik

Vue.js adalah hadir dengan berbagai tema yang menarik, baik yang gratis maupun berbayar. Anda juga bisa menggunakan tema yang terang atau dark mode sesuai kebutuhan.



Gambar 4. 1 Tema Yang Menarik

Tidak hanya itu, tema yang ditawarkan juga dilengkapi dengan berbagai fitur unggulan dan komponen yang lengkap. Dengan begitu, Anda tidak perlu

membuat rancangan tampilan dari awal dan hanya perlu fokus menangani alur website saja.

Fitur Vue JS

Saat ini, Vue JS menyediakan banyak fitur yang dapat membantu developer untuk mengembangkan website. Fitur ini cukup penting untuk Anda ketahui. Mari belajar Vue JS dengan melihat fitur-fiturnya.

1. Computed Properties

Fitur ini adalah fitur yang penting dari Vue JS. Fitur computed properties sangat membantu Anda untuk melihat perubahan yang terjadi pada elemen user interface, tanpa harus melakukan penambahan kode.

2. Event Handling

Event handling (v-on) adalah atribut yang ditambahkan ke elemen DOM untuk event Vue JS. Dengan bantuan event handling, Anda bisa membuat website menjadi lebih interaktif.

3. Components

Fitur components ini sangat penting dalam Vue JS, karena dapat digunakan untuk membuat elemen kustom pada HTML. Elemen kustom ini juga dapat digunakan secara berulang-ulang.

4. Data Binding

Data binding pada Vue JS digunakan untuk memanipulasi dan memberikan values pada atribut HTML. Misalnya, mengubah style dengan bantuan binding directive dari Vue JS yaitu v-bind.

5. Virtual DOM

Document Object Model atau DOM adalah interface yang memudahkan program untuk mengakses dan menambahkan konten dalam dokumen. Vue JS sendiri memiliki Virtual DOM yang merupakan replika dari DOM. Fitur ini menyimpan perubahan dokumen di replika DOM sehingga Anda lebih dulu mereview perubahan sebelum menerapkannya di DOM. Fitur ini juga ada pada [React JS](#) dan [Ember.js](#).

6. Lightweight

Fitur ini membuat script dari Vue JS ini menjadi lebih sederhana, tetapi masih powerful dan sangat cepat.

7. Routing

Fitur ini digunakan untuk penghubung antar halaman. Fitur ini menggunakan vue-router sebagai tambahannya. Fitur ini dapat membuat website *Single Page Application* (SPA) hanya dengan menggunakan satu file HTML yang bisa memanggil semua halaman yang Anda punya.

8. Watchers

Fitur watchers dapat diterapkan pada data yang telah berubah. Contoh, bentuk elemen input. Anda tidak perlu menambahkan event tambahan pada elemen input. Karena watcher dapat menangani perubahan data apa pun yang membuat kode lebih sederhana dan cepat.

9. Directives

Vue JS memiliki perintah bawaan seperti v-if, v-else, v-show, v-on, v-bind, dan v-model, yang digunakan untuk melakukan berbagai aksi pada front-end website.

10. Templates

Vue JS memiliki template berbasis HTML yang menghubungkan DOM dengan data instance Vue JS. Vue JS mengkompilasi template menjadi fungsi DOM Render virtual.

BAB V

CARA INSTALL VUE JS DI WINDOWS

CARA INSTALASI

Hal pertama yang harus Anda lakukan sebelum belajar Vue JS adalah menginstal Node.js. Node JS adalah virtual environment untuk framework lain, termasuk Vue JS. Setelah Anda menginstal Node.js, Anda akan bertemu dengan yang namanya NPM. NPM adalah sebuah tool untuk mengatur package Java Script. NPM diibaratkan sebagai Composer versi Node.js. NPM ini sangat dibutuhkan sebelum Anda menginstal Vue JS. Untuk lebih jelasnya, berikut ini adalah langkah-langkah Install Vue JS di Windows.

Langkah 1: Download dan Install Node JS

Untuk mengunduh file installer Node JS Anda bisa klik disini. Sesuaikan tipe sistem operasi Windows Anda apakah menggunakan 64bit atau 32bit. Klik salah satu dan tunggu hingga proses download selesai.

The screenshot shows the Node.js download page. At the top, there are two tabs: 'LTS' (Recommended For Most Users) and 'Current' (Latest Features). Below the tabs, there are three main options: 'Windows Installer', 'macOS Installer', and 'Source Code'. Each option has a corresponding icon and a download link. Below this, there is a table with download links for various operating systems and architectures. The table has three columns: 'Windows Installer (.msi)', '32-bit', and '64-bit'. The rows are: 'Windows Binary (.zip)', 'macOS Installer (.pkg)', 'macOS Binary (.tar.gz)', 'Linux Binaries (x64)', 'Linux Binaries (ARM)', and 'Source Code'. The 'Source Code' row has a link to 'node-v12.16.1.tar.gz'. Red arrows point to the '32-bit' and '64-bit' columns in the first row of the table.

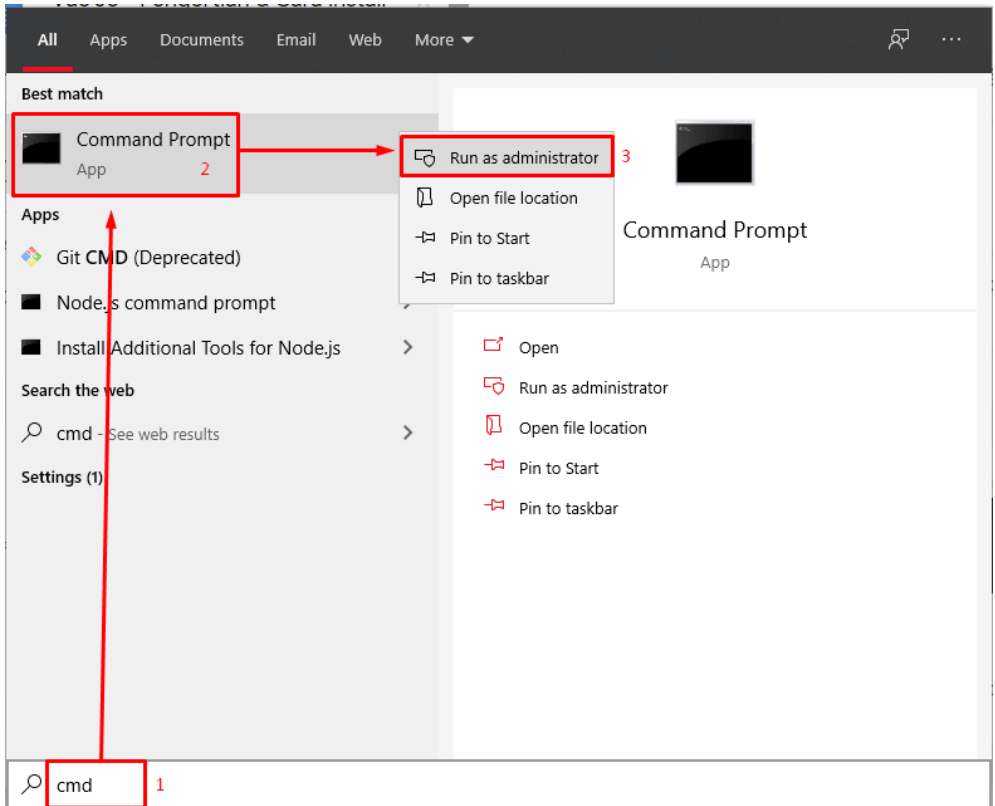
Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v12.16.1.tar.gz	

Gambar 5. 1 Download dan Install Node JS

Setelah proses download selesai, buka setup instalasi Node JS dan ikuti instruksi instalasinya. Tunggu hingga proses instalasi selesai.

Langkah 2: Cek Instalasi Node JS

Selanjutnya, cek apakah instalasi Node JS Anda berhasil atau tidak. Caranya dengan menggunakan cmd. Buka cmd dengan cara klik tombol windows lalu ketik **cmd** pada menu pencarian. Klik kanan pada aplikasi **Command Prompt** lalu pilih **Run as Administrator**. Seperti contoh gambar di bawah ini.



Gambar 5. 2 Cek Instalasi Node JS

Kemudian, masukkan perintah berikut untuk memastikan proses instalasi Node JS Anda berhasil.

```
node -v
```

Maka output dari perintah di atas akan seperti gambar di bawah ini:



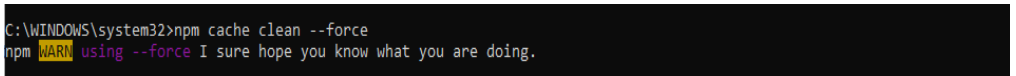
Gambar 5. 3 Cek Version Node JS

Langkah 3: Install Vue JS

Untuk menghindari terjadinya error saat proses instalasi Vue JS, bersihkan cache pada cmd menggunakan perintah berikut:

```
npm cache clean --force
```

Output dari perintah di atas akan menampilkan seperti gambar di bawah ini.



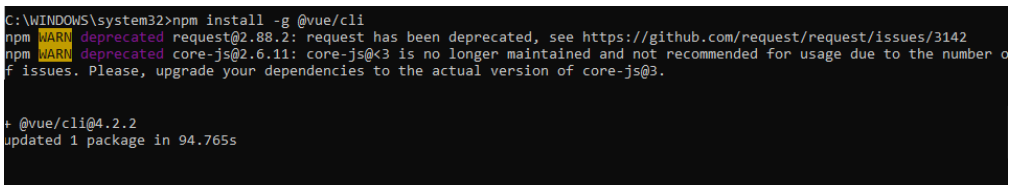
Gambar 5. 4 Bersihkan Cache

Selanjutnya install Vue JS menggunakan perintah berikut.

```
npm install -g @vue/cli
```

Pastikan saat proses install Anda terhubung ke dalam jaringan internet. Karena, proses install membutuhkan download file dari resource Vue JS yang ada di Node JS.

Tunggu hingga proses instalasi selesai, seperti contoh gambar di bawah ini:

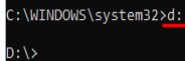


Gambar 5. 5 Proses Install

Langkah 4: Buat Project Vue JS

Setelah Anda selesai melakukan instalasi Vue JS, tidak lengkap rasanya apabila Anda tidak langsung mencoba langsung untuk membuat project menggunakan Vue JS.

Pertama, masuk ke direktori tempat Anda nanti akan menyimpan project Vue JS. Pada panduan ini menggunakan direktori Local Disk: D.



```
C:\WINDOWS\system32>cd:  
D:\>
```

Gambar 5. 6 Buat Project Vue JS pada disk D

Selanjutnya, untuk membuat project baru Vue JS Anda cukup jalankan perintah berikut ke dalam Command Prompt Anda. myproject adalah nama folder tempat menyimpan project Vue JS, Anda juga bisa menggunakan nama lain.

vue create myproject

Kemudian, Anda akan diminta untuk memilih opsi instalasi pada Vue CLI. Terdapat dua pilihan yaitu **default** dan **manually**. Pada panduan ini kami menggunakan pilihan **default**. Karena untuk menyesuaikan pengaturan project sesuai dengan Vue JS. Lalu tekan **ENTER**.



```
Administrator: Command Prompt - vue create myprojects  
Vue CLI v4.2.2  
? Please pick a preset:  
> default (babel, eslint)  
  Manually select features
```

Gambar 5. 7 Create Vue JS Folder

Setelah itu proses pembuatan project akan berjalan. Anda perlu tunggu hingga proses ini selesai. Apabila proses pembuatan project berhasil akan muncul pesan **Successfully created project “myproject”**.

```

35 packages are looking for funding
  run `npm fund` for details
[ ] Invoking generators...
[ ] Installing additional dependencies...
added 64 packages from 40 contributors in 23.313s
37 packages are looking for funding
  run `npm fund` for details
[ ] Running completion hooks...
[ ] Generating README.md...
[ ] Successfully created project myprojects.
[ ] Get started with the following commands:

$ cd myprojects
$ npm run serve

```

Gambar 5. 8 Vue JS Telah Sukses Dibuat

Selanjutnya Anda perlu masuk ke dalam direktori project. Gunakan perintah yang telah diberikan saat proses instalasi selesai. Pada panduan ini menggunakan perintah.

cd myprojects

Setelah Anda masuk ke dalam direktori project, jalankan Vue JS menggunakan perintah berikut:

npm run serve

Perintah diatas akan menghasilkan output seperti gambar di bawah ini.

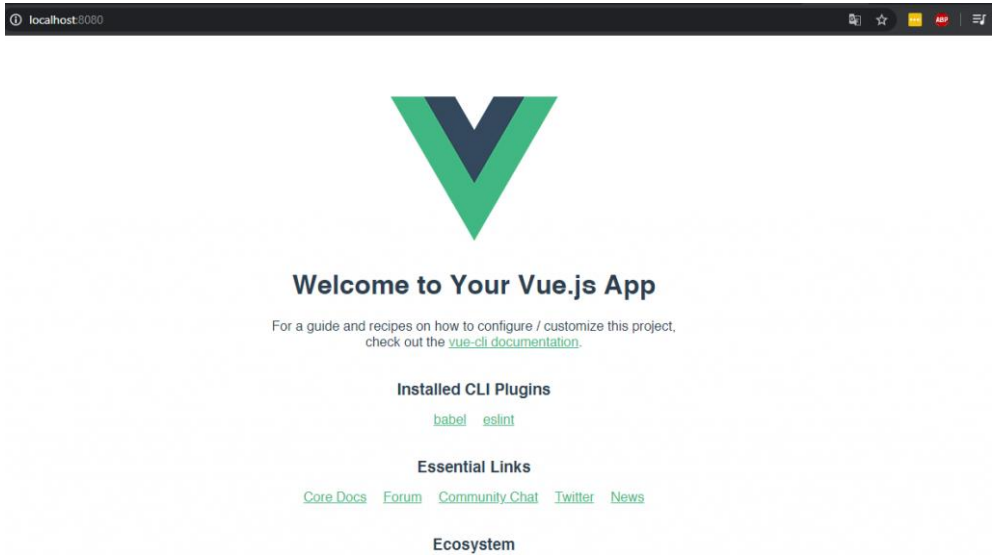
```

D:\myprojects>npm run serve
> myprojects@0.1.0 serve D:\myprojects
> vue-cli-service serve
[INFO] Starting development server...
98% after emitting CopyPlugin
[DONE] Compiled successfully in 3349ms
App running at:
- Local: http://localhost:8080/
- Network: http://192.168.1.11:8080/
Note that the development build is not optimized.
To create a production build, run npm run build.

```

Gambar 5. 9 npm run serve

Yang terakhir buka browser Anda dan akses salah satu dari IP / localhost yang ada pada gambar di atas. Maka Anda akan mendapatkan tampilan halaman dari Vue JS.



Gambar 5. 10 Tampilan Front End

Setelah itu Anda bisa melakukan proses develop front-end dari website atau aplikasi.

BAB VI

CARA INSTALL GOLANG

Apa Itu Golang?

Golang (atau biasa disebut dengan **Go**) adalah bahasa pemrograman yang dikembangkan di **Google** oleh **Robert Griesemer**, **Rob Pike**, dan **Ken Thompson** pada tahun 2007 dan mulai diperkenalkan ke publik tahun 2009. Penciptaan bahasa Go didasari bahasa **C** dan **C++**, oleh karena itu gaya sintaksnya mirip.

1. Kelebihan Go

Go memiliki kelebihan dibanding bahasa lainnya, beberapa di antaranya:

- Mendukung konkurensi di level bahasa dengan pengaplikasian cukup mudah
- Mendukung pemrosesan data dengan banyak prosesor dalam waktu yang bersamaan (*parallel processing*)
- Memiliki *garbage collector*
- Proses kompilasi sangat cepat
- Bukan bahasa pemrograman yang hirarkial dan bukan *strict OOP*, memberikan kebebasan ke developer perihal bagaimana cara penulisan kode.
- Dependensi dan *tooling* yang disediakan terbilang lengkap.
- Dukungan komunitas sangat bagus. Banyak tools yang tersedia secara gratis dan *open source* yang bisa langsung dimanfaatkan.

Sudah banyak industri dan perusahaan yg menggunakan Go sampai level production, termasuk di antaranya adalah Google sendiri, dan juga tempat di mana penulis bekerja.

2. INSTALASI GOLANG (STABLE & UNSTABLE)

Hal pertama yang perlu dilakukan sebelum bisa menggunakan Go adalah meng-*install*-nya terlebih dahulu. Panduan instalasi sebenarnya sudah disediakan di situs resmi Go <http://golang.org/doc/install#install>.

Di sini penulis mencoba meringkas petunjuk instalasi pada *link* di atas, agar lebih mudah untuk diikuti terutama untuk pembaca yang baru belajar.

Go yang digunakan adalah versi **1.20**, direkomendasikan menggunakan versi tersebut.

URL untuk mengunduh *installer* Go: <https://golang.org/dl/>. Silakan langsung unduh dari *link* tersebut lalu lakukan proses instalasi, atau bisa mengikuti petunjuk pada chapter ini.

INSTALASI GO STABLE

A. INSTALASI GO DI WINDOWS

1. Download terlebih dahulu *installer*-nya di <https://golang.org/dl/>. Pilih *installer* untuk sistem operasi Windows sesuai jenis bit yang digunakan.
2. Setelah ter-*download*, jalankan *installer*, klik *next* hingga proses instalasi selesai. *By default* jika anda tidak merubah path pada saat instalasi, Go akan ter-*install* di C:\go. *Path* tersebut secara otomatis akan didaftarkan dalam *PATH environment variable*.
3. Buka *Command Prompt* / *CMD*, eksekusi perintah berikut untuk mengecek versi Go.
4. go version
5. Jika output adalah sama dengan versi Go yang ter-*install*, menandakan proses instalasi berhasil.

Sering terjadi, command go version tidak bisa dijalankan meskipun instalasi sukses. Solusinya bisa dengan restart CMD (tutup CMD, kemudian buka lagi). Setelah itu coba jalankan ulang command di atas.

B. INSTALASI GO DI MACOS

Cara termudah instalasi Go di MacOS adalah menggunakan Homebrew.

1. *Install* terlebih dahulu Homebrew (jika belum ada), caranya jalankan perintah berikut di **terminal**.
2. `$ ruby -e "$(curl -fsSL http://git.io/pVOI)"`
3. *Install* Go menggunakan command brew.
4. `$ brew install go`
5. Tambahkan path binary Go ke *PATH environment variable*.
6. `$ echo "export PATH=$PATH:/usr/local/go/bin" >> ~/.bash_profile`
7. `$ source ~/.bash_profile`
8. Jalankan perintah berikut mengecek versi Go.
9. go version
10. Jika output adalah sama dengan versi Go yang ter-*install*, menandakan proses instalasi berhasil.

C. INSTALASI GO DI LINUX

1. Unduh arsip *installer* dari <https://golang.org/dl/>, pilih installer untuk Linux yang sesuai dengan jenis bit komputer anda. Proses download bisa dilakukan lewat CLI, menggunakan `wget` atau `curl`.
2. `$ wget https://storage.googleapis.com/golang/go1...`
3. Buka terminal, *extract* arsip tersebut ke `/usr/local`.
4. `$ tar -C /usr/local -xzf go1...`
5. Tambahkan path binary Go ke PATH *environment variable*.
6. `$ echo "export PATH=$PATH:/usr/local/go/bin" >> ~/.bashrc`
7. `$ source ~/.bashrc`
8. Selanjutnya, eksekusi perintah berikut untuk mengetes apakah Go sudah terinstal dengan benar.
9. `go version`
10. Jika output adalah sama dengan versi Go yang ter-*install*, menandakan proses instalasi berhasil.

3. VARIABEL GOROOT

By default, setelah proses instalasi Go selesai, secara otomatis akan muncul *environment variable* GOROOT. Isi dari variabel ini adalah lokasi di mana Go ter-*install*.

Sebagai contoh di Windows, ketika Go di-*install* di `C:\go`, maka path tersebut akan menjadi isi dari GOROOT.

Silakan gunakan command `go env` untuk melihat informasi konfigurasi *environment* yang ada.

4. INSTALASI GO UNSTABLE/DEVELOPMENT

Jika pembaca tertarik untuk mencoba versi development Go, ingin mencoba fitur yang belum dirilis secara official, ada beberapa cara:

- Instalasi dengan *build from source* <https://go.dev/doc/install/source>
- Gunakan command `go install`, contohnya seperti `go install golang.org/dl/go1.18beta1@latest`. Untuk melihat versi unstable yang bisa di-*install* silakan merujuk ke <https://go.dev/dl/#unstable>

5. GO MODULES

Pada bagian ini kita akan belajar cara inisialisasi project menggunakan Go Modules (atau Modules).

6. PENJELASAN

Go modules merupakan manajemen dependensi resmi untuk Go. Modules ini diperkenalkan pertama kali di go1.11, sebelum itu pengembangan project Go dilakukan dalam GOPATH.

Modules digunakan untuk menginisialisasi sebuah project, sekaligus melakukan manajemen terhadap *3rd party* atau *library* lain yang dipergunakan.

Modules penggunaannya adalah lewat CLI. Dan jika teman-teman sudah sukses meng-*install* Go, maka otomatis bisa mempergunakan Go Modules.

Modules atau Module di sini merupakan istilah untuk project ya. Jadi jangan bingung.

7. INISIALISASI PROJECT MENGGUNAKAN GO MODULES

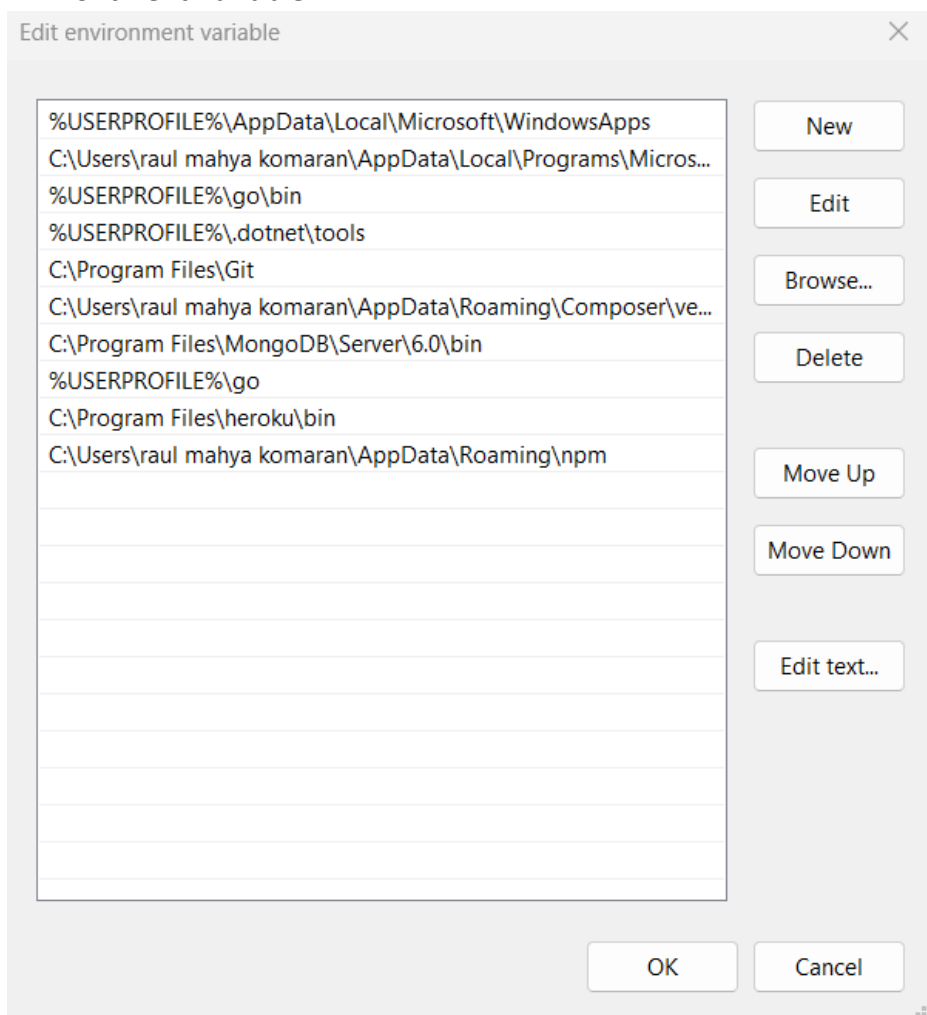
Command `go mod init` digunakan untuk menginisialisasi project baru.

Mari kita praktekan, buat folder baru, bisa via CLI atau lewat browser/finder.

```
mkdir project-pertama  
cd project-pertama  
go mod init project-pertama  
dir
```

Bisa dilihat pada *command* di atas ada direktori `project-pertama`, dibuat. Setelah masuk ke direktori tersebut, perintah `go mod init project-pertama` dijalankan. Dengan ini maka kita telah menginisialisasi direktori `project-pertama` sebagai sebuah project Go dengan nama `project-pertama` (kebetulan di sini nama project sama dengan nama direktori-nya).

8. Environment Variable



Gambar 6. 1 Environment Variable

```

C:\Windows\system32\cmd.exe

C:\Users\novalagung\Desktop>mkdir project-pertama
C:\Users\novalagung\Desktop>cd project-pertama
C:\Users\novalagung\Desktop\project-pertama>go mod init project-pertama
go: creating new go.mod: module project-pertama
C:\Users\novalagung\Desktop\project-pertama>dir
Volume in drive C has no label.
Volume Serial Number is 9CCD-195B

Directory of C:\Users\novalagung\Desktop\project-pertama

05/03/2020  02:18 PM  <DIR>          .
05/03/2020  02:18 PM  <DIR>          ..
05/03/2020  02:18 PM                32 go.mod
               1 File(s)                32-bytes
               2 Dir(s)  21,317,541,888 bytes free

```

Gambar 6. 2 Skema Penulisan Command

Skema penulisan command go mod:

go mod init <nama-project>

go mod init project-pertama

Untuk nama project, umumnya adalah disamakan dengan nama direktori, tapi bisa saja sebenarnya menggunakan nama yang lain.

Nama project dan Nama module merupakan istilah yang sama.

Eksekusi perintah `go mod init` menghasilkan satu buah file baru bernama `go.mod`. File ini digunakan oleh Go toolchain untuk menandai bahwa folder di mana file tersebut berada adalah folder project. Jadi jangan di hapus ya file tersebut.

9. GOPATH DAN WORKSPACE

PERINGATAN! Setup Go project menggunakan GOPATH kurang dianjurkan untuk Go versi terbaru. Lebih baik gunakan [A.3. Setup Go Modules](#). Tapi meski demikian, bukan berarti GOPATH tidak berguna sama sekali, jadi silakan ikuti panduan berikut jika mau.

10. VARIABEL GOPATH

GOPATH adalah variabel yang digunakan oleh Go sebagai rujukan lokasi di mana semua folder project disimpan, kecuali untuk yg diinisialisasi menggunakan Go Modules. GOPATH berisikan 3 buah sub-folder: `src`, `bin`, dan `pkg`.

Project di Go bisa ditempatkan dalam `$GOPATH/src`. Sebagai contoh anda ingin membuat project dengan nama belajar, maka harus dibuatkan sebuah folder dengan nama belajar, ditempatkan dalam src (`$GOPATH/src/belajar`). Path separator yang digunakan sebagai contoh di buku ini adalah slash `/`. Khusus pengguna Windows, path separator adalah backslash `\`.

11. SETUP WORKSPACE

Lokasi folder yang akan dijadikan sebagai workspace bisa ditentukan sendiri. Anda bisa menggunakan alamat folder mana saja, bebas, tapi jangan gunakan path tempat di mana Go ter-*install* (tidak boleh sama dengan GOROOT). Lokasi tersebut harus didaftarkan dalam path variable dengan nama GOPATH. Sebagai contoh, penulis memilih path `$HOME/Documents/go`, maka saya daftarkan alamat tersebut. Caranya:

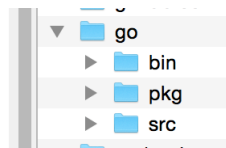
- Bagi pengguna **Windows**, tambahkan path folder tersebut ke **path variable** dengan nama GOPATH. Setelah variabel terdaftar, cek apakah path sudah terdaftar dengan benar.
Sering terjadi GOPATH tidak dikenali meskipun variabel sudah didaftarkan. Jika hal seperti ini terjadi, restart CMD, lalu coba lagi.
- Bagi pengguna Mac OS, export path ke `~/ .bash_profile`. Untuk Linux, export ke `~/ .bashrc`
- `$ echo "export GOPATH=$HOME/Documents/go" >> ~/.bash_profile`
- `$ source ~/.bash_profile`

Cek apakah path sudah terdaftar dengan benar.

```
novalagung:~ $ source ~/.bash_profile
novalagung:~ $ echo $GOPATH
/Users/novalagung/Documents/go
novalagung:~ $
```

Setelah GOPATH berhasil dikenali, perlu disiapkan 3 buah sub folder di dalamnya, dengan kriteria sebagai berikut:

- Folder `src`, adalah path di mana project Go disimpan
- Folder `pkg`, berisi file hasil kompilasi
- Folder `bin`, berisi file executable hasil build



Struktur di atas merupakan struktur standar workspace Go. Jadi pastikan penamaan dan hirarki folder adalah sama.

BAB VII

BERIKUT ADALAH PERBANDINGAN KOMPREHENSIF ANTARA TIGA BAHASA PEMROGRAMAN POPULER: RUST, GOLANG (GO), DAN PYTHON.

1. Kinerja

Tabel 7. 1 Kinerja (Rust vs Golang vs Python)

Rust	Golang	Python
Rust dikenal karena kinerjanya yang sangat tinggi. Bahasa ini dirancang untuk menghasilkan kode yang aman, konkuren, dan efisien secara default	Go juga memiliki kinerja yang baik. Meskipun tidak secepat Rust, namun Go mampu mengeksekusi kode dengan cepat, terutama dalam aplikasi konkuren.	Python memiliki kinerja yang relatif lebih lambat dibandingkan Rust dan Go. Bahasa ini lebih fokus pada kemudahan penggunaan dan keterbacaan kode, daripada kinerja yang optimal.

2. Keselamatan

Tabel 7. 2 Keselamatan (Rust vs Golang vs Python)

Rust	Golang	Python
Rust memiliki sistem peminjaman yang unik yang mencegah kesalahan umum seperti data race dan null pointer. Ini membuat Rust menjadi	Go juga menyediakan beberapa fitur keselamatan, tetapi tidak seterperinci Rust. Namun, Go memiliki rutinitas (goroutine) yang berguna untuk	Python tidak memiliki mekanisme keselamatan yang ketat seperti Rust dan Go. Meskipun demikian, dengan menggunakan praktik pemrograman

bahasa yang sangat aman secara default	mengelola konkurensi dengan aman.	yang baik, kesalahan dapat dihindari dalam Python juga.
--	-----------------------------------	---

3. Kemudahan Pengembangan

Tabel 7. 3 Kemudahan dan Pengembangan

Rust	Golang	Python
Rust memiliki kurva pembelajaran yang lebih curam. Bahasa ini sangat ketat dan menuntut disiplin dalam penulisan kode. Namun, setelah terbiasa dengan konsep-konsepnya, Rust dapat memberikan fleksibilitas dan keselamatan yang tinggi.	Go dirancang untuk menjadi mudah dipahami dan digunakan. Sintaksisnya sederhana, dan Go memiliki banyak pustaka standar yang kuat. Hal ini menjadikan Go sebagai bahasa yang mudah dipelajari oleh pemula.	Python diakui sebagai salah satu bahasa pemrograman yang paling mudah dipelajari dan dibaca. Kode Python cenderung lebih pendek dan lebih mudah dipahami dibandingkan dengan Rust dan Go.

4. Ekosistem

Tabel 7. 4 Ekosistem (Rust vs Golang vs Python)

Rust	Golang	Python
Meskipun Rust adalah bahasa yang relatif baru, ekosistemnya terus berkembang pesat. Rust memiliki sistem manajemen paket yang kuat (Cargo) dan banyak pustaka tingkat industri yang berkualitas tinggi.	Go memiliki ekosistem yang matang dan stabil. Go mempunyai sistem manajemen paket yang kuat (Go Modules) dan pustaka standar yang kaya, sehingga memudahkan pengembangan aplikasi.	Python memiliki ekosistem yang sangat luas dan matang. Ada ribuan pustaka pihak ketiga yang tersedia melalui pip, yang membuat Python sangat fleksibel dan cocok untuk hampir semua jenis proyek.

5. Penggunaan

Tabel 7. 5 Penggunaan (Rust vs Golang vs Python)

Rust	Golang	Python
Rust sering digunakan untuk pengembangan sistem yang memerlukan kinerja tinggi, keamanan yang ketat, dan konkurensi. Ini termasuk sistem operasi, perangkat lunak infrastruktur, dan pengembangan game	Go banyak digunakan dalam pengembangan web, layanan cloud, dan aplikasi konkuren. Go juga sering digunakan untuk pembuatan alat dan sistem backend.	Python sering digunakan untuk berbagai keperluan, termasuk pengembangan web, analisis data, pembelajaran mesin, dan otomatisasi tugas. Python juga sering digunakan sebagai skrip dan bahasa prototyping.

Kesimpulannya, Rust sangat cocok untuk aplikasi yang memerlukan kinerja tinggi dan keamanan yang ketat, Go cocok untuk pengembangan web dan konkurensi, sementara Python cocok untuk pemrograman umum, analisis data, dan pemrosesan skrip. Pilihan bahasa pemrograman tergantung pada kebutuhan proyek dan preferensi pengembang.

Berikut adalah perbandingan antara Rust, Golang, dan Python dalam hal sumber daya, modul sintaks, dan kecepatan:

6. Sumber Daya

Tabel 7. 6 Sumber Daya (Rust vs Golang vs Python)

Rust	Golang	Python
Rust memiliki kebutuhan sumber daya yang relatif rendah. Hal ini disebabkan oleh sistem kepemilikan yang ketat pada Rust,	Golang juga memiliki konsumsi sumber daya yang rendah. Pemrograman Golang didesain dengan fokus pada efisiensi sumber	Python biasanya membutuhkan lebih banyak sumber daya dibandingkan Rust dan Golang. Python adalah bahasa pemrograman

yang memungkinkan manajemen memori yang efisien dan penghapusan overhead alokasi atau dealokasi yang tidak perlu.	daya, dan memiliki manajemen memori otomatis yang efisien serta sistem pengumpulan sampah yang terintegrasi.	yang lebih dinamis, memiliki pengelolaan memori otomatis yang lebih kompleks, dan biasanya lebih lambat daripada bahasa dengan kompilasi yang lebih kuat seperti Rust atau Golang.
---	--	--

7. Modul Sintaks

Tabel 7. 7 Modul Sintaks (Rust vs Golang vs Python)

Rust	Golang	Python
Rust memiliki sintaks yang lebih kompleks dibandingkan dengan Golang dan Python. Bahasa ini menggunakan konsep kepemilikan, pinjaman, dan peminjaman berdasarkan aturan yang ketat. Rust juga menawarkan fitur seperti pattern matching, generics, dan sistem tipe yang kuat.	Golang memiliki sintaks yang lebih sederhana dan mudah dipahami. Bahasa ini dirancang untuk menyederhanakan pengembangan perangkat lunak dengan menggunakan sintaks yang konsisten dan intuitif. Golang juga memiliki fitur seperti goroutine untuk pemrograman konkurensi dan pemrosesan paralel.	Python terkenal dengan sintaks yang sederhana dan mudah dipelajari. Python dikenal dengan filosofi "batteries included," yang berarti bahwa banyak modul dan pustaka bawaan yang tersedia. Python juga memiliki fitur-fitur seperti dynamic typing dan duck typing yang membuatnya lebih fleksibel dan mudah digunakan.

8. Kecepatan

Tabel 7. 8 Kecepatan (Rust vs Golang vs Python)

Rust	Golang	Python
------	--------	--------

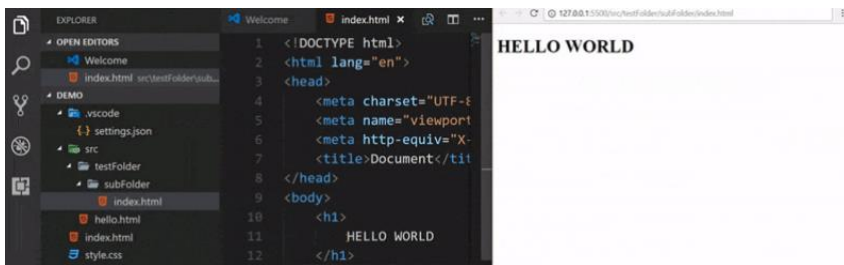
<p>Rust dikenal memiliki performa yang sangat baik dan dapat bersaing dengan bahasa pemrograman tingkat rendah seperti C atau C++.</p> <p>Rust memungkinkan pengembang untuk mengontrol secara langsung memori dan performa kode mereka, dan dengan kompilasi yang ketat, Rust dapat menghasilkan kode yang cepat dan efisien.</p>	<p>Golang juga dikenal memiliki performa yang baik. Meskipun tidak secepat Rust dalam beberapa kasus, Golang dirancang untuk memberikan keseimbangan antara kecepatan dan produktivitas. Kode Golang dikompilasi menjadi kode mesin yang efisien dan biasanya berjalan lebih cepat daripada Python.</p>	<p>Python dikenal sebagai bahasa pemrograman yang lebih lambat dibandingkan dengan Rust dan Golang. Python adalah bahasa pemrograman interpretatif yang dijalankan melalui bytecode, yang mengenakan overhead eksekusi. Namun, dengan menggunakan modul pustaka eksternal seperti NumPy dan Cython, kinerja Python dapat ditingkatkan untuk beberapa tugas terkait komputasi numerik dan pemrosesan data.</p>
--	---	---

Perlu dicatat bahwa perbandingan ini sangat umum dan tergantung pada konteks penggunaan yang spesifik. Keputusan dalam memilih bahasa pemrograman harus dipertimbangkan berdasarkan kebutuhan proyek, preferensi pengembang, dan lingkungan pengembangan yang tersedia.

BAB VIII

EXTENSION YANG HARUS DI SIAPKAN

1. Live Server

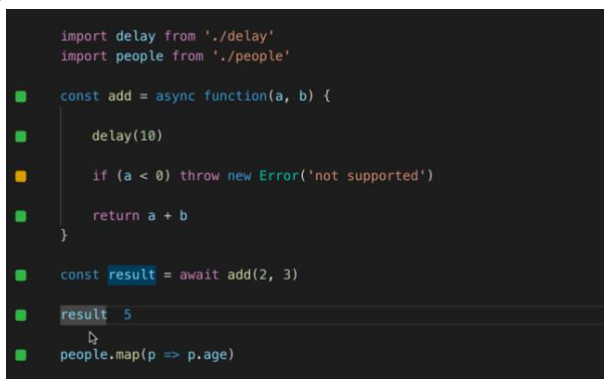


Gambar 8. 1 Live Server

Setiap kali Anda menyimpan kode, Anda akan langsung melihat perubahan yang ditampilkan di *browser*. Ini berarti jauh lebih cepat dalam menemukan kesalahan dan jauh lebih mudah untuk melakukan beberapa eksperimen cepat dengan kode Anda.

Live Server meluncurkan *server* pengembangan lokal dengan fitur *live reload* baik untuk halaman statis dan dinamis. Ini adalah salah satu ekstensi yang saya gunakan.

2. Quokka.js



Gambar 8. 2 Quokka.js

Jika Anda sedang membuat proyek dengan bahasa pemrograman *JavaScript* atau *TypeScript*, maka Anda wajib menginstal ekstensi ini. Ekstensi ini bertujuan untuk mempercepat pengembangan dengan menampilkan nilai *runtime* dalam *IDE* saat membuat kode.

Ini adalah ekstensi yang sederhana dan ringan, ini cocok untuk digunakan untuk pengembang berpengalaman dan pemula. Ini gratis untuk digunakan, tetapi jika Anda seorang profesional, Anda juga dapat membeli lisensi *Pro* yang memungkinkan Anda memodifikasi nilai *runtime* tanpa harus mengubah kode Anda.

3. Code Spell Checker

```

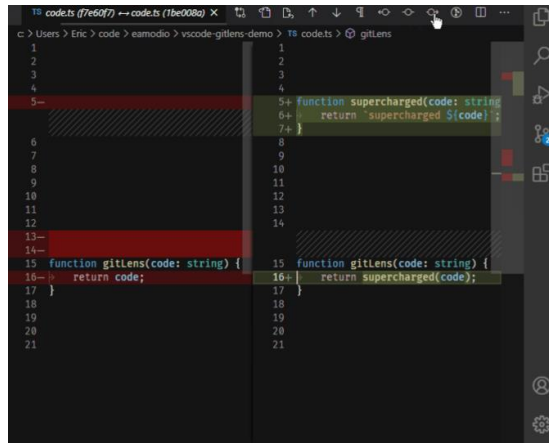
• validator.ts src
1  import {
2    TextDocument, Diagnostic, DiagnosticSeverity,
3  } from 'vscode-languageserver';
4  import { isWordInDictionary } from './spellChecker';
5  import * as Text from './util/text';
6
7  import * as Rx from 'rx';
8  import { merge } from 'tsmerge';
9
10
11  /**
12   * default constants
13   */
14  const defaultMaxNumberOfProblems = 200;
15  const defaultMinWordLength = 4;
16
17  export interface ValidationOptions {
18    maxNumberOfProblems?: number;
19    minWordLength?: number;
20    // words to always flag as an error
21    flagWords?: string[];
22  }
23
24  export function validateTextDocument(textDocument: TextDocument, options: ValidationOptions) {
25    return validateTextDocumentAsync(textDocument, options)
26      .toArray()

```

Gambar 8. 3 Code Spell Checker

Jika Anda tidak menguasai bahasa *Inggris*, maka ekstensi ini cocok digunakan untuk memeriksa kesalahan ejaan ketika membuat kesalahan ketik. Code Spell Checker menggaris bawahi kata-kata yang tidak dikenali dalam file kamusnya. Ekstensi ini tersedia dalam berbagai bahasa.

4. GitLens

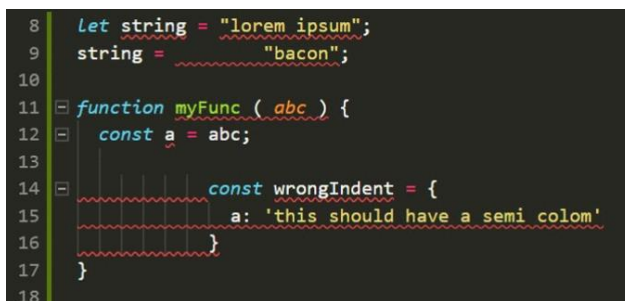


Gambar 8.4 GitLens

GitLens meningkatkan kemampuan *Git* dari *Visual Studio Code*, ini adalah ekstensi yang memungkinkan Anda untuk melihat siapa, mengapa, dan bagaimana baris kode telah berubah dari waktu ke waktu. Ini adalah ekstensi yang sangat dapat diandalkan dan disesuaikan.

Anda dapat menelusuri dan menjelajahi riwayat *file*, melihat anotasi kesalahan *git* untuk setiap baris *file*, dan bahkan menambahkan anotasi *hover* perubahan (*diff*), yang semuanya sepenuhnya dapat disesuaikan.

5. Prettier

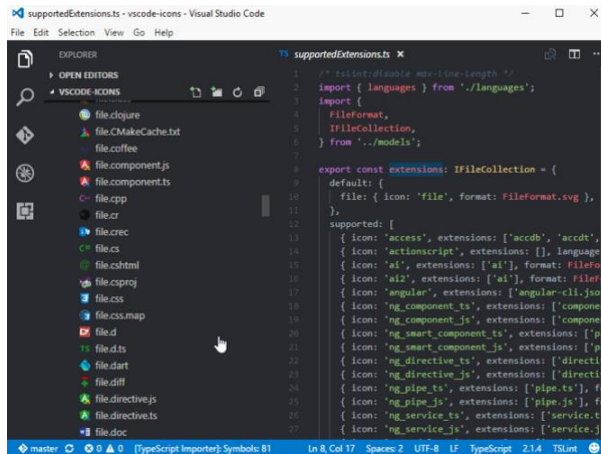


Gambar 8.5 Prettier

Prettier adalah formater kode yang bekerja sangat baik yang dapat merapikan kode yang acak-acakan menjadi lebih terstruktur dan rapi, karena ekstensi ini memberlakukan gaya yang konsisten.

Anda dapat mengaturnya sehingga memformat kode Anda setiap kali Anda menyimpannya, secara signifikan mengurangi jumlah waktu yang Anda butuhkan untuk memformat kode Anda. Ini adalah ekstensi andalan saya untuk memperbaiki kode.

6. VSCode Icons



Gambar 8. 6 VSCode Icons

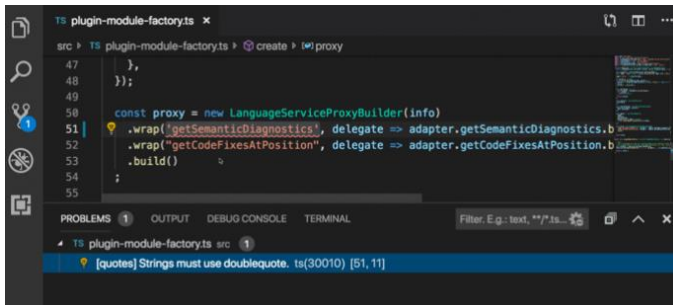
Jika Anda bosan dengan tampilan *icon* itu-itu saja, maka ekstensi ini cocok untuk menghilangkan kebosanan itu. Karena ekstensi ini dapat mengubah *icon* menjadi lebih menarik dan berbeda antara satu dengan yang lain.

7. Debugger for Chrome

Jika Anda seorang pengembang dan menggunakan bahasa pemrograman khususnya *JavaScript*. Maka ekstensi Debugger for Chrome menghemat waktu untuk menemukan banyak *bug*.

Dengan menggunakan ekstensi ini, Anda dapat mengedit kode *JavaScript* Anda langsung di *browser Chrome* atau melalui aplikasi *Visual Studio Code*. Ini akan menemukan *bug* kode mana yang bersembunyi

8. ESLint

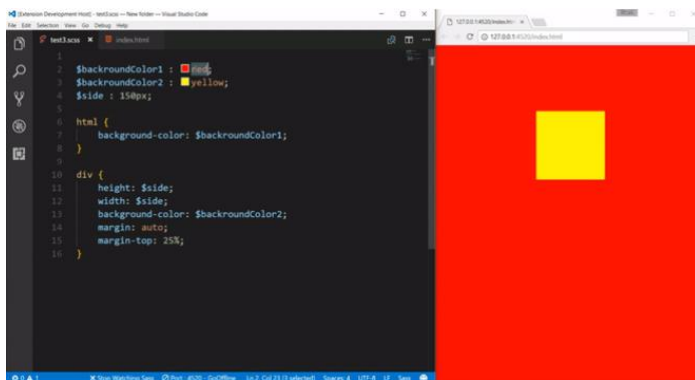


Gambar 8. 7 ESLint

Ekstensi ini mengintegrasikan ESLint ke dalam *Visual Studio Code*, dan mengharuskan Anda untuk menginstal ESLint di komputer Anda baik secara lokal maupun global.

Ekstensi ini dapat dikonfigurasi untuk memformat kode Anda secara otomatis, ini juga dapat dikonfigurasi dengan sempurna untuk menunjukkan kesalahan atau peringatan ini kepada Anda.

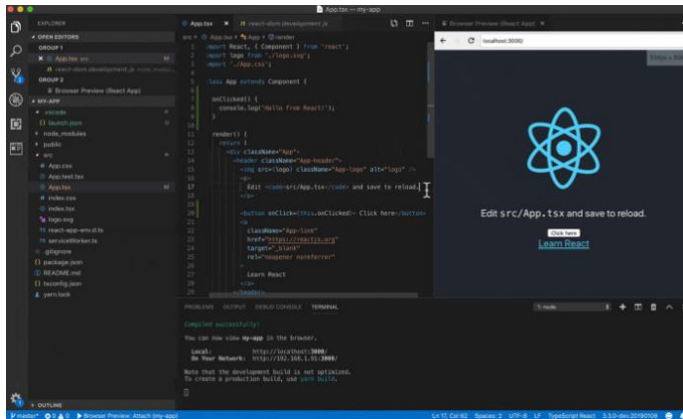
9. Live Saas Compiler



Gambar 8. 8 Live Saas Compiler

Ekstensi Visual Studio Code yang membantu Anda mengkompilasi / mentranspirasi *file SASS / SCSS* Anda ke *file CSS* secara *realtime* dengan memuat ulang browser langsung.

10. Browser Preview



Gambar 8. 9 Browser Preview

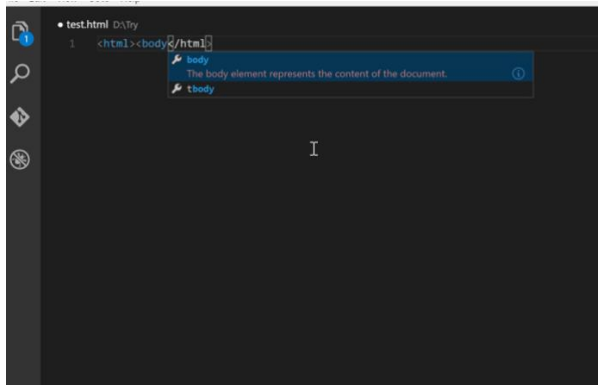
Browser Preview digunakan membuka pratinjau browser nyata di dalam editor yang dapat Anda *debug*. Browser Preview didukung oleh *Chromium* dan bekerja dengan memulai instance *Chromium* dalam proses baru.

Hal ini memungkinkan cara yang aman untuk membuat konten web di dalam *Visual Studio Code*, dan memungkinkan fitur menarik seperti *debugging in-editor* dan banyak lagi!

11. Path Intellisense

Path Intellisense membuat bekerja dengan *path*/jalur file mudah dengan menawarkan perlengkapan otomatis atau saran untuk kode Anda. Cukup mulai mengetikkan *path* dalam kutipan dan akan secara otomatis menyelesaikan jalur untuk Anda atau menyarankan berbagai opsi berbeda. Jika Anda menggunakan *file* tersembunyi, Path Intellisense dapat membantu Anda dengan itu juga.

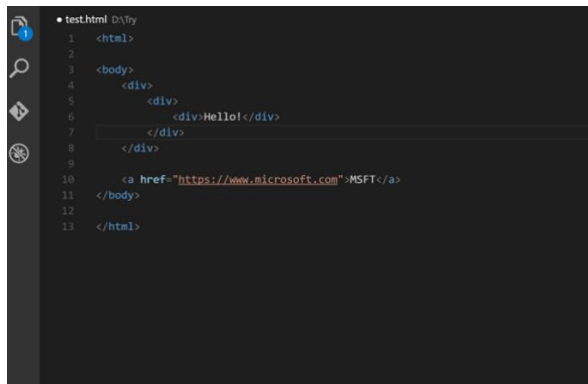
12. Auto Close Tag



Gambar 8. 10 Auto Close Tag

Seperti namanya, ekstensi ini digunakan untuk menutup *tag* *HTML* atau *XML* secara otomatis untuk *IDE*. Jadi, ketika Anda mengetikkan *tag* pembuka, ekstensi ini akan membuat *tag* penutup secara otomatis.

13. Auto Rename Tag

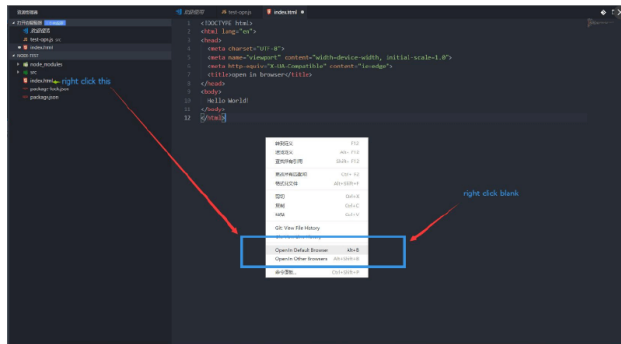


Gambar 8. 11 Auto Rename Tag

Ekstensi Auto Rename Tag menyoroti *tag* yang cocok dan secara otomatis mengganti nama *tag* yang Anda ubah. Ekstensi ini berfungsi untuk *HTML*, *XML*, *PHP*, dan *JavaScript*.

Jadi, Anda hanya perlu mengganti nama *tag* pembuka atau penutup, dan yang lainnya akan diganti namanya secara otomatis. Sederhana, tapi efektif.

14. Open in Browser



Gambar 8. 12 Open In Browser

Seperti namanya ekstensi *Visual Studio Code* digunakan untuk melihat *file HTML* di *browser*. Ini membuka halaman *HTML* di *browser default* Anda, tetapi Anda juga dapat memilih *browser* lain.

Visual Studio Code tidak mempunyai fitur untuk membuka *file* langsung di *browser*. Ekstensi ini menambahkan item **Open With Default Browser**.

15. Highlight Matching Tag



Gambar 8. 13 Highlight Matching Tag

Ekstensi ini menyoroti *tag* pembukaan dan / atau penutupan yang cocok. Secara opsional juga menunjukkan jalur untuk menandai di bilah status. Meskipun Visual Studio Code memiliki beberapa pencocokan *tag* dasar, hanya saja – dasar. Ekstensi ini akan mencoba mencocokkan *tag* di mana saja: dari atribut *tag*, bagian dalam *string*, *file* apa pun, sementara juga menyediakan opsi *styling* yang luas untuk menyesuaikan

bagaimana *tag* disorot. *Markup* yang didukung secara resmi: *HTML*, *JSX.XML*, *Vue*, *Angular*, *PHP*.

Kesimpulan

Visual Studio Code memiliki *library* atau *marketplace* yang didalamnya sangat banyak ekstensi untuk dipilih atau digunakan. Akan tetapi saya hanya mencakup 15 teratas yang direkomendasikan atau kebanyakan orang sering gunakan.

Demikian artikel **15 Ekstensi Visual Studio Code Terbaik**. Jika Anda memiliki ekstensi lain yang belum ada di daftar diatas, jangan ragu untuk memberitahu kami di komentar. Terima kasih telah membaca dan semoga bermanfaat.

BAB IX

MODUL GOLANG

Modul Golang Yang Akan Digunakan

Berikut adalah modul golang yang akan digunakan pada proses pembuatan CRUD didalam tutorial ini :

- github.com/99designs/gqlgen v0.17.27
- github.com/vektah/gqlparser/v2 v2.5.1
- go.mongodb.org/mongo-driver v1.11.3
- github.com/agnivade/levenshtein v1.1.1 // indirect
- github.com/cpuguy83/go-md2man/v2 v2.0.2 // indirect
- github.com/golang/snappy v0.0.1 // indirect
- github.com/gorilla/websocket v1.5.0 // indirect
- github.com/hashicorp/golang-lru/v2 v2.0.1 // indirect
- github.com/klauspost/compress v1.13.6 // indirect
- github.com/mitchellh/mapstructure v1.5.0 // indirect
- github.com/montanaflynn/stats v0.0.0-20171201202039-1bf9dbcd8cbe // indirect
- github.com/pkg/errors v0.9.1 // indirect
- github.com/russross/blackfriday/v2 v2.1.0 // indirect
- github.com/urfave/cli/v2 v2.24.4 // indirect
- github.com/xdg-go/pbkdf2 v1.0.0 // indirect
- github.com/xdg-go/scram v1.1.1 // indirect
- github.com/xdg-go/stringprep v1.0.3 // indirect
- github.com/xrash/smetrics v0.0.0-20201216005158-039620a65673 // indirect
- github.com/youmark/pkcs8 v0.0.0-20181117223130-1be2e3e5546d // indirect
- golang.org/x/crypto v0.0.0-20220622213112-05595931fe9d // indirect
- golang.org/x/mod v0.8.0 // indirect
- golang.org/x/sync v0.1.0 // indirect
- golang.org/x/sys v0.5.0 // indirect
- golang.org/x/text v0.7.0 // indirect
- golang.org/x/tools v0.6.0 // indirect

- `gopkg.in/yaml.v3 v3.0.1 // indirect`
BERIKUT PENJELASANNYA

Berikut adalah penjelasan tentang beberapa modul yang disebutkan dalam daftar:

1. github.com/99designs/gqlgen v0.17.27:

Modul ini adalah alat yang digunakan untuk menghasilkan kode GraphQL server dalam bahasa Go. Gqlgen menyediakan banyak fitur yang memudahkan pengembangan server GraphQL, seperti menghasilkan resolver otomatis, validasi skema, dan dokumentasi interaktif.

2. github.com/vektah/gqlparser/v2 v2.5.1:

Modul `gqlparser/v2` adalah parser GraphQL untuk bahasa pemrograman Go. Ini memungkinkan pengembang untuk mem-parsing skema GraphQL dan mengakses dan memanipulasi struktur data GraphQL.

3. go.mongodb.org/mongo-driver v1.11.3:

Modul `mongo-driver` adalah driver resmi MongoDB untuk bahasa Go. Modul ini menyediakan API untuk berinteraksi dengan MongoDB, termasuk koneksi, query, pembaruan, dan operasi lainnya.

4. github.com/agnivade/levenshtein v1.1.1:

Modul Levenshtein digunakan untuk menghitung jarak Levenshtein antara dua string. Jarak Levenshtein adalah ukuran dari perbedaan antara dua string, yang didefinisikan sebagai jumlah operasi penyisipan, penghapusan, atau penggantian karakter yang diperlukan untuk mengubah satu string menjadi string lainnya.

5. github.com/golang/snappy v0.0.1:

Modul `snappy` adalah implementasi Go dari algoritma kompresi Snappy. Snappy adalah algoritma kompresi cepat dan non-streaming yang digunakan untuk mengompresi dan mengurangi ukuran data.

6. github.com/gorilla/websocket v1.5.0:

Modul `websocket` adalah implementasi WebSocket dalam bahasa Go. Ini menyediakan fungsi-fungsi untuk berkomunikasi melalui protokol WebSocket dan membangun aplikasi real-time yang mendukung komunikasi dua arah antara klien dan server.

7. github.com/pkg/errors v0.9.1:

Modul `errors` adalah implementasi paket `errors` yang diperluas dalam bahasa Go. Modul ini menyediakan jenis error yang dapat ditingkatkan dengan informasi tambahan seperti stack trace, yang membantu dalam debugging dan melacak asal-muasal error.

8. gopkg.in/yaml.v3 v3.0.1:

Modul `yaml` adalah parser YAML yang digunakan untuk membaca dan menulis data dalam format YAML. YAML adalah format serialisasi data yang mudah dibaca oleh manusia dan sering digunakan untuk konfigurasi dan pertukaran data.

Penjelasan di atas hanya mencakup beberapa modul dari daftar yang diberikan. Modul lainnya mungkin memiliki fungsionalitas yang berbeda, seperti pemrosesan teks, enkripsi, pengujian, dan sebagainya.

BAB X

PERUSAHAAN BESAR YANG MENGGUNAKAN BAHASA PEMROGRAMAN GOLANG VUE

Ada beberapa perusahaan terkemuka yang menggunakan Golang (Go) dan Vue.js dalam pengembangan perangkat lunak mereka. Berikut adalah beberapa contoh perusahaan yang menggunakan kedua teknologi tersebut:

Google: Google menggunakan Golang untuk banyak proyek internal dan eksternal mereka. Beberapa produk Google yang menggunakan Golang antara lain Kubernetes, Docker, gRPC, dan Google Cloud Platform. Google juga menggunakan Vue.js dalam beberapa aplikasi mereka, seperti Google Cloud Console.

Netflix: Netflix adalah perusahaan hiburan digital yang menggunakan Golang untuk sistem manajemen kontennya. Mereka juga menggunakan Vue.js dalam beberapa bagian antarmuka pengguna mereka.

MENGAPA HARUS MENGGUNAKAN GRAPHQL

GolangGraphQL (gqlgen) adalah alat yang populer untuk membangun server GraphQL menggunakan bahasa pemrograman Go. Berikut adalah beberapa alasan mengapa Anda mungkin ingin mempertimbangkan menggunakan GolangGraphQL daripada solusi lain:

Kinerja yang Tinggi: Golang (Go) dikenal karena kinerjanya yang tinggi. Dengan menggunakan GolangGraphQL, Anda dapat membangun server GraphQL yang sangat responsif dan efisien. Bahasa Go secara alami memiliki performa yang baik dan dapat menangani beban tinggi dengan baik.

Mudah Digunakan dan Produktivitas Tinggi: GolangGraphQL (gqlgen) menyederhanakan pengembangan server GraphQL. Alat ini memungkinkan Anda untuk menghasilkan resolver otomatis, validasi skema, dan dokumentasi interaktif dengan sedikit usaha. Selain itu, Go memiliki sintaks yang sederhana dan mudah dipelajari, yang dapat meningkatkan produktivitas pengembangan.

Integrasi yang Baik dengan Lingkungan Go: GolangGraphQL dapat dengan mudah diintegrasikan dengan ekosistem Go yang luas. Anda dapat menggunakan pustaka dan modul Go yang ada untuk mengatasi kebutuhan spesifik Anda dalam pengembangan aplikasi GraphQL. Selain itu, Go memiliki dukungan yang baik untuk manajemen dependensi dengan Go Modules.

Tingkat Skalabilitas yang Tinggi: GolangGraphQL dirancang untuk menangani beban yang tinggi dan bersifat skalabel secara alami. Dengan pengelolaan memori yang efisien dan model konkurensi goroutine yang kuat, Anda dapat membangun server GraphQL yang dapat dengan mudah menangani banyak permintaan secara bersamaan.

Ekosistem dan Komunitas yang Aktif: Golang memiliki ekosistem dan komunitas yang berkembang pesat. Ada banyak modul, pustaka, dan alat yang tersedia untuk membantu dalam pengembangan dengan GolangGraphQL. Selain itu, Go memiliki dokumentasi yang baik dan komunitas yang aktif, yang berarti Anda dapat dengan mudah menemukan sumber daya dan dukungan yang Anda butuhkan.

Namun, penting untuk diingat bahwa pemilihan teknologi harus didasarkan pada kebutuhan proyek dan preferensi tim pengembang. Ada banyak solusi lain yang baik untuk membangun server GraphQL, seperti Node.js dengan Apollo Server, Python dengan Graphene, atau Ruby dengan GraphQL-Ruby. Pastikan untuk mempertimbangkan kebutuhan proyek, pengetahuan dan keahlian tim pengembang, serta faktor-faktor lain yang relevan sebelum membuat keputusan.

Twitch: Twitch, platform streaming langsung yang populer, menggunakan Golang dalam beberapa layanan backend mereka. Mereka juga menggunakan Vue.js dalam pengembangan antarmuka pengguna mereka.

Alibaba: Alibaba, perusahaan e-commerce terbesar di Tiongkok, menggunakan Golang dalam berbagai proyek internal. Mereka juga menggunakan Vue.js dalam beberapa proyek front-end.

Dailymotion: Dailymotion, platform berbagi video, menggunakan Golang untuk sistem backend mereka. Vue.js juga digunakan dalam beberapa bagian antarmuka pengguna mereka.

The New York Times: The New York Times menggunakan Golang dalam beberapa sistem backend mereka, termasuk sistem pengiriman konten. Mereka juga menggunakan Vue.js dalam pengembangan antarmuka pengguna.

CloudFlare: CloudFlare, penyedia layanan jaringan dan keamanan, menggunakan Golang dalam infrastruktur mereka. Vue.js juga digunakan dalam beberapa aplikasi internal.

Perlu diingat bahwa ini hanya beberapa contoh perusahaan yang menggunakan Golang dan Vue.js, dan masih banyak perusahaan lain yang menggunakan kombinasi teknologi tersebut dalam pengembangan perangkat lunak mereka.

Apa itu graphql

GraphQL adalah bahasa query dan lingkungan runtime yang digunakan untuk mengambil dan memanipulasi data dari server. Ini adalah teknologi yang dikembangkan oleh Facebook dan dirilis secara terbuka pada tahun 2015. GraphQL menyediakan antarmuka yang fleksibel dan efisien antara klien dan server, di mana klien dapat menentukan dengan tepat data apa yang mereka butuhkan dan server memberikan hanya data yang diminta tersebut.

Berikut adalah beberapa komponen utama GraphQL dan fungsi-fungsinya:

Skema (Schema):

Skema dalam GraphQL mendefinisikan struktur tipe data yang tersedia di server. Skema terdiri dari tipe-tipe objek, tipe skalar, tipe enumerasi, dan tipe lainnya. Ini menyediakan kontrak yang jelas antara klien dan server tentang data yang dapat dikirim dan diterima.

Operasi (Operation):

Operasi dalam GraphQL mengacu pada tindakan yang ingin dilakukan oleh klien, yaitu query, mutation, dan subscription.

Query: Digunakan untuk mengambil data dari server. Klien dapat mendefinisikan struktur data yang dibutuhkan dalam querynya dan server akan mengembalikan data sesuai dengan permintaan tersebut.

Mutation: Digunakan untuk mengubah atau memodifikasi data di server. Klien dapat mengirim permintaan untuk membuat, mengupdate, atau menghapus data di server.

Subscription: Digunakan untuk membuat sambungan real-time antara klien dan server, sehingga server dapat mengirimkan pembaruan data secara langsung kepada klien saat ada perubahan.

Resolver:

Resolver adalah fungsi yang bertanggung jawab untuk mengeksekusi operasi yang didefinisikan dalam skema. Setiap bidang dalam skema memiliki resolver terkait yang menentukan bagaimana data harus diambil atau diproses. Resolver ini digunakan untuk menghubungkan permintaan klien dengan sumber data yang sesuai di server.

Tipe Scalar:

GraphQL mendukung tipe data skalar seperti String, Int, Float, Boolean, dan ID. Tipe scalar memungkinkan klien dan server untuk bertukar data primitif dengan mudah.

Validasi Skema:

GraphQL memiliki mekanisme validasi skema yang kuat. Saat operasi dikirim ke server, skema akan diverifikasi untuk memastikan bahwa operasi tersebut sesuai dengan struktur dan tipe data yang telah didefinisikan.

Graf (Graph):

GraphQL menggunakan representasi data dalam bentuk graf. Dengan menggunakan graf, klien dapat dengan mudah menentukan relasi antar tipe dan mengambil data yang berkaitan.

Dalam pengembangan dengan Golang dan GraphQL, kita menggunakan paket dan pustaka Go yang telah dibuat untuk mempermudah pembangunan server GraphQL. Contohnya, gqlgen adalah alat yang populer yang memungkinkan kita untuk menghasilkan boilerplate code dan resolver otomatis berdasarkan skema GraphQL yang didefinisikan.

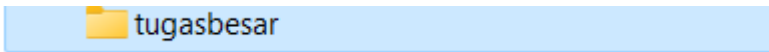
Dengan menggunakan GraphQL dalam pengembangan dengan Golang, kita dapat membangun API yang efisien, fleksibel, dan tanggap terhadap kebutuhan klien. GraphQL memungkinkan klien untuk secara presisi meminta data yang dibutuhkan dan menghindari over-fetching atau under-fetching data. Selain itu, dengan tipe-tipe yang didefinisikan dalam skema, dokumentasi yang kuat, dan validasi skema, GraphQL membantu dalam membangun dan memelihara sistem yang terdokumentasi dengan baik dan konsisten.

BAB XII

CARA MEMBUAT CRUD

LANGKAH MUDAH MEMBUAT CRUD + FRONT END

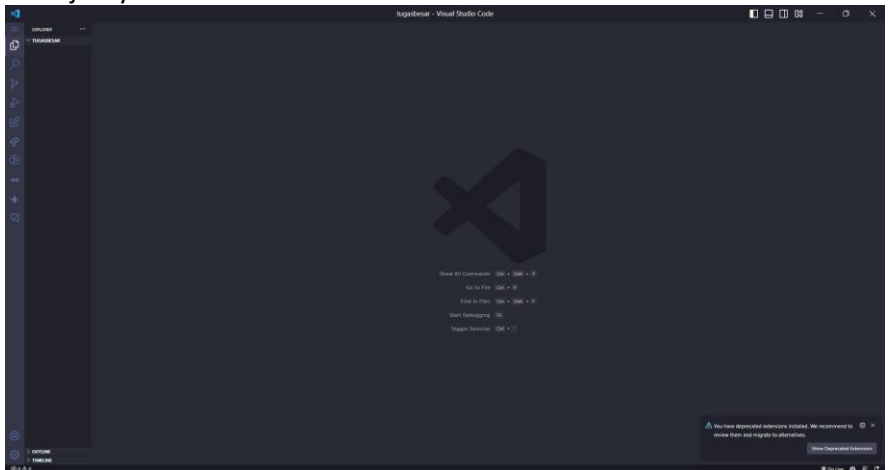
- 1) Buatlah folder



Gambar 12. 1 Folder Tugas Besar

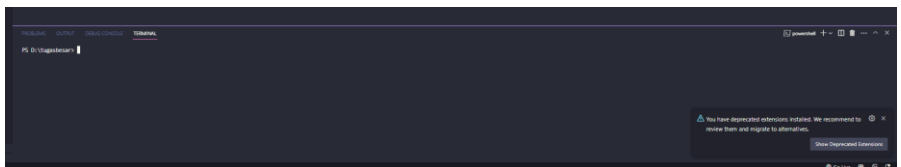
Buatlah folder seperti gambar diatas ini

- 2) Selanjutnya masukan ke VSC



Gambar 12. 2 Masuk Ke Visual Studio Code

- 3) Klik Shift + ctrl + ` bersamaan



Gambar 12. 3 Buka Terminal di VS Code

- 4) Hingga seperti di atas ini
- 5) Masukan go mod init github.com/akhil/gql-yt



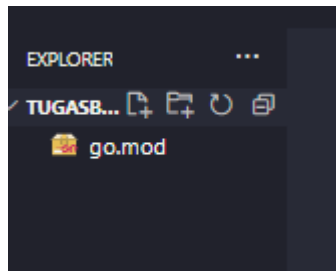
```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\tugasbesar> go mod init github.com/akhil/gql-yt
go: creating new go.mod: module github.com/akhil/gql-yt
PS D:\tugasbesar>

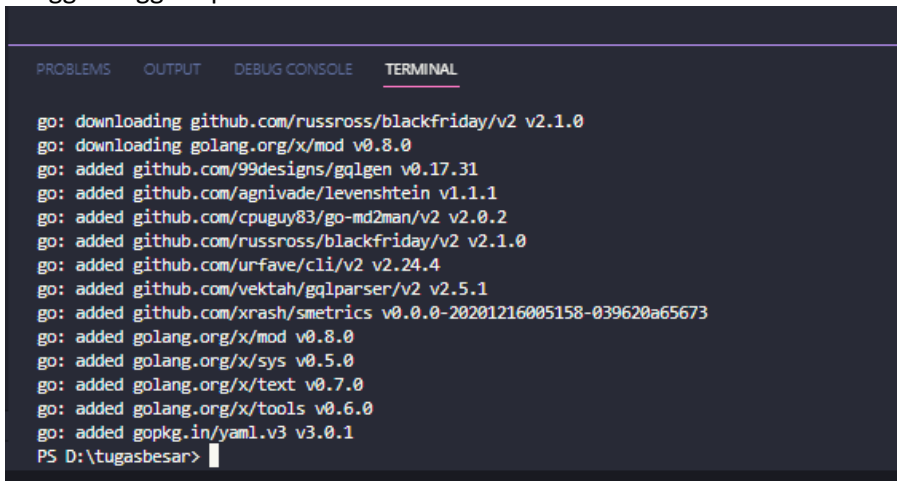
```

Gambar 12. 4 Menginput Go Mod Init



Gambar 12. 5 Hasil Go Mod Init

- 6) Go get github.com/99designs/gqlgen
Tunggu hingga seperti ini



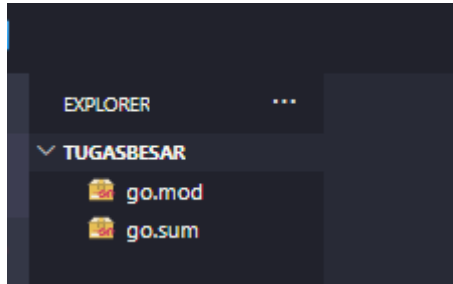
```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

go: downloading github.com/russross/blackfriday/v2 v2.1.0
go: downloading golang.org/x/mod v0.8.0
go: added github.com/99designs/gqlgen v0.17.31
go: added github.com/agnivade/levenshtein v1.1.1
go: added github.com/cpuguy83/go-md2man/v2 v2.0.2
go: added github.com/russross/blackfriday/v2 v2.1.0
go: added github.com/urfave/cli/v2 v2.24.4
go: added github.com/vektah/gqlparser/v2 v2.5.1
go: added github.com/xrash/smetrics v0.0.0-20201216005158-039620a65673
go: added golang.org/x/mod v0.8.0
go: added golang.org/x/sys v0.5.0
go: added golang.org/x/text v0.7.0
go: added golang.org/x/tools v0.6.0
go: added gopkg.in/yaml.v3 v3.0.1
PS D:\tugasbesar>

```

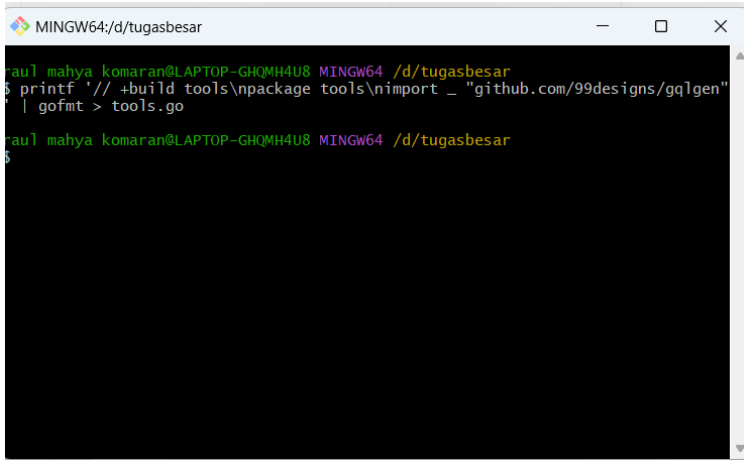
Gambar 12. 6 Menginput Go get github.com/99designs/gqlgen



Gambar 12. 7 Hasil Go Get

Berikut yang harus disiapkan dan cara menggunakannya secara sederhana

- Buat folder baru untuk Project mkdir gql-yt
 - Mod init proyek Anda, berikan nama apa pun yang Anda sukai go mod init github.com/akhil/gql-yt
 - Dapatkan gql gen untuk proyek Anda, dapatkan github.com/99designs/gqlgen
 - Tambahkan gqlgen ke tools.go printf '// +build tools\npackage tools\nimport _ "github.com/99designs/gqlgen"' | gofmt > tools.go
 - Dapatkan semua dependensi menjadi mod rapi
 - Inisialisasi proyek Anda, jalankan github.com/99designs/gqlgen init
 - Setelah Anda menulis skema graphql, jalankan ini - jalankan github.com/99designs/gqlgen generate
 - Setelah Anda membangun proyek, ini adalah kueri untuk berinteraksi dengan API -
- 7) Selanjutnya melanjutkan printf '// +build tools\npackage tools\nimport _ "github.com/99designs/gqlgen"' | gofmt > tools.go

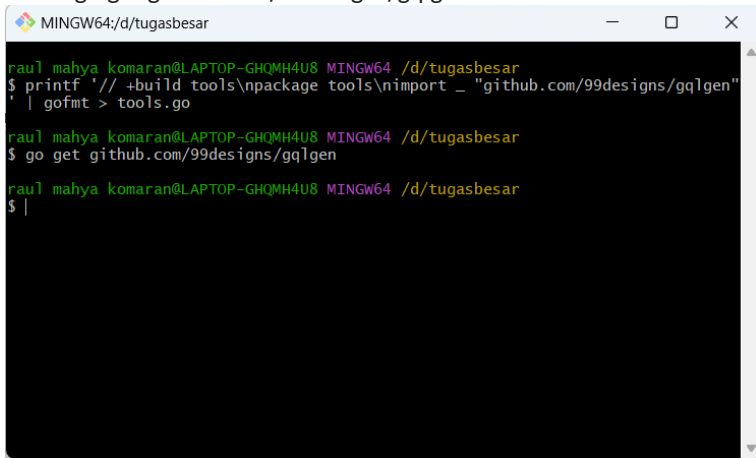
A terminal window titled 'MINGW64:/d/tugasbesar' with a black background and green text. The prompt is 'raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar'. The command entered is '\$ printf '// +build tools\npackage tools\nimport _ "github.com/99designs/gqlgen" | gofmt > tools.go'.

```
raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ printf '// +build tools\npackage tools\nimport _ "github.com/99designs/gqlgen"
| gofmt > tools.go

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$
```

Gambar 12. 8 Melakukan command pada terminal

8) Masukan go get github.com/99designs/gqlgen

A terminal window titled 'MINGW64:/d/tugasbesar' with a black background and green text. The prompt is 'raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar'. The command entered is '\$ go get github.com/99designs/gqlgen'.

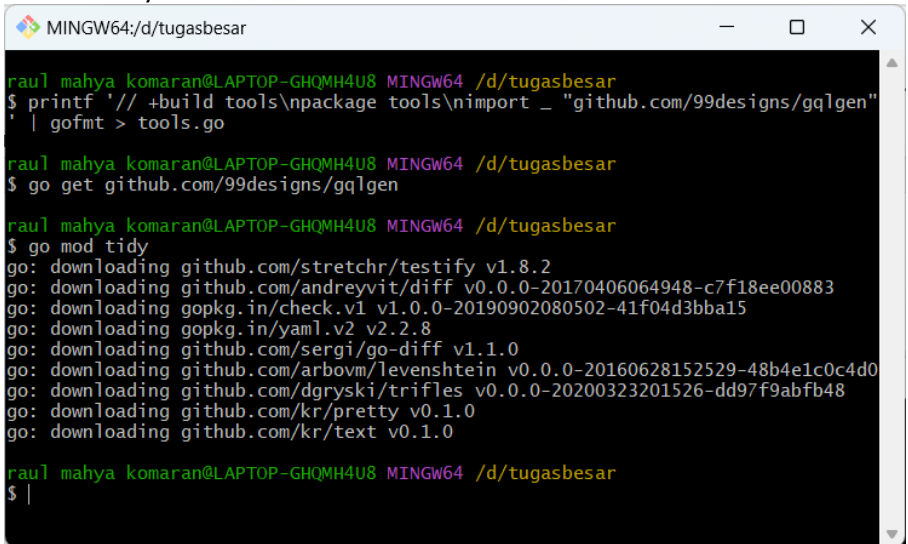
```
raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ printf '// +build tools\npackage tools\nimport _ "github.com/99designs/gqlgen"
| gofmt > tools.go

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ go get github.com/99designs/gqlgen

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ |
```

Gambar 12. 9 memasukan go get github.com/99designs/gqlgen

9) Go mod Tidy



```

MINGW64:/d/tugasbesar
raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ printf '// +build tools\npackage tools\nimport _ "github.com/99designs/gqlgen"
' | gofmt > tools.go

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ go get github.com/99designs/gqlgen

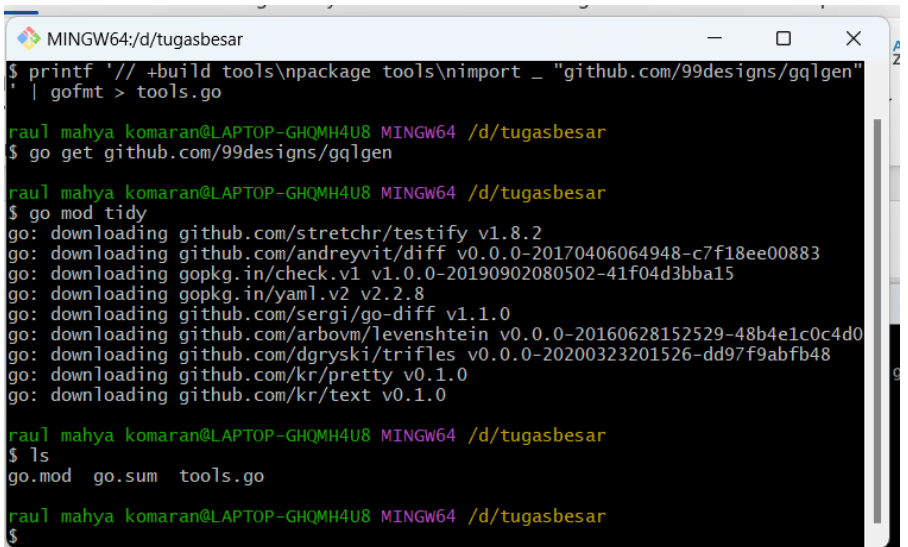
raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ go mod tidy
go: downloading github.com/stretchr/testify v1.8.2
go: downloading github.com/andybalmain/ci-go v0.0.0-20170406064948-c7f18ee00883
go: downloading gopkg.in/check.v1 v1.0.0-20190902080502-41f04d3bba15
go: downloading gopkg.in/yaml.v2 v2.2.8
go: downloading github.com/sergi/go-diff v1.1.0
go: downloading github.com/arbovm/levenshtein v0.0.0-20160628152529-48b4e1c0c4d0
go: downloading github.com/dgryski/trifles v0.0.0-20200323201526-dd97f9abfb48
go: downloading github.com/kr/pretty v0.1.0
go: downloading github.com/kr/text v0.1.0

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ |

```

Gambar 12. 10 Melakukan go mod tidy

10) Kemudian ls



```

MINGW64:/d/tugasbesar
$ printf '// +build tools\npackage tools\nimport _ "github.com/99designs/gqlgen"
' | gofmt > tools.go

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ go get github.com/99designs/gqlgen

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ go mod tidy
go: downloading github.com/stretchr/testify v1.8.2
go: downloading github.com/andybalmain/ci-go v0.0.0-20170406064948-c7f18ee00883
go: downloading gopkg.in/check.v1 v1.0.0-20190902080502-41f04d3bba15
go: downloading gopkg.in/yaml.v2 v2.2.8
go: downloading github.com/sergi/go-diff v1.1.0
go: downloading github.com/arbovm/levenshtein v0.0.0-20160628152529-48b4e1c0c4d0
go: downloading github.com/dgryski/trifles v0.0.0-20200323201526-dd97f9abfb48
go: downloading github.com/kr/pretty v0.1.0
go: downloading github.com/kr/text v0.1.0

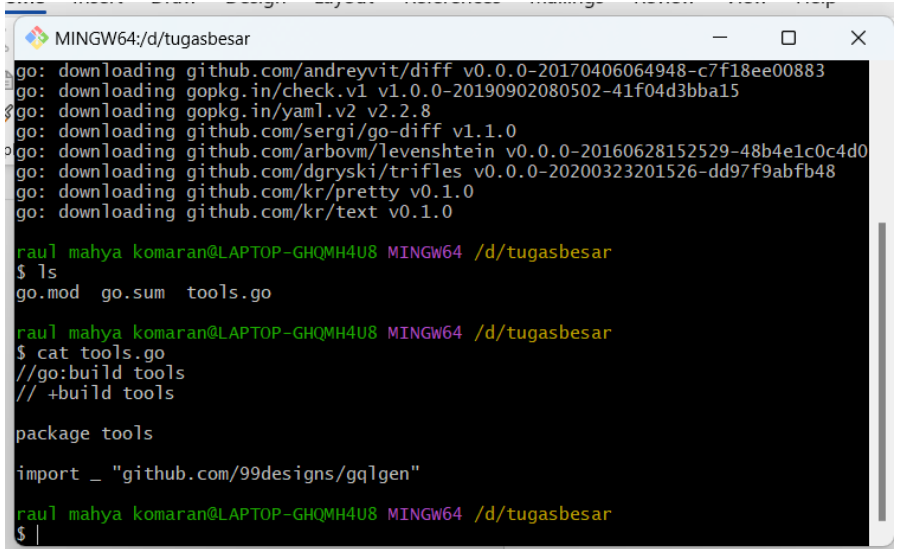
raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ ls
go.mod go.sum tools.go

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$

```

Gambar 12. 11 Melakukan ls

11) Masukan cat tools.go



```

MINGW64:/d/tugasbesar
go: downloading github.com/andreyvit/diff v0.0.0-20170406064948-c7f18ee00883
go: downloading gopkg.in/check.v1 v1.0.0-20190902080502-41f04d3bba15
go: downloading gopkg.in/yaml.v2 v2.2.8
go: downloading github.com/sergi/go-diff v1.1.0
go: downloading github.com/arbovm/levenshtein v0.0.0-20160628152529-48b4e1c0c4d0
go: downloading github.com/dgryski/trifles v0.0.0-20200323201526-dd97f9abfb48
go: downloading github.com/kr/pretty v0.1.0
go: downloading github.com/kr/text v0.1.0

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ ls
go.mod go.sum tools.go

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ cat tools.go
//go:build tools
// +build tools

package tools

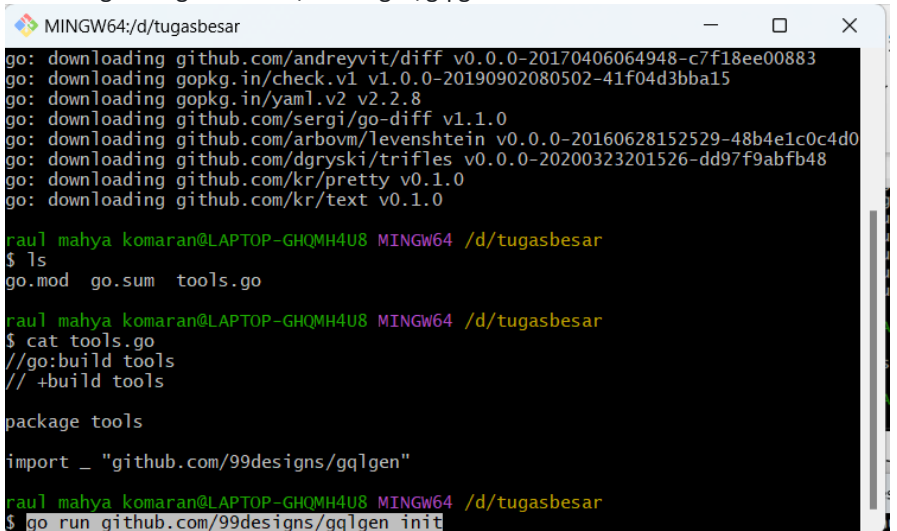
import _ "github.com/99designs/gqlgen"

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ |

```

Gambar 12. 12 Menginputkan cat tools.go

12) Ketikan go run github.com/99designs/gqlgen init



```

MINGW64:/d/tugasbesar
go: downloading github.com/andreyvit/diff v0.0.0-20170406064948-c7f18ee00883
go: downloading gopkg.in/check.v1 v1.0.0-20190902080502-41f04d3bba15
go: downloading gopkg.in/yaml.v2 v2.2.8
go: downloading github.com/sergi/go-diff v1.1.0
go: downloading github.com/arbovm/levenshtein v0.0.0-20160628152529-48b4e1c0c4d0
go: downloading github.com/dgryski/trifles v0.0.0-20200323201526-dd97f9abfb48
go: downloading github.com/kr/pretty v0.1.0
go: downloading github.com/kr/text v0.1.0

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ ls
go.mod go.sum tools.go

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ cat tools.go
//go:build tools
// +build tools

package tools

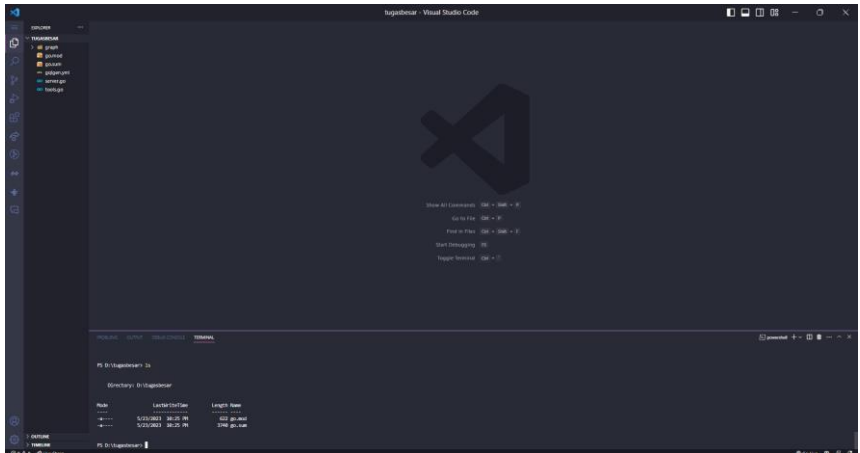
import _ "github.com/99designs/gqlgen"

raul mahya komaran@LAPTOP-GHQM4U8 MINGW64 /d/tugasbesar
$ go run github.com/99designs/gqlgen init

```

Gambar 12. 13 Mengetikan go run github.com/99designs/gqlgen init

13) Ketikan code .



Gambar 12. 14 Masuk Ke VS Code

14) Bualah folder dengan nama database dan isi dari folder database itu Namanya database.go
Masukan ini

```
package database
```

```
import (
    "context"
    "log"
    "time"

    "github.com/akhil/gql-yt/graph/model"
    "go.mongodb.org/mongo-driver/bson"
    "go.mongodb.org/mongo-driver/bson/primitive"
    "go.mongodb.org/mongo-driver/mongo"
    "go.mongodb.org/mongo-driver/mongo/options"
    "go.mongodb.org/mongo-driver/mongo/readpref"
)
```

```
var connectionString string =
    "mongodb://raulgantengbanget:0nGCVIPPoCsXNhqG@ac-oilbpwk-shard-00-00.9ofhjs3.mongodb.net:27017,ac-oilbpwk-shard-00-01.9ofhjs3.mongodb.net:27017,ac-oilbpwk-shard-00-02.9ofhjs3.mongodb.net:27017/test?replicaSet=atlas-13x7kp-shard-0&ssl=true&authSource=admin"
```

```

type DB struct {
    client *mongo.Client
}

func Connect() *DB {
    client, err :=
mongo.NewClient(options.Client().ApplyURI(connectionString))
    if err != nil {
        log.Fatal(err)
    }
    ctx, cancel := context.WithTimeout(context.Background(),
30*time.Second)
    defer cancel()
    err = client.Connect(ctx)
    if err != nil {
        log.Fatal(err)
    }
    err = client.Ping(ctx, readpref.Primary())
    if err != nil {
        log.Fatal(err)
    }
    return &DB{
        client: client,
    }
}

func (db *DB) GetJob(id string) *model.JobListing {
    jobCollec := db.client.Database("graphql-job-board").Collection("jobs")
    ctx, cancel := context.WithTimeout(context.Background(),
30*time.Second)
    defer cancel()

    _id, _ := primitive.ObjectIDFromHex(id)
    filter := bson.M{"_id": _id}
    var JobListing model.JobListing
    err := jobCollec.FindOne(ctx, filter).Decode(&JobListing)
    if err != nil {
        log.Fatal(err)
    }
}

```

```

    return &JobListing
}

func (db *DB) GetJobs() []*model.JobListing {
    jobCollec := db.client.Database("graphql-job-board").Collection("jobs")
    ctx, cancel := context.WithTimeout(context.Background(),
30*time.Second)
    defer cancel()
    var JobListings []*model.JobListing
    cursor, err := jobCollec.Find(ctx, bson.D{})
    if err != nil {
        log.Fatal(err)
    }
    if err = cursor.All(context.TODO(), &JobListings); err != nil {
        panic(err)
    }
    return JobListings
}

func (db *DB) CreateJobListing(jobInfo model.CreateJobListingInput)
*model.JobListing {
    jobCollec := db.client.Database("graphql-job-board").Collection("jobs")
    ctx, cancel := context.WithTimeout(context.Background(),
30*time.Second)
    defer cancel()
    inserted, err := jobCollec.InsertOne(ctx, bson.M{
        "title":    jobInfo.Title,
        "description": jobInfo.Description,
        "url":      jobInfo.URL,
        "company":  jobInfo.Company,
    })
    if err != nil {
        log.Fatal(err)
    }
    InsertedID := inserted.InsertedID.(primitive.ObjectID).Hex()
    returnJobListing := model.JobListing{ID: InsertedID,
        Title:    jobInfo.Title,
        Company:  jobInfo.Company,
        Description: jobInfo.Description,
        URL:      jobInfo.URL,
    }
}

```



```

    return &returnJobListing
}
func (db *DB) UpdateJobListing(jobId string, jobInfo
model.UpdateJobListingInput) *model.JobListing {
    jobCollec := db.client.Database("graphql-job-board").Collection("jobs")
    ctx, cancel := context.WithTimeout(context.Background(),
30*time.Second)
    defer cancel()

    updateJobInfo := bson.M{}
    if jobInfo.Title != nil {
        updateJobInfo["title"] = jobInfo.Title
    }
    if jobInfo.Description != nil {
        updateJobInfo["description"] = jobInfo.Description
    }
    if jobInfo.URL != nil {
        updateJobInfo["url"] = jobInfo.URL
    }
    _id, _ := primitive.ObjectIDFromHex(jobId)
    filter := bson.M{"_id": _id}
    update := bson.M{"$set": updateJobInfo}

    results := jobCollec.FindOneAndUpdate(ctx, filter, update,
options.FindOneAndUpdate().SetReturnDocument(1))

    var jobListing model.JobListing

    if err := results.Decode(&jobListing); err != nil {
        log.Fatal(err)
    }

    return &jobListing
}
func (db *DB) DeleteJobListing(jobId string) *model.DeleteJobResponse {
    jobCollec := db.client.Database("graphql-job-board").Collection("jobs")
    ctx, cancel := context.WithTimeout(context.Background(),
30*time.Second)
    defer cancel()

    _id, _ := primitive.ObjectIDFromHex(jobId)

```

```

filter := bson.M{"_id": _id}
_, err := jobCollec.DeleteOne(ctx, filter)
if err != nil {
    log.Fatal(err)
}
return &model.DeleteJobResponse{DeletedJobID: jobId}
}

```

15) schema.graphqls didalam file ini masukan

```
# GraphQL schema example
```

```
#
```

```
# https://gqlgen.com/getting-started/
```

```
type JobListing {
```

```
  _id: ID!
```

```
  title: String!
```

```
  description: String!
```

```
  company: String!
```

```
  url: String!
```

```
}
```

```
type Query {
```

```
  jobs: [JobListing!]!
```

```
  job(id: ID!): JobListing!
```

```
}
```

```
type Mutation {
```

```
  createJobListing(input: CreateJobListingInput!): JobListing!
```

```
  updateJobListing(id: ID!, input: UpdateJobListingInput!): JobListing!
```

```
  deleteJobListing(id: ID!): DeleteJobResponse!
```

```
}
```

```
input CreateJobListingInput {
```

```
  title: String!
```

```
  description: String!
```

```
  company: String!
```

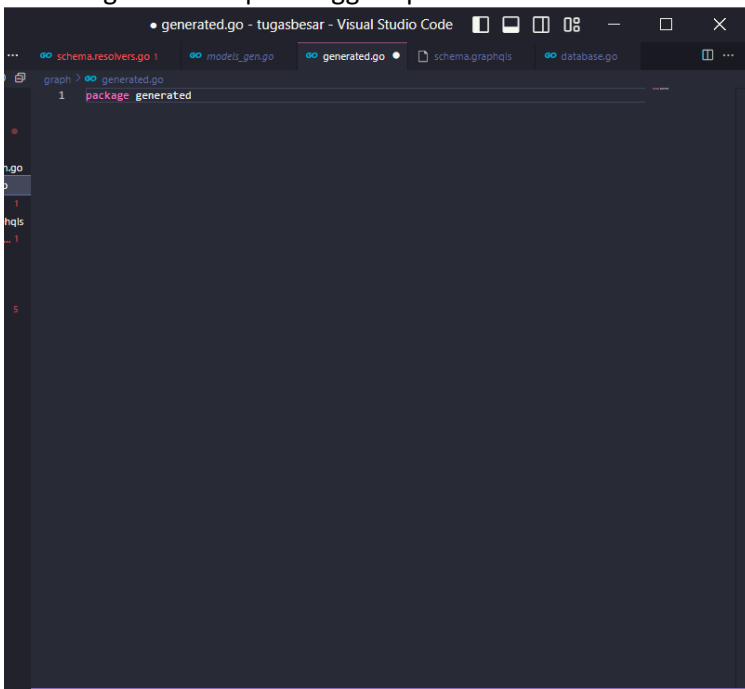
```
  url: String!
```

```
}
```

```
input UpdateJobListingInput {
  title: String
  description: String
  url: String
}
```

```
type DeleteJobResponse {
  deletedJobId: String!
}
```

16) Didalam file generate hapus hingga seperti ini



Gambar 12. 15 Hapus codingan pada file generate

17) Di dalam file schema.resolvers.go

```
package graph
```

```
// This file will be automatically regenerated based on the schema, any
// resolver implementations
```

```
// will be copied through when generating and any unknown code will be
// moved to the end.
// Code generated by github.com/99designs/gqlgen version v0.17.27

import (
    "context"

    "github.com/akhil/gql-yt/database"
    "github.com/akhil/gql-yt/graph/model"
)

// CreateJobListing is the resolver for the createJobListing field.
func (r *mutationResolver) CreateJobListing(ctx context.Context, input
model.CreateJobListingInput) (*model.JobListing, error) {
    return db.CreateJobListing(input), nil
}

// UpdateJobListing is the resolver for the updateJobListing field.
func (r *mutationResolver) UpdateJobListing(ctx context.Context, id
string, input model.UpdateJobListingInput) (*model.JobListing, error) {
    return db.UpdateJobListing(id, input), nil
}

// DeleteJobListing is the resolver for the deleteJobListing field.
func (r *mutationResolver) DeleteJobListing(ctx context.Context, id string)
(*model.DeleteJobResponse, error) {
    return db.DeleteJobListing(id), nil
}

// Jobs is the resolver for the jobs field.
func (r *queryResolver) Jobs(ctx context.Context) ([]*model.JobListing,
error) {
    return db.GetJobs(), nil
}

// Job is the resolver for the job field.
func (r *queryResolver) Job(ctx context.Context, id string)
(*model.JobListing, error) {
    return db.GetJob(id), nil
}
```

```
// Mutation returns MutationResolver implementation.
func (r *Resolver) Mutation() MutationResolver { return
&mutationResolver{r} }

// Query returns QueryResolver implementation.
func (r *Resolver) Query() QueryResolver { return &queryResolver{r} }

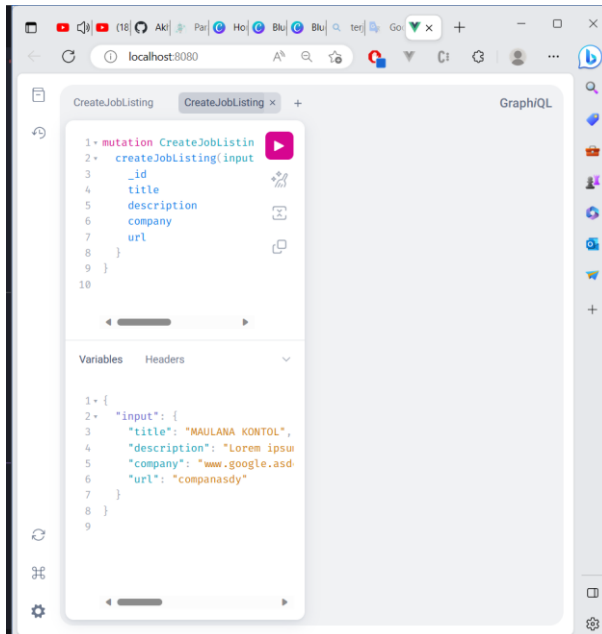
type mutationResolver struct{ *Resolver }
type queryResolver struct{ *Resolver }

// !!! WARNING !!!
// The code below was going to be deleted when updating resolvers. It has
// been copied here so you have
// one last chance to move it out of harms way if you want. There are two
// reasons this happens:
// - When renaming or deleting a resolver the old code will be put in
// here. You can safely delete
// it when you're done.
// - You have helper methods in this file. Move them out to keep these
// resolver files clean.
var db = database.Connect()
```

18) Setelah itu go mod tidy

19) Lalu go run server.go

Hingga muncul seperti ini



Gambar 12. 16 Hasil Tampilan Pada Halaman Local Host

PERJALANAN GOLANG DENGAN GRAPHQL PASTI PENUH DENGAN WARNA JIKA HARUS DICERITAKAN, SETIAP ORANG PUNYA CERITA DAN KISAHNYA MASING-MASING. PASTINYA SEMUAH JALAN YANG DILALUI MENGINGINKANKAN HASIL YANG TERBAIK TERSEBUTLAH KISAH SI GOLANG, SEBUAH PLATFROM PEMBUAT APLIKASI BACKEND KISAH GOLANG SI MEMBUAT LOGIC INI PENUH PENUH DENGAN PERGOLAKAN BATIK YANG MENGHARUKAN, SEMANGAT KALIAN YANG SEDANG MENCOBA METAVERSE SAYANG BAGI KALIAN YANG PUNYA PONTENSI, SEBUAH BUKU YANG PATUT DI BACA, BAIK OLEH ANAK-ANAK DAN DEWASA.

