

**KIAT MAHIR AI DENGAN GOLANG**

Pengenalan Mesin Learning Menggunakan Golang

**Ardini Yuanita Lubis**

**Bernadus Billy Riantono**

**Yadi Mulyadi**

|  |  |
| --- | --- |
|  | **PT. Penerbit Buku Pedia**  **2023** |

# KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh,

Puji syukur kami panjatkan ke hadirat Allah SWT, karena atas limpahan rahmat dan karunia-Nya, kami dapat menyelesaikan penulisan buku ini. Buku ini merupakan hasil kerja keras dan dedikasi dari berbagai pihak yang telah berkontribusi secara langsung maupun tidak langsung.

Kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada para pembimbing dan dosen pembimbing kami yang telah memberikan arahan dan panduan selama proses penulisan. Tak lupa, ucapan terima kasih kami sampaikan kepada keluarga dan teman-teman yang selalu memberikan semangat dan dukungan moral dalam setiap langkah kami. Kata-kata penuh doa dan dorongan dari mereka telah menjadi sumber inspirasi dan motivasi bagi kami.

Buku ini tidak lepas dari hasil kajian dan referensi yang kami peroleh dari berbagai sumber. Oleh karena itu, kami mengucapkan terima kasih kepada para penulis, akademisi, dan institusi yang telah menyediakan literatur dan data yang berharga.

Semoga buku ini dapat memberikan manfaat dan kontribusi bagi pembaca yang ingin memperdalam pengetahuan tentang topik yang dibahas. Kami menyadari bahwa buku ini tidak sempurna, oleh karena itu kritik dan saran yang membangun sangat kami harapkan untuk perbaikan di masa mendatang.

Akhir kata, semoga segala upaya yang kami lakukan dalam penulisan buku ini bisa menjadi amal jariyah yang bermanfaat bagi kami dan semua pihak yang terlibat. Semoga Allah SWT senantiasa memberikan hidayah dan rahmat-Nya kepada kita semua.

Wassalamu'alaikum warahmatullahi wabarakatuh.

[Penulis]

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc141777360)

[DAFTAR ISI ii](#_Toc141777361)

[BAB 1 PENGENALAN PEMROGRAMAN GOLANG 1](#_Toc141777362)

[A. SEJARAH SINGKAT GOLANG 1](#_Toc141777363)

[B. FUNGSI, KEUNGGULAN DAN FITUR GOLANG 2](#_Toc141777364)

[C. INSTALLASI GOLANG 5](#_Toc141777365)

[D. GO COMMAND 8](#_Toc141777366)

[E. PROGRAM HELLO WORLD 9](#_Toc141777367)

[BAB 2 TIPE DATA, VARIABEL DAN KONSTANTA 12](#_Toc141777368)

[A. TIPE DATA DASAR 12](#_Toc141777369)

[B. DEKLARASI VARIABEL 16](#_Toc141777370)

[C. KONSTANTA 17](#_Toc141777371)

[D. TIPE DATA CUSTOM 19](#_Toc141777372)

[E. ZERO VALUE 22](#_Toc141777373)

[BAB 3 OPERASI DASAR DATA 24](#_Toc141777374)

[A. OPERASI ARITMATIKA 24](#_Toc141777375)

[B. OPERASI LOGIKA 28](#_Toc141777376)

[C. OPERASI PERBANDINGAN 30](#_Toc141777377)

[D. OPERASI STRING 35](#_Toc141777378)

[E. OPERASI ARRAY 40](#_Toc141777379)

[BAB 4 PENGOLAHAN DATA PADA GOLANG 44](#_Toc141777380)

[A. STRING MANIPULASI 45](#_Toc141777381)

[B. OPERASI FILE 46](#_Toc141777382)

[C. MANIPULASI STRUKTUR DATA 48](#_Toc141777383)

[D. PENCARIAN DAN PENGURUTAN 51](#_Toc141777384)

[E. OPERASI MATEMATIKA DAN STATISTIK 56](#_Toc141777385)

[F. PEMROSESAN PARALEL 60](#_Toc141777386)

[G. PEMROSESAN JSON/XML 61](#_Toc141777387)

[H. OPERASI KONVERSI TIPE DATA 64](#_Toc141777388)

[BAB 5 MACHINE LEARNING PADA BAHASA PEMROGRAMAN GO 71](#_Toc141777389)

[A. KONSEP MACHINE LEARNING 71](#_Toc141777390)

[B. PERAN SERTA KELEBIHAN DAN KEKURANGAN BAHASA GO UNTUK MACHINE LEARNING 73](#_Toc141777391)

[C. LIBRARY BAHASA GO UNTUK MEMBANGUN MESIN LEARNING 76](#_Toc141777392)

[D. TEKNIK PREPROCESSING DATA PADA BAHASA GO 81](#_Toc141777393)

[E. ALGORITMA MACHINE LEARNING PADA BAHASA GO 83](#_Toc141777394)

[F. KIAT MEMBANGUN MACHINE LEARNING MENGGUNAKAN GOLANG 84](#_Toc141777395)

[DAFTAR PUSTAKA 86](#_Toc141777396)

# BAB 1 PENGENALAN PEMROGRAMAN GOLANG

## SEJARAH SINGKAT GOLANG

Golang, yang merupakan singkatan dari Go Language, adalah sebuah bahasa pemrograman yang dikembangkan oleh Google dan dikelola oleh perusahaan tersebut. Menurut Computer Hope, Golang mulai dikembangkan pada tahun 2007 oleh tiga pengembang di Google, yaitu Ken Thompson, Robert Griesemer, dan Rob Pike. Pada tahun 2009, Golang secara resmi dirilis ke publik.

Bahasa pemrograman ini memungkinkan penulisan kode secara statis dan menghasilkan kode biner yang dikompilasikan untuk mesin. Golang, sebuah bahasa pemrograman yang muncul pada abad ke-21, berupaya menggabungkan fitur-fitur dari bahasa pemrograman sebelumnya. Bahasa pemrograman ini dapat digunakan untuk membuat website, aplikasi, dan perangkat lunak.

Tujuan utama pengembangan Golang adalah menciptakan sebuah bahasa pemrograman yang efisien, mudah dipelajari, dan cocok untuk pengembangan perangkat lunak skala besar. Golang didesain dengan maksud mengatasi tantangan yang dihadapi oleh bahasa pemrograman lain, seperti waktu kompilasi yang lambat, kompleksitas sintaksis yang tinggi, dan kesulitan dalam memanfaatkan keunggulan sistem multiprosesor modern.

Dalam proses pengembangannya, Golang menganut beberapa prinsip desain yang krusial, termasuk kesederhanaan, kejelasan, efisiensi, keandalan, dan dukungan terhadap pemrograman konkuren. Golang juga mendorong pendekatan praktis yang membantu pengembang dalam membangun perangkat lunak yang dapat diskalakan dan mudah dikelola.

Golang dirilis dengan lisensi open source, yang mengizinkan komunitas pengembang untuk berpartisipasi, memberikan masukan, dan mengembangkan ekosistem bahasa tersebut. Sejak dirilis, Golang telah meraih popularitas yang signifikan dan digunakan secara luas di berbagai bidang, termasuk pengembangan perangkat lunak web, sistem backend, pemrograman jaringan, dan banyak lagi.

## FUNGSI, KEUNGGULAN DAN FITUR GOLANG

Saat ini, mempelajari Golang menjadi sangat penting karena bahasa ini memberikan bantuan yang signifikan dalam pengembangan perangkat lunak secara efisien dan memiliki tingkat pembelajaran yang mudah. Salah satu nilai tambah terbesar dari Golang adalah keamanan dan skalabilitas yang dimilikinya. Selain itu, bahasa Go juga cocok digunakan dalam pembuatan situs e-commerce.

Sebagai seorang back end programmer dalam sebuah perusahaan, Anda pasti menghadapi tantangan seperti peningkatan aktivitas pengguna dalam aplikasi. Untuk mengatasi masalah tersebut, Anda dapat mengandalkan fitur Golang yang disebut concurrency (pemrograman konkuren) dan footprint yang memiliki kapasitas memori yang rendah. Fitur-fitur ini membantu dalam mengelola dan menangani aktivitas pengguna yang meningkat secara efisien.

Berikut ini adalah beberapa fungsi yang dimiliki oleh Golang dan dapat diterapkan dalam bisnis atau pekerjaan :

* Dapat membangun tim developer yang lebih scalable, baik berupa startup maupun perusahaan korporasi dalam bidang teknologi informasi.
* Membangun sebuah sistem yang lebih kompleks serta membutuhkan kinerja yang tinggi.
* Mengembangkan teknologi cloud computing yang memiliki skalabilitas yang mumpuni.
* Dapat mengembangkan kode server jaringan, terutama pada layanan mikro dan web server.
* Mampu untuk merancang aplikasi berbasis web yang lebih aman.

Walaupun Golang adalah bahasa pemrograman open-source yang relatif baru, Google merancangnya dengan tujuan memenuhi kebutuhan pengembang yang bekerja pada proyek skala besar. Terdapat beberapa keunggulan Golang yang diungkapkan oleh Scale Focus, berikut adalah beberapa di antaranya adalah sebagai berikut :

1. Kode yang Sederhana

Struktur data dan algoritma dalam Golang menggunakan kode yang sederhana dan mudah dipelajari. Hal ini memberikan manfaat besar bagi para pengembang, karena dapat menghemat waktu yang dibutuhkan dalam mengimplementasikan dan memahami struktur data dan algoritma tersebut.

1. Memiliki Performa yang Kuat

Kesederhanaan Go secara efektif meningkatkan kinerjanya menjadi lebih kuat. Golang memungkinkan siklus pengembangan yang lebih singkat dan efisien. Selain itu, tanpa keberadaan mesin virtual, Golang mampu meningkatkan kecepatan secara signifikan. Beberapa keunggulan Golang meliputi kemampuan untuk bekerja dengan cepat, kompilasi kode yang lebih cepat, dan kemudahan dalam pemeliharaan.

1. Cocok untuk Proyek Skala Besar

Efektivitas menjadi poin utama, sehingga banyak perusahaan yang memilih Golang. Selain karena kecepatannya, Golang memungkinkan untuk digunakan pada proyek skala besar dan lebih menantang.

1. Dirancang untuk Multi-Core Processors

Golang didesain untuk era cloud computing (komputasi awan) modern dan proses paralel yang digunakan oleh prosesor saat ini. Bahasa pemrograman komputer yang sudah lebih dulu populer, seperti Java, JavaScript, Ruby, Python, C, dan C++ dibuat sebelum komputer multi-core digunakan secara luas. Nah, dengan Golang, akan lebih mudah untuk menggunakan semua CPU core tanpa mempersulit proses pengembangan.

1. Golang memiliki Garbage Collection yang Cepat

Go dilengkapi dengan fitur garbage collection, yang dapat meningkatkan performa aplikasi, karena memori sistem tidak dijejali dengan 'sampah' yang dapat memberatkan aplikasi.

1. Dirancang untuk Internet

Dengan library standar Golang, pengembang bisa membangun layanan web yang kompleks tanpa library pihak ketiga. Hal ini membuat Golang menjadi bahasa pemrograman yang sempurna untuk pengembangan website.

1. Struktur Data dan Algoritmanya Mudah untuk Dipelajari

Dengan library standar Golang, pengembang bisa membangun layanan web yang kompleks tanpa library pihak ketiga. Hal ini membuat Golang menjadi bahasa pemrograman yang sempurna untuk pengembangan website.

1. Maintenance Mudah

Dengan library standar Golang, pengembang bisa membangun layanan web yang kompleks tanpa library pihak ketiga. Hal ini membuat Golang menjadi bahasa pemrograman yang sempurna untuk pengembangan website.

1. Bersifat Open-Source

Platform Golang bersifat open-source dan membuatnya memiliki banyak penggemar di kalangan programmer. Komunitas pengguna Golang juga terbilang sangat masif. Golang dapat dipelajari melalui berbagai forum, tutorial, dan proyek open-source.

1. Terkenal di Dunia Industri

Dengan mengadopsi bahasa pemrograman Golang dapat menghasilkan coding yang lebih cepat dan menjadikan programmer lebih produktif dalam bekerja. Inilah mengapa, Golang populer di berbagai industri.

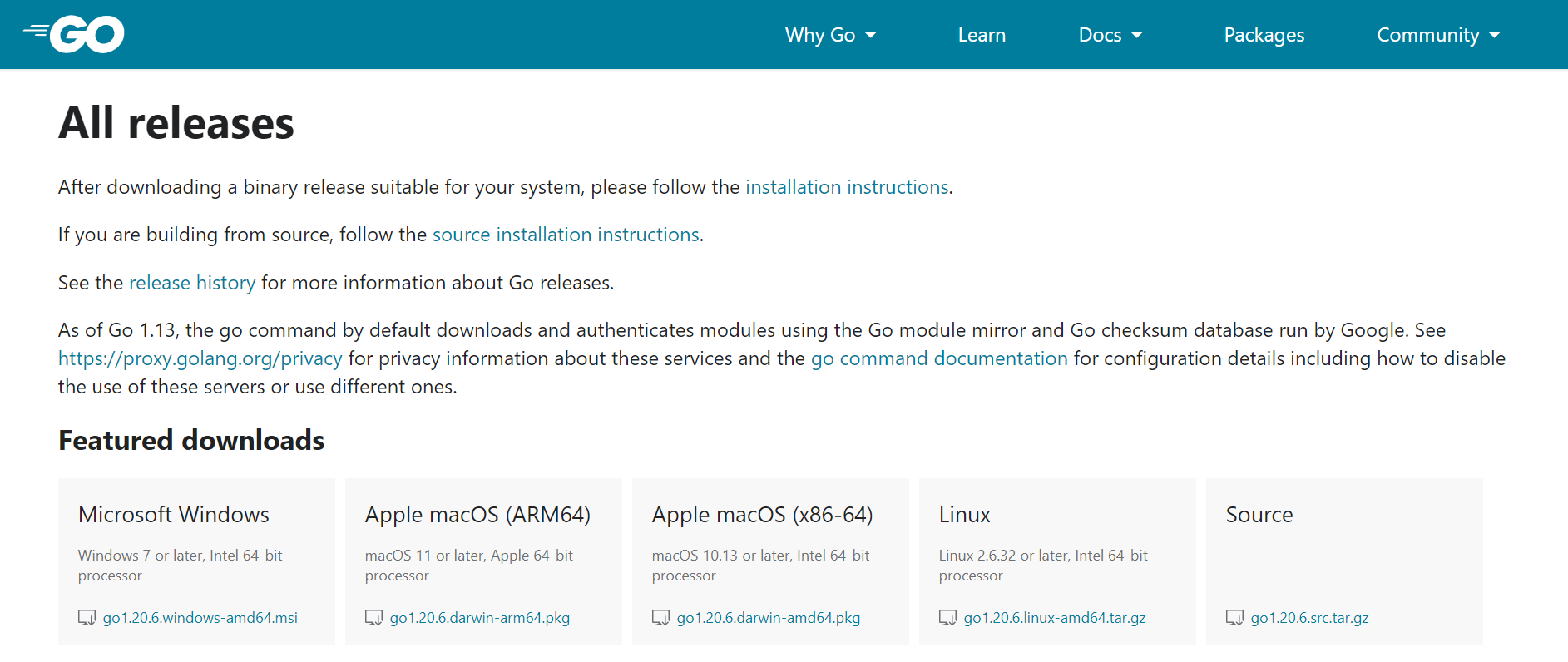
Golang memiliki beberapa fitur utama yang membuatnya populer di kalangan pengembang. Berikut adalah beberapa fitur penting dari Golang :

1. Sintaks yang Sederhana dan Mudah Dibaca: Go dirancang dengan sintaks yang sederhana dan mudah dipahami. Hal ini memungkinkan pengembang untuk menulis kode yang ringkas dan mudah dibaca, sehingga mempermudah kolaborasi dan pemeliharaan kode.
2. Penanganan Garbage Collection (Pengelolaan Memori Otomatis): Go memiliki garbage collector yang membebaskan pengembang dari beban manual pengelolaan memori. Garbage collector secara otomatis menghapus objek yang tidak lagi digunakan, sehingga membantu mencegah kebocoran memori dan memudahkan pengembangan aplikasi.
3. Konkurensi (Goroutines): Go memiliki konsep konkurensi yang memungkinkan eksekusi tugas secara bersamaan (concurrent) dengan goroutines. Goroutines adalah unit eksekusi yang ringan, mirip dengan thread, tetapi menggunakan sedikit sumber daya sistem. Goroutines memungkinkan pengembang untuk menulis kode konkuren secara lebih mudah dan efisien.
4. Sistem Pengemasan (Package System): Go menggunakan sistem pengemasan yang kuat untuk mengatur dan mengelompokkan kode ke dalam paket-paket yang terpisah. Paket-paket ini memfasilitasi pemisahan tugas, pengelolaan dependensi, dan penggunaan kembali kode.
5. Pengelolaan Kesalahan (Error Handling): Go memiliki pendekatan yang konsisten untuk pengelolaan kesalahan. Go menggunakan nilai khusus bernama "error" untuk mengindikasikan kegagalan dalam suatu operasi. Pendekatan ini memudahkan pengembang untuk mengelola dan menangani kesalahan secara jelas dan konsisten.
6. Kompilasi Cepat: Go memiliki kompilator yang cepat, yang memungkinkan waktu kompilasi yang singkat. Hal ini memungkinkan pengembang untuk mengembangkan dan menguji kode dengan cepat, meningkatkan produktivitas dalam siklus pengembangan perangkat lunak.
7. Performa dan Efisiensi: Go dirancang dengan fokus pada performa dan efisiensi. Bahasa ini dirancang untuk menghasilkan kode yang cepat, baik dalam eksekusi maupun penggunaan memori.
8. Penanganan Kesalahan yang Ekspresif: Go memiliki pendekatan yang jelas dan ekspresif dalam menangani kesalahan. Fitur seperti pengecekan kesalahan yang tegas (explicit error checking) dan penggunaan fungsi multiple return value memudahkan pengembang untuk mengelola dan menangani kesalahan dengan baik.
9. Pembuatan Perangkat Lunak Berskala Besar: Go didesain untuk mengatasi masalah dalam pengembangan perangkat lunak berskala besar. Dukungan bawaan untuk konkurensi, pemrograman jaringan, pengujian unit, dan pemeliharaan kode memudahkan pengembang untuk mengembangkan dan menjaga aplikasi yang kompleks.
10. Kaya dengan Pustaka Standar: Go dilengkapi dengan pustaka standar yang kaya, yang menyediakan fungsi-fungsi umum yang sering digunakan seperti manipulasi string, pemrosesan file, pengkodean JSON, dan banyak lagi. Pustaka standar ini membantu pengembang mengembangkan aplikasi dengan lebih cepat dan efisien.

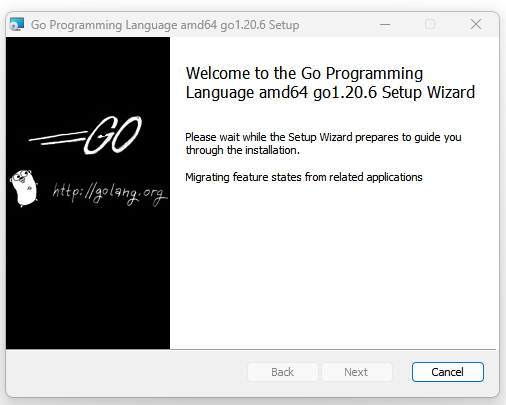
## INSTALLASI GOLANG

Sebelum kita memulai untuk membuat sebuah program dengan menggunakan Bahasa Golang, terlebih dahulu kita harus menginstall Bahasa Golang tersebut. Berikut ini adalah Langkah-langkah untuk melakukan installasi Go :

1. Download paket instalasi Go pada situs resminya di <https://golang.org/dl/> sesuai dengan system operasi yang digunakan. Pilih versi yang sesuai dengan arsitektur dan system operasi (misalnya Windows, MacOS atau Linux)

****

1. Setelah unduhan selesai, buka paket instalasi dan ikuti petunjuk untuk melakukan instalasi. Proses instalasi akan bervariasi tergantung pada system operasi yang digunakan.

****

1. Pada saat instalasi aka nada pilihan direktori sebagai posisi installasi Go. Pilih direktori yang sesuai atau menggunakan konfigurasi default.
2. Setelah proses instalasi selesai, lakukan pengecekan apakah instalasi berhasil atau tidak dengan membuka command prompt atau dapat menggunakan shortcut pada keyboard Win + R, ketik cmd, kemudian enter. Lalu ketikkan perintah go atau go version, lalu tekan enter seperti pada gambar berikut :



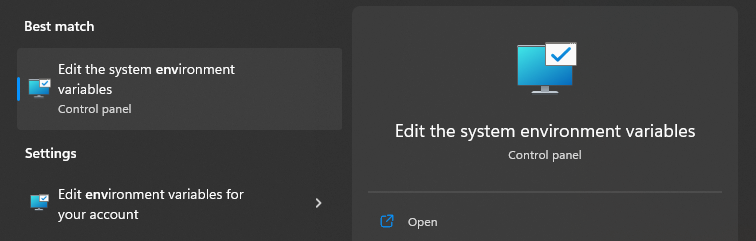
Instalasi berhasil apabila sudah menampilkan respon versi Go yang ter-install seperti gambar di bawah ini :



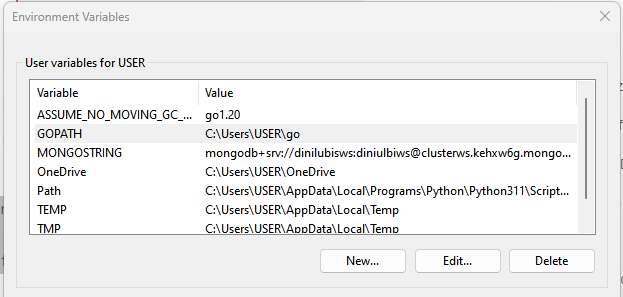
1. Langkah berikutnya adalah mengatur environment variable agar system dapat mengenali perintah Go. Caranya tergantung pada system operasi yang digunakan :

* **Windows**

Buka **Start** dan langsung ketikkan **env**, secara otomatis sistem akan memunculkan **Edit the System Environment Variables**, dan pada tab **Advanced**, klik **Environment Variables**. Cari variabel **GOPATH** dan edit sesuai dengan direktori yang diinginkan.



Apabila sudah terupdate, cek kembali status variabel **GOPATH** melalui command prompt dengan perintah **go env**. Jika sudah berubah, maka sudah bisa mulai menggunakan Go dan menyimpan file go pada direktori yang sudah di-set.



* **MacOS**

Buka terminal dan buka file “/.bash\_profile” kemudian tambahkan baris berikut ke dalam file :



Lalu simpan dan jalankan perintah berikut :



* **LINUX**

Buka terminal dan buka file “/.bashrc” kemudian tambahkan baris berikut ke dalam file :

****

Lalu simpan dan jalankan perintah berikut :

****

## GO COMMAND

Dalam bahasa pemrograman Go, tersedia perintah **go** yang digunakan untuk berbagai keperluan dalam pengembangan aplikasi. Perintah ini dieksekusi melalui command line interface (CLI), seperti kompilasi, pengujian, dan eksekusi program. Tabel berikut menyajikan daftar perintah **go** beserta penjelasan penggunaannya.

Tabel 1.1 Perintah bahasa Go

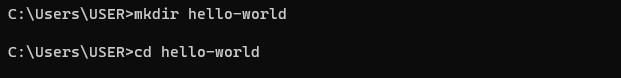
|  |  |
| --- | --- |
| Perintah | Deskripsi |
| go mod init | Digunakan untuk inisialisasi project pada Go. |
| go run | Digunakan untuk eksekusi file program |
| go test | Digunakan untuk keperluan unit test |
| go build | Digunakan untuk mengkompilasi file program |
| go get | Digunakan untuk mend-download package yang akan digunakan pada project |
| go mod tidy | Digunakan untuk memvalidasi dependensi. Jika ada dependensi yang belum ter-download makan akan di download secara otomatis |
| go mod vendor | Digunakan untuk vendoring |

## PROGRAM HELLO WORLD

Setelah semua persiapan telah selesai, sekarang saatnya memulai sesi pemrograman. Kita akan membuat program pertama, yaitu sebuah aplikasi kecil yang menampilkan teks "Hello, world".

* Inisialisasi Project

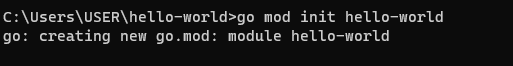
Buatlah sebuah direktori dengan nama "hello-world" di lokasi yang Anda inginkan. Selanjutnya, masuk ke direktori tersebut dan jalankan perintah untuk menginisialisasi proyek.



Lalu ketikkan perintah berikut untuk menginisialisasi project :

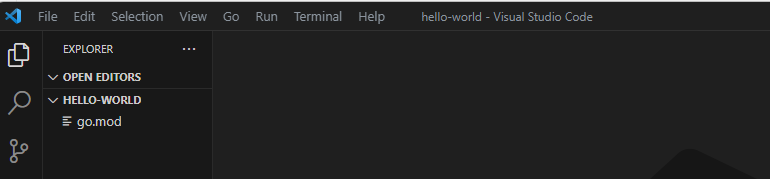


Maka pada layer akan menampilkan hasil seperti pada gambar berikut ini :



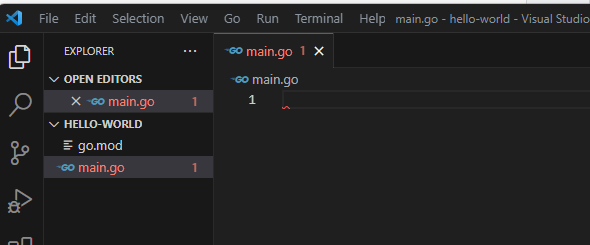
* Load Project Folder ke Editor

Mulailah dengan membuka editor, dalam hal ini penulis menggunakan Visual Studi Code sebagai editornya. Temukan opsi menu untuk menambahkan proyek dan pilih folder proyek "hello-world". Beberapa editor juga mendukung cara memuat proyek dengan cukup mengklik dan menarik (klik-drag) folder proyek ke dalam editor.

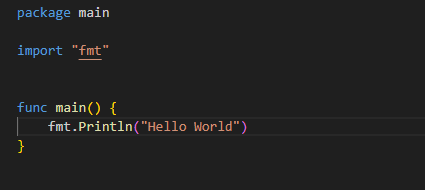


* Menyiapkan File Program

Pada project yang telah dibuat, siapkan sebuah file berektensi .go dengan nama bebas. File Program berektensi .go ini akan berisi source code Go.



Lalu ketikkan kode program sederhana berikut untuk menampilkan text **Hello World**  ke layer output.

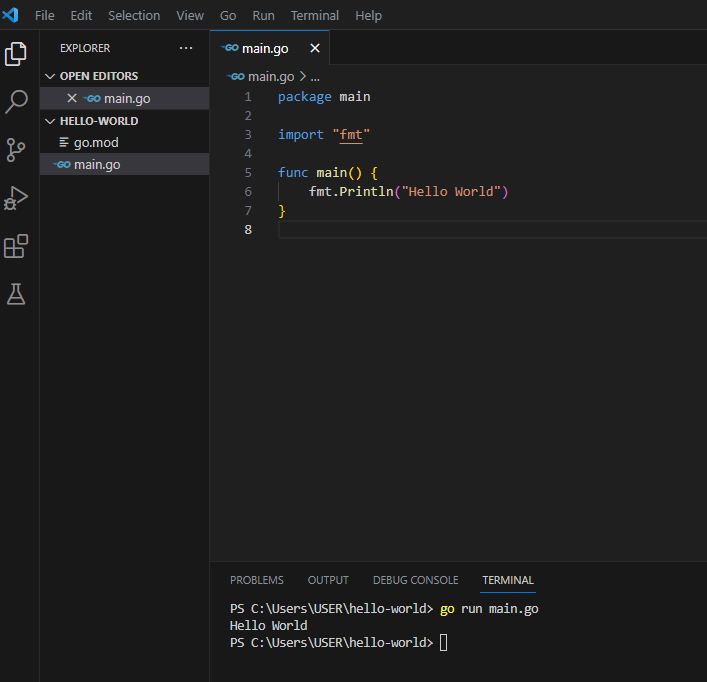


* Running Program Pertama: Hello Word

Setelah membuat source code, buka terminal, lalu masuk ke direktori project kemudian jalankan program dengan menggunakan perintah berikut :



Dan akan menampilkan hasil seperti pada gambar di bawah ini :



Program sederhana pertama dengan menggunakan Go telah berhasil dibuat dan dijalankan.

# BAB 2 TIPE DATA, VARIABEL DAN KONSTANTA

## TIPE DATA DASAR

Bahasa pemrograman memerlukan pemrosesan dan organize data. Data tersebut disimpan sebagai bilangan biner ( bilangan 0 dan 1) dalam memori komputer. Dalam Bahasa pemgorgraman Go (Golang) memiliki berbagai tipe data dasar yang digunakan untuk mendefinisikan variabel dan nilai dalam program. Berikut ini adalah tipe data dasar dalam Bahasa Pemrograman Go :

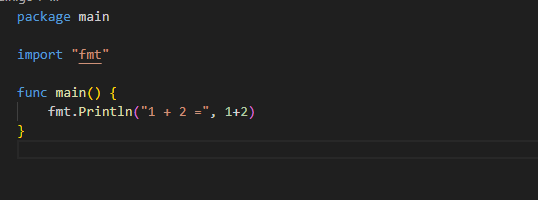
1. Integers

Adalah tipe bilangan bulat dalam bahasa Go (Golang) terdiri dari beberapa tipe data, yaitu:

* **Int** : Tipe data int digunakan untuk menyimpan bilangan bulat dengan ukuran yang bervariasi tergantung pada platform sistem. Ukuran int umumnya adalah 32-bit pada sistem 32-bit dan 64-bit pada sistem 64-bit.
* **int8** : Tipe data int8 menyimpan bilangan bulat dengan ukuran 8-bit (signed), sehingga rentang nilainya adalah -128 hingga 127.
* int16: Tipe data int16 menyimpan bilangan bulat dengan ukuran 16-bit (signed), sehingga rentang nilainya adalah -32768 hingga 32767.
* **int32** atau **run e**: Tipe data int32 (alias rune) menyimpan bilangan bulat dengan ukuran 32-bit (signed), dan sering digunakan untuk menyimpan representasi kode Unicode.
* **int64** : Tipe data int64 menyimpan bilangan bulat dengan ukuran 64-bit (signed), sehingga rentang nilainya lebih luas dari tipe data int lainnya, yaitu -9223372036854775808 hingga 9223372036854775807.
* **Uint** : Tipe data uint digunakan untuk menyimpan bilangan bulat non-negatif (unsigned) dengan ukuran yang bervariasi tergantung pada platform sistem.
* **uint8** atau **byte** : Tipe data uint8 (alias byte) menyimpan bilangan bulat non-negatif dengan ukuran 8-bit, sehingga rentang nilainya adalah 0 hingga 255.
* **uint16** : Tipe data uint16 menyimpan bilangan bulat non-negatif dengan ukuran 16-bit, sehingga rentang nilainya adalah 0 hingga 65535.
* **uint32** : Tipe data uint32 menyimpan bilangan bulat non-negatif dengan ukuran 32-bit, sehingga rentang nilainya adalah 0 hingga 4294967295.
* **uint64** : Tipe data uint64 menyimpan bilangan bulat non-negatif dengan ukuran 64-bit, sehingga rentang nilainya adalah 0 hingga 18446744073709551615.

|  |  |  |  |
| --- | --- | --- | --- |
| Data Type | Memory Usage – Number of Bits | Minimum Value | Maximum Value |
| int8 | 8 | -128 | 127 |
| int16 | 16 | -32768 | 32767 |
| int32 | 32 | -2147483648 | 2147483647 |
| int64 | 64 | -9223372036854775808 | 9223372036854775807 |
| uint8 | 8 | 0 | 255 |
| uint16 | 16 | 0 | 65535 |
| uint32 | 32 | 0 | 4294967295 |
| uint64 | 64 | 0 | 18446744073709551615 |

Berikut ini contoh penggunaan tipe data integer pada program go :



Jika program tersebut dijalankan, maka outputnya adalah sebagai berikut :

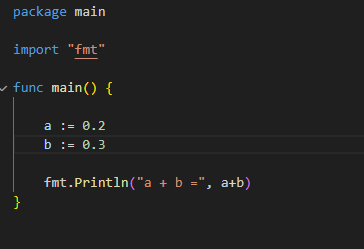


1. Floating

Tipe data floating adalah jenis bilangan yang digunakan untuk merepresentasikan bilangan real atau bilangan yang mengandung desimal, seperti 0.25, 110.333, -7.18, dan sebagainya. Berikut adalah tipe data float yang tersedia dalam Go :

* float32: Tipe data float32 digunakan untuk menyimpan bilangan desimal dengan presisi 32-bit. Rentang nilainya sekitar ±1.18E-38 hingga ±3.4E38, dengan ketelitian sekitar 7 digit decimal.
* float64: Tipe data float64 digunakan untuk menyimpan bilangan desimal dengan presisi 64-bit. Rentang nilainya sekitar ±2.23E-308 hingga ±1.8E308, dengan ketelitian sekitar 15 digit desimal.

Berikut ini contoh penggunaan tipe data floating pada program go :

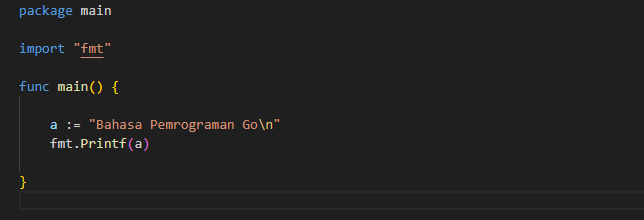


Jika program tersebut dijalankan, maka outputnya adalah sebagai berikut :



1. String

Tipe data string dalam bahasa pemrograman Go (Golang) adalah tipe data yang digunakan untuk menyimpan dan memanipulasi teks atau urutan karakter. Berikut ini contoh penggunaan tipe data floating pada program go :

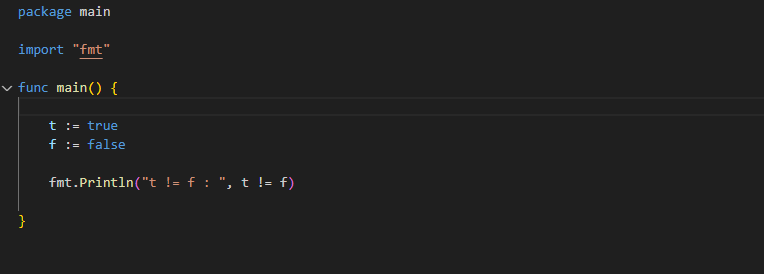


Jika program tersebut dijalankan, maka outputnya adalah sebagai berikut :



1. Boolean

Tipe data boolean dalam bahasa Go (Golang) adalah tipe data yang digunakan untuk menyimpan nilai kebenaran (true atau false). Berikut ini contoh penggunaan tipe data floating pada program go :



Jika program tersebut dijalankan, maka outputnya adalah sebagai berikut :



## DEKLARASI VARIABEL

Setelah memahami jenis-jenis tipe data pada Golang, selanjutnya kita akan melanjutkan pembahasan tentang deklarasi variable pada Golang dan bagaimana penggunaannya. Variabel adalah nama yang diberikan ke alamat memori untuk menyimpan nilai tipe tertentu. Variabel dalam golang di defenisikan dengan keyword **var.** Dalam Bahasa Go, setiap variabel yang dideklarasikan harus digunakan dalam kode program. Jika tidak digunakan maka akan terjadi error ketika program di compile. Berikut adalah beberapa contoh deklarasi variabel yang umum digunakan :

1. Deklarasi variabel dengan tipe data yang ditentukan



Contoh penggunaannya pada program Go :



1. Deklarasi variabel dengan inisialisasi nilai awal



Contoh penggunaannya pada program Go :



1. Deklarasi variabel dengan tipe data yang ditentukan secara otomatis (type inference)



Contoh penggunaannya pada program Go :



1. Deklarasi multi variabel



Contoh penggunaannya pada program Go :



## KONSTANTA

Konstanta dalam Golang adalah nilai yang tidak berubah dan konsisten dalam program. Konstanta dideklarasikan dengan nilai yang ditentukan saat deklarasi dan tidak dapat diubah selama program berjalan. Konstanta sering digunakan untuk menyimpan nilai-nilai yang tetap dan tidak berubah, seperti konstanta matematis, konfigurasi program, atau nilai-nilai yang telah ditentukan sebelumnya.

Dalam Go, deklarasi konstanta dilakukan menggunakan kata kunci **const** dan diikuti oleh nama konstanta dan nilai yang ditetapkan. Berikut adalah contoh deklarasi konstanta dalam Go :

****

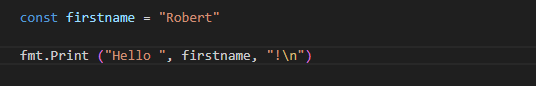
Contoh penggunaan konstanta dalam Go :

****

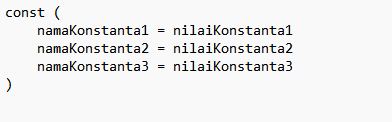
Dalam Go, deklarasi konstanta dilakukan menggunakan kata kunci **const** dan diikuti oleh nama konstanta dan nilai yang ditetapkan. Berikut adalah contoh deklarasi konstanta dalam Go :



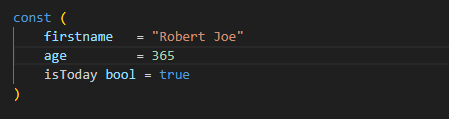
Deklarasi konstanta dengan type inference juga dapat diterapkan, yaitu dengan menghilangkan tipe data saat deklarasi.



Selain itu, konstanta juga dapat dideklarasikan secara bersamaan (multi Konstanta) dengan cara sebagai berikut :



Contoh penggunaan konstanta dalam Go :



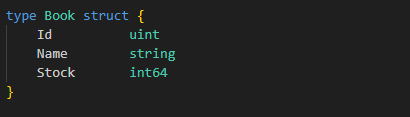
* firstname, dideklarasikan dengan metode type inference dengan tipe data string dan nilainya "Robert Joe"
* age, dideklarasikan dengan metode type inference dengan tipe data int dan nilainya 365
* isToday, dideklarasikan dengan metode manifest typing dengan tipe data bool dan nilainya true

## TIPE DATA CUSTOM

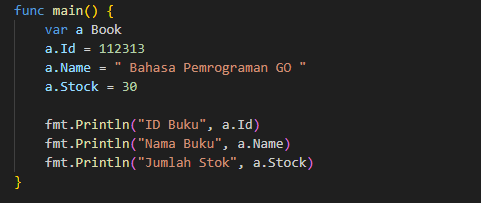
Dalam bahasa Go (Golang), kita dapat membuat tipe data custom (custom types) dengan menggunakan keyword **type**. Tipe data custom memungkinkan untuk membuat tipe data baru berdasarkan tipe data bawaan atau tipe data custom yang sudah ada. Ini dapat membantu dalam mengorganisir dan mengelompokkan data dengan lebih jelas sesuai dengan kebutuhan aplikasi yang akan dikembangkan. Berikut ini adalah contoh tipe data custom :

1. Struct

Stuct adalah singkatan dari structure yang merupakan tipe data terstruktur dalam Bahasa Go yang memungkinkan pengguna untuk menggabungkan beberapa tipe data yang berbeda dalam satu entitas. Pada struct memungkinkan untuk membuat tipe data baru dengan Menyusun fields-fields yang terkait. Berikut adalah contoh deklarasi struct :



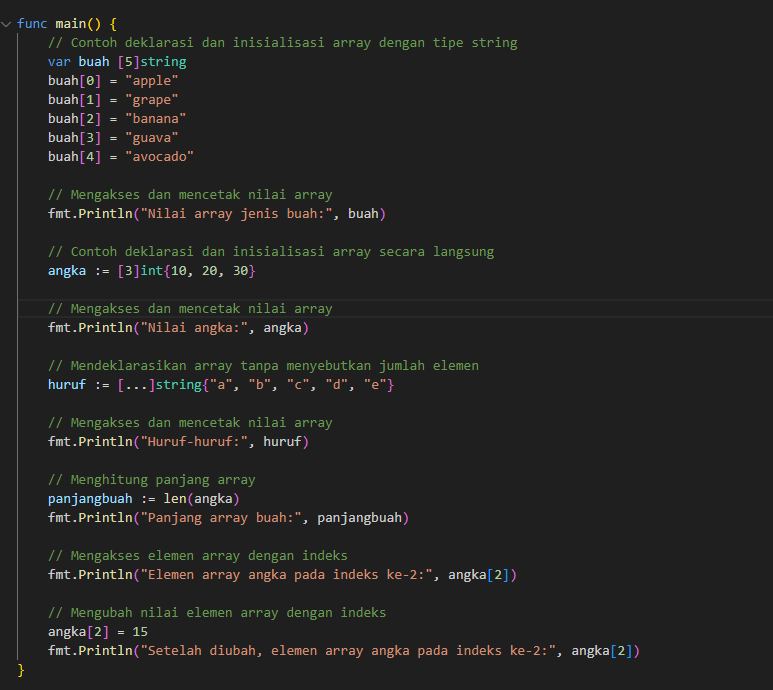
Contoh penggunan struct :



1. Tipe Data Custom pada Tipe Data Array



Pada contoh di atas, **ExArray** adalah tipe data custom yang diberikan pada tipe data array [4]int. Variabel dengan tipe data **ExArray** dapat dibuat dimana **ExArray** adalah tipe data array dengan panjang tetap sebesar 4. Berikut contoh penggunaan tipe data array :

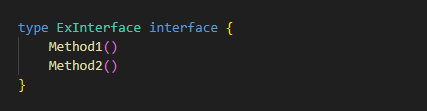


1. Tipe Data Custom pada TIpe Data Slice



Pada contoh id atas, **ExSlice** merupakan tipe data custom yang diberikan pada tipe data slice []string. MySlice dapat digunakan untuk membuat variabel dengan tipe slice yang berisi elemen-elemen string.

1. Tipe Data Custom pada Tipe Data Interface



Pada contoh di atas, **MyInterface** adalah tipe data custom yang diberikan pada tipe data interface yang mendefinisikan dua metode, yaitu Method1 dan Method2. Kita dapat menggunakan tipe data **MyInterface** untuk menyatakan bahwa suatu objek harus mengimplementasikan metode-metode tersebut.

1. Tipe Data Custom pada Tipe Data Function



Dalam contoh di atas, **MyFunction** adalah tipe data custom yang diberikan pada tipe data function yang mengambil argumen bertipe string dan mengembalikan nilai bertipe bool. Kita dapat menggunakan tipe data **MyFunction** untuk membuat variabel yang dapat menyimpan referensi ke fungsi yang memiliki tipe dan signature yang sama.

1. Tipe Data Custom pada Tipe Data Pointer



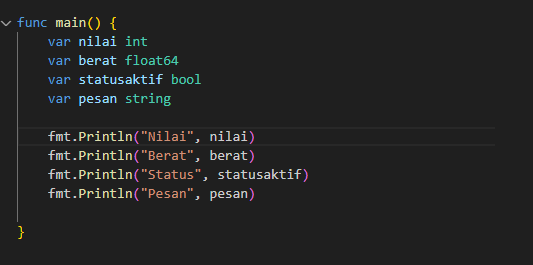
Pada contoh ini, **MyPointer** adalah tipe data custom yang diberikan pada tipe data pointer \*int. Anda dapat menggunakan tipe data **MyPointer** untuk membuat variabel yang dapat menyimpan alamat memori dari tipe data int.

## ZERO VALUE

Zero value dalam bahasa Go (Golang) merujuk pada nilai default yang diberikan kepada variabel saat mereka dideklarasikan tanpa inisialisasi eksplisit. Setiap tipe data dalam Go memiliki zero value yang sesuai. Berikut adalah beberapa contoh zero value untuk tipe data dasar dalam Go :

* Int : Zero value adalah 0.
* float32 dan float64 : Zero value adalah 0.0.
* bool : Zero value adalah false.
* String : Zero value adalah string kosong ("").
* Nil : Zero value untuk tipe data reference (pointer, slice, map, channel, interface) adalah nil. Ini menunjukkan bahwa variabel tersebut tidak menunjuk ke objek yang valid atau tidak memiliki nilai.

Contoh penggunaannya dalam pemrograman Go adalah sebagai berikut :

****

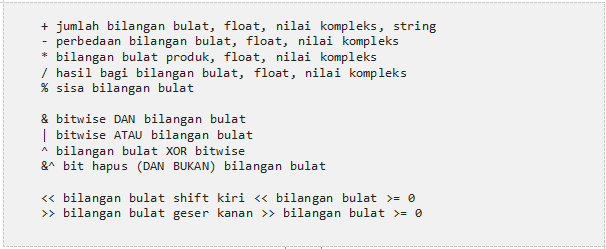
Dalam contoh di atas, variabel nilai akan memiliki zero value 0, variabel berat akan memiliki zero value 0.0, variabel aktif akan memiliki zero value false, dan variabel pesan akan memiliki zero value string kosong (""). Zero value dapat digunakan sebagai nilai awal yang aman saat kita tidak memiliki nilai yang spesifik untuk diberikan kepada variabel pada saat deklarasi. Namun, perlu diperhatikan bahwa zero value dapat berbeda dengan nol yang secara eksplisit diberikan kepada variabel setelah inisialisasi.

# BAB 3 OPERASI DASAR DATA

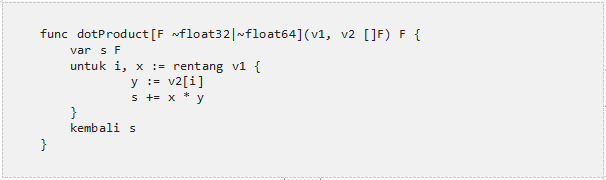
Operasi data pada bahasa Go (Golang) mencakup berbagai tugas dan operasi untuk memanipulasi, mengakses, memproses, dan menganalisis data.

## OPERASI ARITMATIKA

Operasi aritmatika digunakan pada nilai numerik dan menghasilkan hasil dengan tipe data yang sama seperti tipe data operand pertama. Ada empat operator aritmatika standar, yaitu penjumlahan (+), pengurangan (-), perkalian (\*), dan pembagian (/), yang berlaku untuk tipe data integer, floating-point, dan kompleks. Selain itu, operator penjumlahan (+) juga berlaku untuk tipe data string. Operasi logika bitwise dan operator shift hanya berlaku untuk tipe data bilangan bulat.



Jika tipe operand adalah sebuah parameter tipe, maka operator tersebut akan diterapkan pada setiap tipe dalam rangkaian tipe tersebut. Operand direpresentasikan sebagai nilai dari argumen tipe yang digunakan oleh parameter tipe, dan operasi dihitung dengan presisi dari argumen tipe tersebut. Sebagai contoh, jika diberikan sebuah fungsi seperti berikut ini :

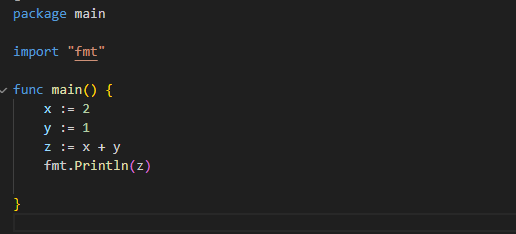


produk x \* y dan penjumlahan masing-masing s += x \* y dihitung dengan float32 atau float64 presisi, bergantung pada argumen tipe untuk F.

Contoh operator aritmatika :

1. Penjumlahan (+)

Operator penjumlahan digunakan untuk menjumlahkan dua nilai atau ekspresi. Berikut adalah contoh penggunaannya pada pemrograman Go

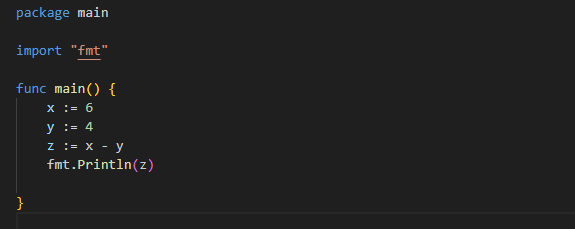


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Pengurangan (-)

Operator pengurangan digunakan untuk mengurangi nilai atau ekspresi kedua dari nilai atau ekspresi pertama. Berikut adalah contoh penggunaannya pada pemrograman Go :

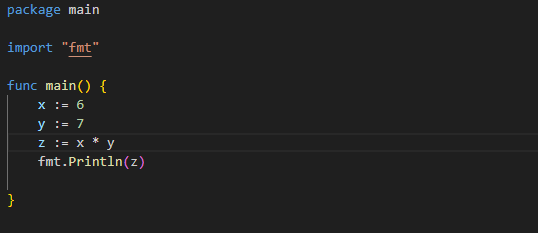


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Perkalian (\*)

Operator perkalian digunakan untuk mengalikan dua nilai atau ekspresi. Berikut adalah contoh penggunaannya pada pemrograman Go :

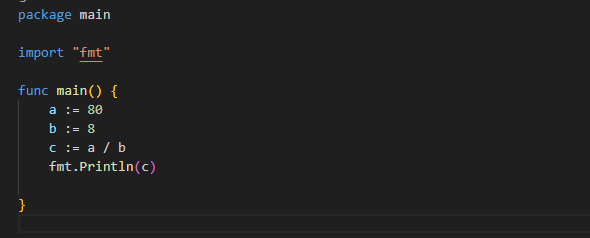


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Pembagian (/)

Operator pembagian digunakan untuk membagi nilai atau ekspresi pertama dengan nilai atau ekspresi kedua Berikut adalah contoh penggunaannya pada pemrograman Go :

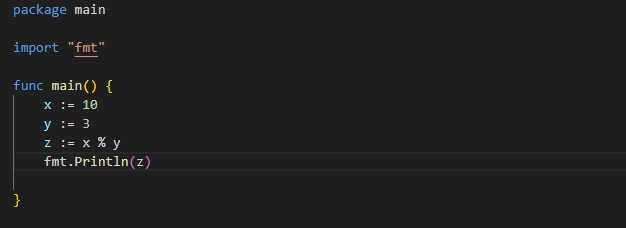


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Operator Bilangan Bulat (%)

Operator Bilangan Bulat mengembalikan sisa pembagian dari dua nilai atau ekspresi. Berikut adalah contoh penggunaannya pada pemrograman Go



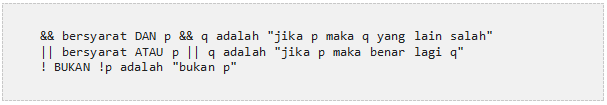
Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



Dalam contoh di atas, kita menghitung sisa hasil bagi antara bilangan bulat 10 dan 3 menggunakan operator %. Hasilnya adalah 1, karena jika kita membagi 10 dengan 3, hasilnya adalah 3 dengan sisa 1.

## OPERASI LOGIKA

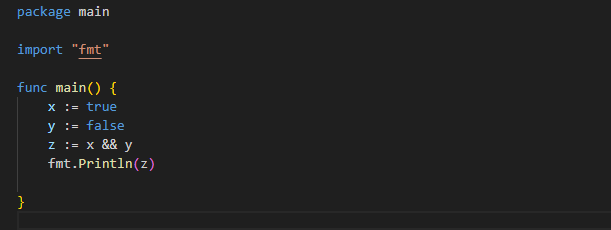
Operasi Logikas berlaku untuk nilai boolean dan menghasilkan hasil dengan tipe yang sama dengan operan. Operan yang tepat dievaluasi secara kondisional.



Berikut adalah beberapa operator logika yang tersedia dalam pemrograman Go beserta contohnya :

1. Operator Logika AND (&&):

Operator logika AND mengembalikan nilai true jika kedua ekspresi bernilai true, dan false jika salah satu atau keduanya bernilai false. Berikut adalah contoh penggunaannya pada pemrograman Go :

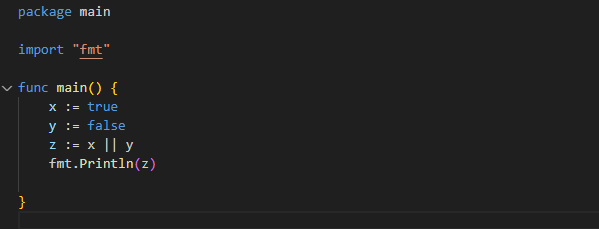


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Operator Logika OR (||):

Operator logika OR mengembalikan nilai true jika salah satu atau kedua ekspresi bernilai true, dan false jika keduanya bernilai false. Berikut adalah contoh penggunaannya pada pemrograman Go :

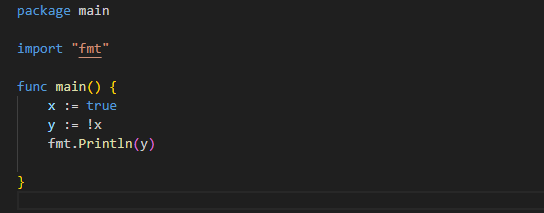


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Operator Logika NOT (!):

Operator logika NOT digunakan untuk membalikkan nilai ekspresi menjadi kebalikannya. Jika ekspresi bernilai true, maka NOT akan mengembalikan false dan sebaliknya. Berikut adalah contoh penggunaannya pada pemrograman Go :



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



## OPERASI PERBANDINGAN

Operasi perbandingan membandingkan dua operan dan menghasilkan nilai boolean yang tidak bertipe. Operasi perbandingan sering digunakan dalam ekspresi kondisional dan perulangan untuk mengontrol alur program berdasarkan hubungan perbandingan antara nilai-nilai.



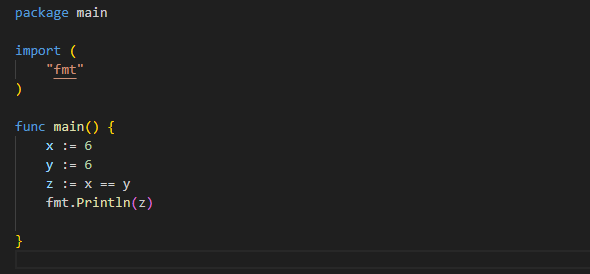
Dalam perbandingan apa pun, operan pertama harus dapat ditetapkan ke jenis operan kedua, atau sebaliknya. Operator kesetaraan == dan != berlaku untuk operan dari jenis yang sebanding . Operator pengurutan <, <=, >, dan >= berlaku untuk operan bertipe terurut . Istilah-istilah ini dan hasil perbandingan didefinisikan sebagai berikut :

1. Tipe data boolean dapat dibandingkan. Dua nilai boolean dianggap sama jika keduanya **true** atau keduanya **false**.
2. Tipe data integer dapat dibandingkan dan diurutkan. Dua nilai integer dapat dibandingkan dengan metode yang umum.
3. Tipe data floating-point dapat dibandingkan dan diurutkan. Dua nilai floating-point dapat dibandingkan seperti yang didefinisikan oleh standar IEEE-754.
4. Tipe data kompleks dapat dibandingkan. Dua nilai kompleks u dan v dianggap sama jika bagian real dari keduanya adalah sama (**real(u)** == **real(v)**), serta bagian imajiner dari keduanya juga sama **(imag(u) == imag(v))**.
5. Tipe data string dapat dibandingkan dan diurutkan. Dua nilai string dapat dibandingkan secara leksikal berdasarkan urutan byte karakternya.
6. Tipe data pointer dapat dibandingkan. Dua nilai pointer sama jika mereka menunjuk ke variabel yang sama atau jika keduanya memiliki nilai nil. Pointer yang menunjuk ke variabel dengan ukuran nol bisa jadi berbeda tergantung pada kasusnya.
7. Tipe data channel dapat dibandingkan. Dua nilai channel dianggap sama jika dibuat oleh panggilan yang sama ke fungsi **make** atau jika keduanya memiliki nilai **nil**.
8. Tipe data interface dapat dibandingkan. Dua nilai interface dianggap sama jika keduanya memiliki tipe dinamis yang identik dan nilai dinamis yang sama atau jika keduanya memiliki nilai **nil**.
9. Jika nilai x adalah tipe non-antarmuka X dan nilai t adalah tipe antarmuka T, maka keduanya dapat dibandingkan jika tipe X sebanding dan X mengimplementasikan T. Keduanya dianggap sama jika tipe dinamis t identik dengan X dan nilai dinamis t sama dengan x.
10. Tipe data struct dapat dibandingkan jika semua tipe bidangnya sebanding. Dua nilai struct dianggap sama jika nilai bidang non-kosong yang sesuai sama. Perbandingan dilakukan dalam urutan bidang sumber, dan proses perbandingan dihentikan segera setelah dua nilai bidang berbeda ditemukan (atau semua bidang telah dibandingkan).
11. Tipe data array dapat dibandingkan jika tipe elemen arraynya sebanding. Dua nilai array dianggap sama jika nilai elemen yang bersesuaian sama. Perbandingan dilakukan dalam urutan indeks yang menaik, dan proses perbandingan dihentikan segera setelah dua nilai elemen berbeda ditemukan (atau semua elemen telah dibandingkan).
12. Tipe data parameter dapat dibandingkan jika benar-benar sebanding.

Berikut adalah beberapa operator perbandingan yang tersedia dalam pemrograman Go beserta contohnya :

1. Operator Sama dengan (==)

Operator ini digunakan untuk memeriksa apakah dua nilai atau ekspresi sama. Berikut adalah contoh penggunaannya pada pemrograman Go :

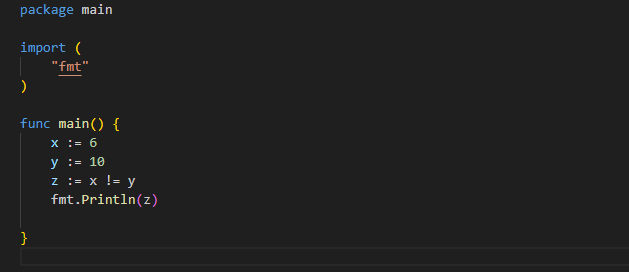


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Operator Tidak Sama dengan (!=)

Operator ini digunakan untuk memeriksa apakah dua nilai atau ekspresi tidak sama. Berikut adalah contoh penggunaannya pada pemrograman Go :

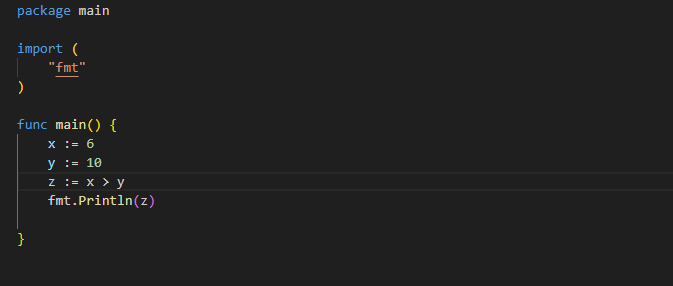


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Operator Lebih Besar dari (>)

Operator ini digunakan untuk memeriksa apakah nilai atau ekspresi pertama lebih besar dari nilai atau ekspresi kedua. Berikut adalah contoh penggunaannya pada pemrograman Go :

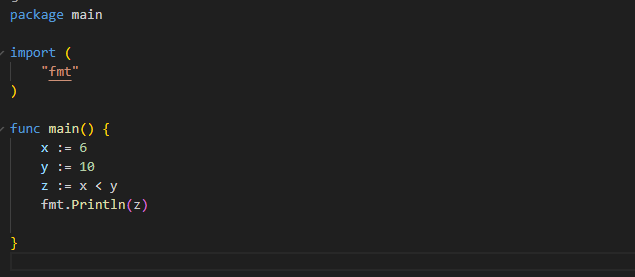


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Operator Lebih Kecil dari (< )

Operator ini digunakan untuk memeriksa apakah nilai atau ekspresi pertama lebih kecil dari nilai atau ekspresi kedua. Berikut adalah contoh penggunaannya pada pemrograman Go :

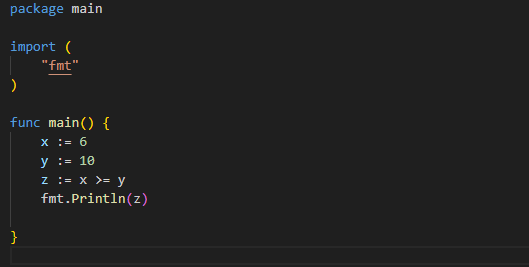


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Operator Lebih Besar atau Sama dengan ( >= )

Operator ini digunakan untuk memeriksa apakah nilai atau ekspresi pertama lebih besar dari atau sama dengan nilai atau ekspresi kedua. Berikut adalah contoh penggunaannya pada pemrograman Go :

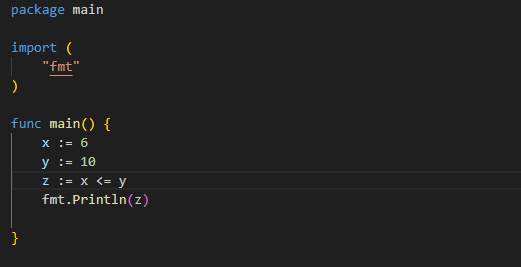


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Operator Lebih Kecil atau Sama dengan ( <= )

Operator ini digunakan untuk memeriksa apakah nilai atau ekspresi pertama lebih kecil dari atau sama dengan nilai atau ekspresi kedua. Berikut adalah contoh penggunaannya pada pemrograman Go :



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :

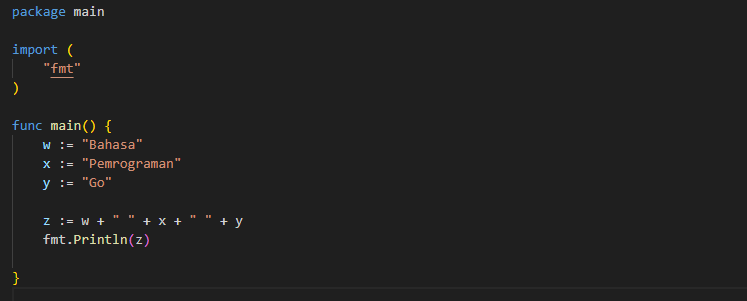


## OPERASI STRING

Operasi string dalam pemrograman Go merujuk pada serangkaian tindakan atau manipulasi yang dapat dilakukan pada data bertipe string. String dalam Go adalah urutan karakter yang diapit oleh tanda kutip ganda (""). Operasi string memungkinkan Anda untuk melakukan berbagai tugas seperti penggabungan, pemotongan, perbandingan, pencarian, dan transformasi pada string. Berikut adalah beberapa contoh operasi string yang umum dilakukan dalam pemrograman Go :

1. Penggabungan String (Concatenation)

Menggabungkan dua atau lebih string menjadi satu string yang lebih Panjang menggunakan operator + . Berikut adalah contoh penggunaannya pada pemrograman Go :

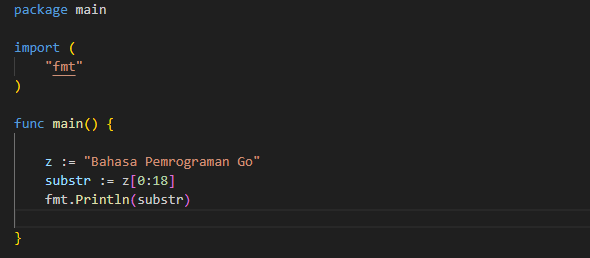


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Pemotongan String (Substring)

Mengambil atau memotong bagian tertentu dari sebuah string dengan menggunakan operator [:]. Berikut adalah contoh penggunaannya pada pemrograman Go :

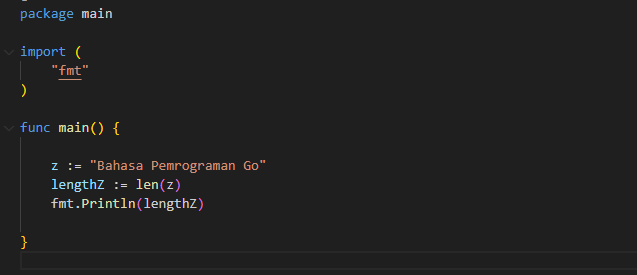


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Panjang String (Len)

Fungsi len() digunakan untuk mengembalikan panjang (jumlah karakter) dari sebuah string. Berikut adalah contoh penggunaannya pada pemrograman Go :

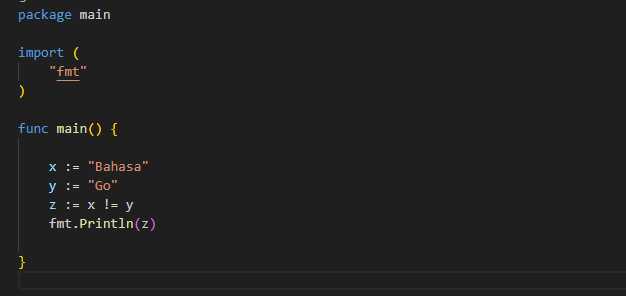


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Perbandingan String

Membandingkan dua string untuk mengecek kesamaan, keberbedaan, atau urutan leksikografis. Operator perbandingan seperti ==, !=, >, <, >=, <= juga bisa digunakan untuk membandingkan dua string. Berikut adalah contoh penggunaannya pada pemrograman Go :

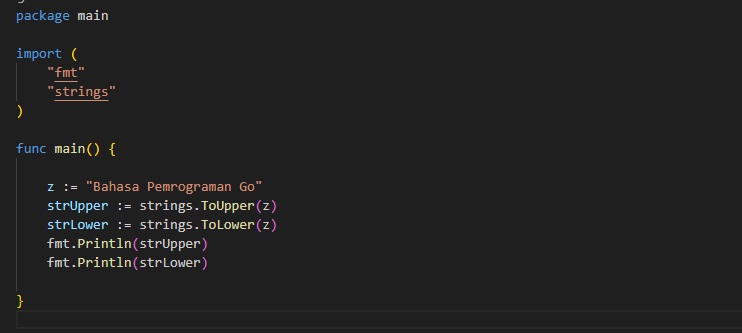


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Pengubahan Huruf Besar/ Huruf Kecil

Untuk melakukan pengubahan huruf besar atau huruf kecil dapat menggunakan fungsi **ToUpper()** dan **ToLower()**. Berikut adalah contoh penggunaannya pada pemrograman Go :

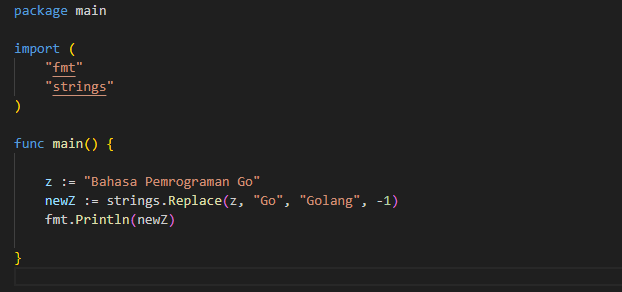


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Penggantian String (Replace)

Untuk menggantikan substring dalam string dengan substring lain dapat menggunakan fungsi **Replace()**. Berikut adalah contoh penggunaannya pada pemrograman Go :

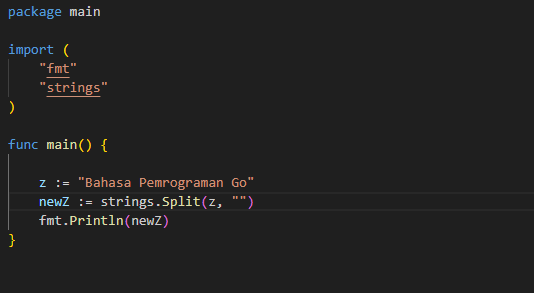


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Pemisahan

Untuk memisahkan string menjadi array berdasarkan delimeter tertentu dapat menggunakan fungsi **Split()**. Berikut adalah contoh penggunaannya pada pemrograman Go :

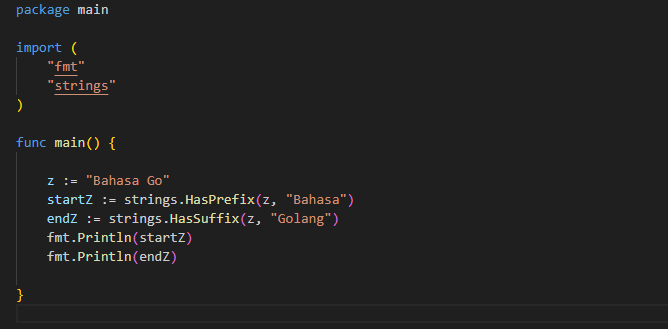


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Pengecekan Awaln dan Akhiran

Untuk memeriksa apakah sebuah string dimulai atau diakhiri dengan substring tertentu dapat menggunakan fungsi **HasPrefix()** dan **HasSuffix()**. Berikut adalah contoh penggunaannya pada pemrograman Go :



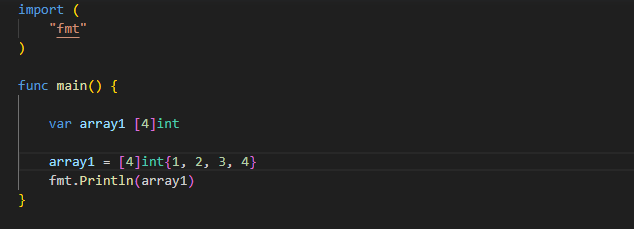
Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



## OPERASI ARRAY

Operasi array dalam pemrograman Go mencakup berbagai tindakan yang dapat dilakukan pada array, seperti inisialisasi, pengaksesan elemen, perubahan nilai elemen, pengulangan, dan perbandingan. Array dalam Go adalah kumpulan elemen dengan tipe data yang sama, yang disimpan dalam urutan tertentu. Berikut adalah contoh operasi array dalam pemrograman Go :

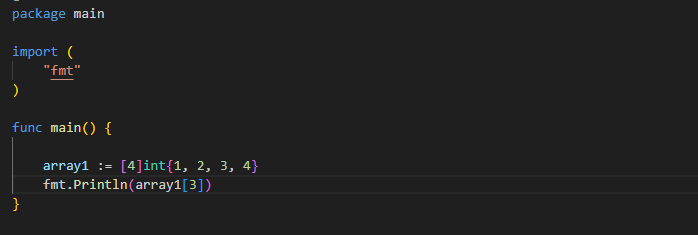
1. Inisialisasi Array : mendefiniskan dan menginisialisasi array dengan nilai awal.



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



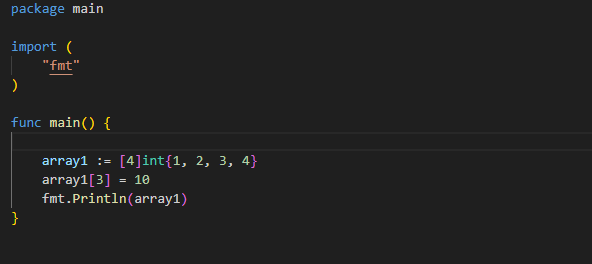
1. Akses Elemen Array : mengakses nilai elemen pada posisi tertentu dalam array.



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



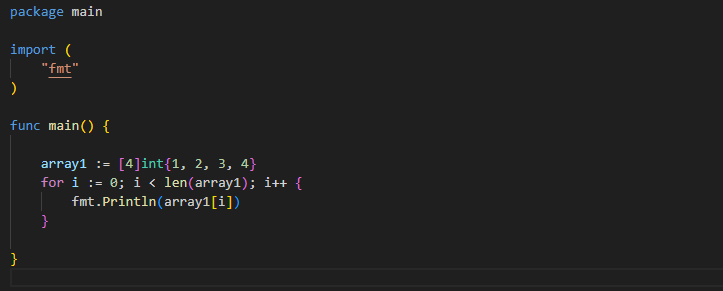
1. Perubahan Nilai Elemen Array : Mengubah nilai elemen pada posisi tertentu dalam array.



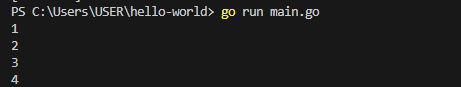
Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



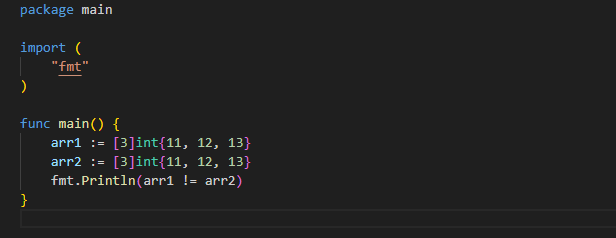
1. Pengulangan Array : Melakukan iterasi atau pengulangan elemen-elemen dalam array.



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



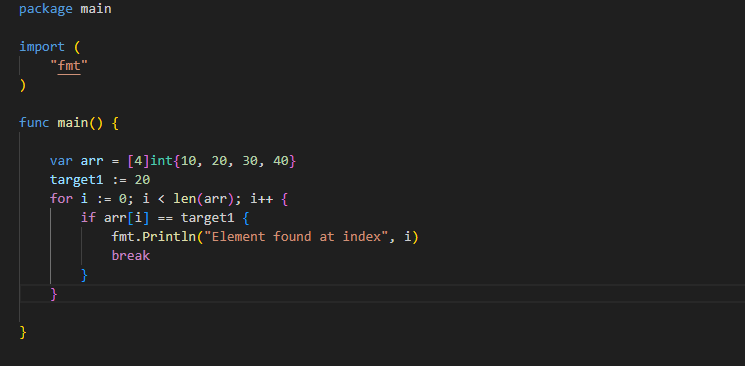
1. Perbandingan Array : Membandingkan dua array untuk mengecek kesamaan atau perbedaan elemen-elemennya.



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Pencarian Array : Melakukan pencarian elemen dalam array menggunakan perulangan dan kondisi.



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



# BAB 4 PENGOLAHAN DATA PADA GOLANG

Pengolahan data adalah proses mengumpulkan, mengorganisir, mengelompokkan, menganalisis, dan menginterpretasikan data untuk menghasilkan informasi yang berguna. Tujuan utama dari pengolahan data adalah untuk mengubah data mentah menjadi bentuk yang lebih berguna dan dapat dimengerti oleh manusia atau sistem komputer. Pengolahan data melibatkan beberapa langkah, antara lain :

1. Pengumpulan data

Data dikumpulkan dari berbagai sumber seperti survei, sensor, database, atau sumber lainnya. Data dapat berupa angka, teks, gambar, suara, atau jenis data lainnya.

1. Pembersihan data

Data mentah seringkali mengandung kesalahan, kehilangan, atau duplikasi. Pada tahap ini, data diperiksa, diperbaiki, dan disaring untuk memastikan kualitasnya sebelum dilanjutkan ke tahap selanjutnya.

1. Penyimpanan data

Data yang telah dikumpulkan dan dibersihkan disimpan dalam format yang sesuai, seperti basis data, file teks, atau format lainnya. Penyimpanan data yang baik penting untuk memudahkan akses dan penggunaan data di masa depan.

1. Analisis data

Data dianalisis menggunakan berbagai metode dan teknik seperti statistik, data mining, machine learning, atau visualisasi data. Tujuan dari analisis data adalah untuk mengidentifikasi pola, tren, hubungan, atau informasi penting lainnya yang terkandung dalam data.

1. Interpretasi dan presentasi data

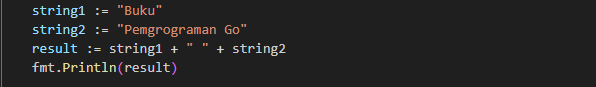
Hasil analisis data diterjemahkan menjadi informasi yang dapat dimengerti oleh pengguna. Informasi ini dapat disajikan dalam bentuk tabel, grafik, laporan, atau visualisasi data lainnya untuk memudahkan pemahaman dan pengambilan keputusan.

Pada umumnya, pengolahan data dalam Golang melibatkan sejumlah konsep dan teknik yang digunakan untuk memanipulasi, mengakses, memproses, dan menganalisis data.

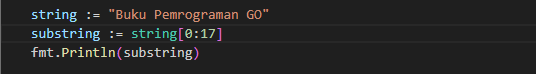
## STRING MANIPULASI

Memanipulasi string, termasuk operasi penggabungan, pemotongan, pencarian, penggantian, pemformatan, dan lainnya menggunakan fungsi-fungsi dalam package **strings** atau operasi string bawaan Go. Contoh penggunaannya pada kode program Bahasa Go adalah sebagai berikut :

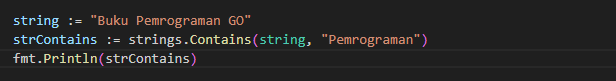
1. Penggabungan String : menggunakan operator **+**



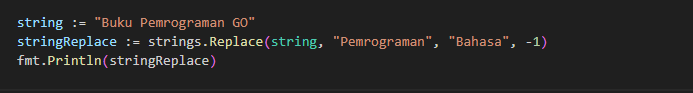
1. Pemotongan String : menggunakan **str[:]**



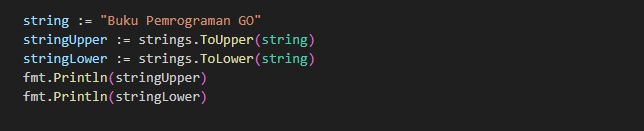
1. Pencarian String : menggunakan **contains()**



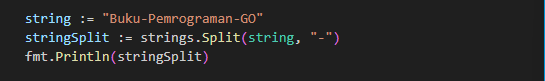
1. Penggantian String : menggunakan **replace()**



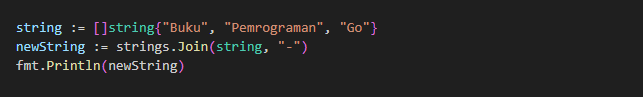
1. Pengubahan String ke huruf besar atau huruf kecil : menggunakan **ToUpper() dan ToLower()**



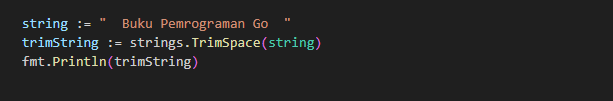
1. Pemisahan String :menggunakan **split()**



1. Penggabungan String dengan separator : menggunakan join



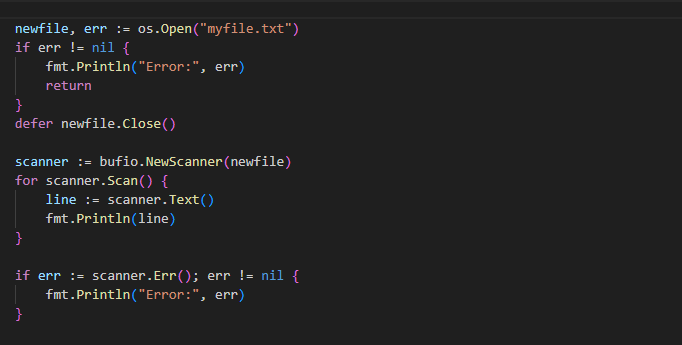
1. Menghilangkan spasi di awal dan di akhir string : menggunakan **TrimSpace()**



## OPERASI FILE

Membaca dan menulis data ke dalam file menggunakan package **os** dan **io**. Melibatkan pembukaan file, pembacaan isi file, penulisan data ke file, pengubahan ukuran file, penghapusan file, dan operasi lain terkait pengolahan file. Berikut ini adalah beberapa contoh proses operasi file :

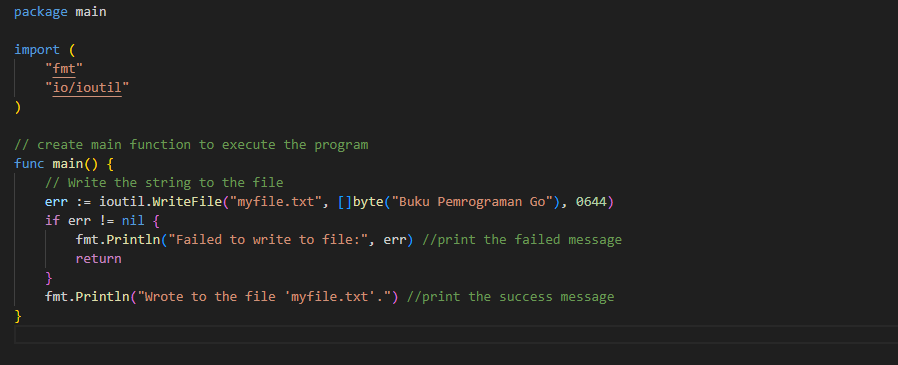
1. Membaca File menggunakan **os.Open**



1. Membuat File Baru menggunakan **os.Create**



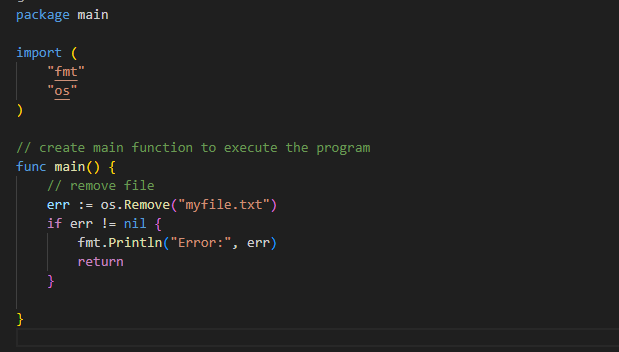
1. Membuat File Baru menggunakan **ioutil.WriteFile**



1. Membaca Semua Isi File menggunakan **ioutil.ReadFile**



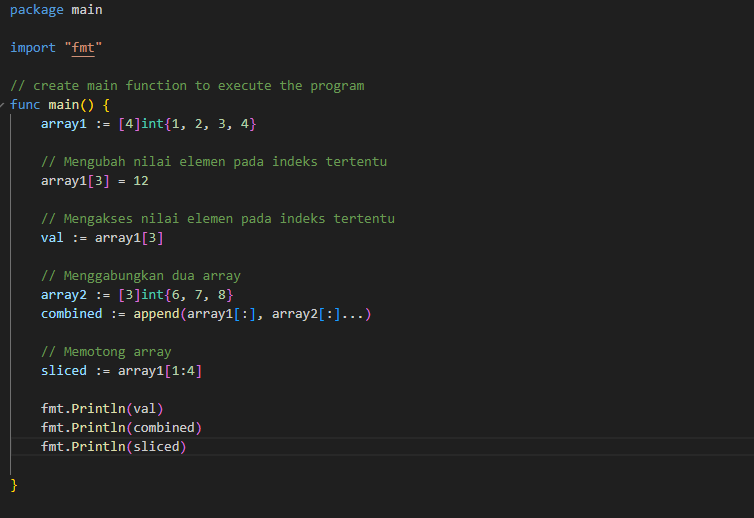
1. Menghapus File menggunakan **os.Remove**



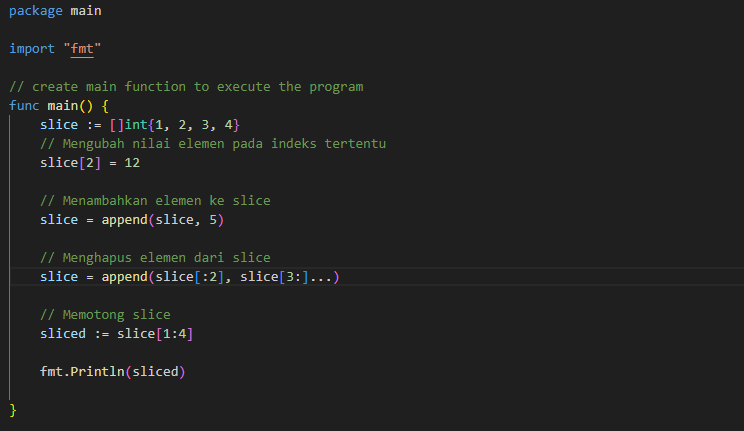
## MANIPULASI STRUKTUR DATA

Memanipulasi dan memproses data dalam struktur data seperti array, slice, map, dan struct. Termasuk operasi seperti akses elemen, perubahan nilai, pemotongan, pengurutan, pencarian, penambahan, penghapusan, dan operasi lain sesuai dengan jenis data struktur yang digunakan. Contoh manipulasi struktur data yang dapat dilakukan pada pemrograman Go :

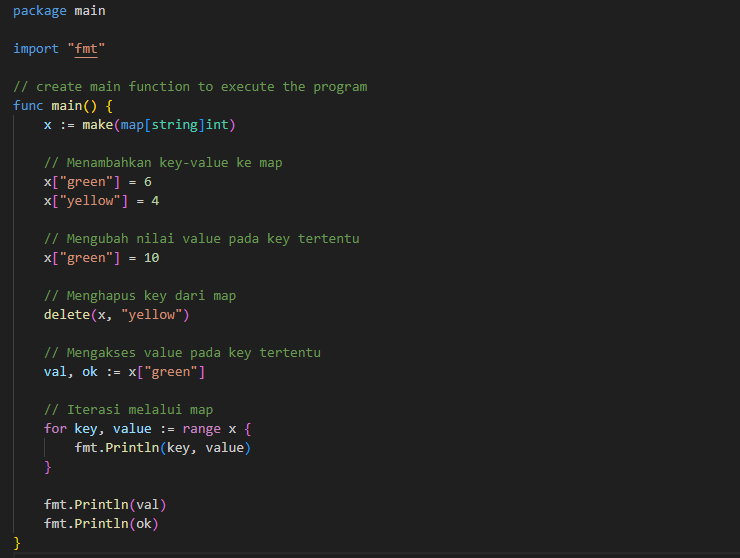
1. Manipulasi Array



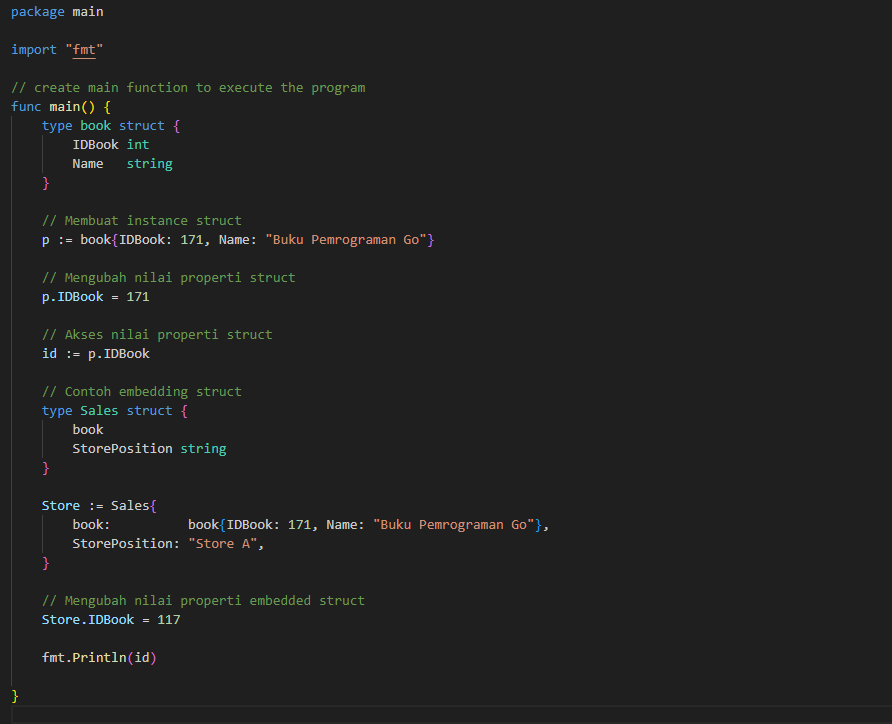
1. Manipulasi Slice



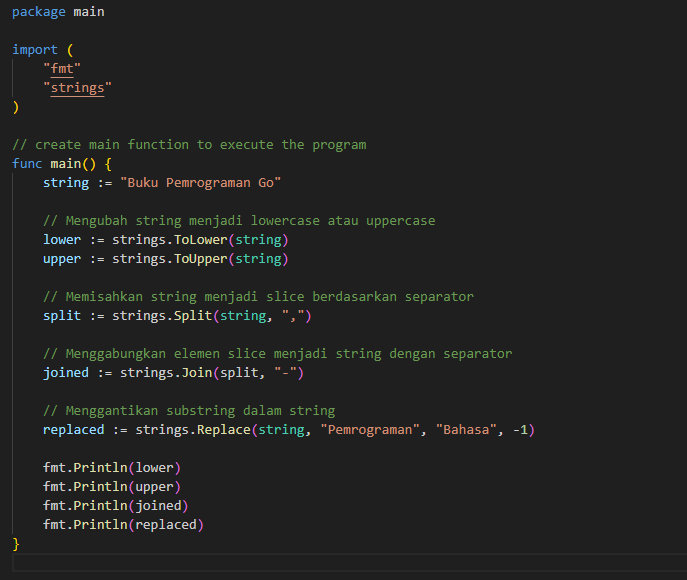
1. Manipulasi Map



1. Manipulasi Struct



1. Manipulasi String

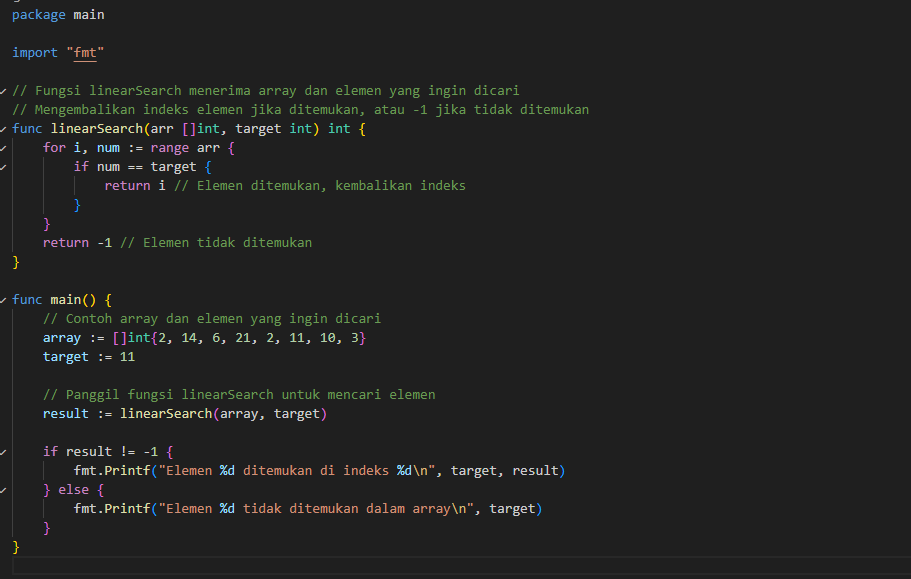


## PENCARIAN DAN PENGURUTAN

Melakukan pencarian dan pengurutan data menggunakan algoritma pencarian dan pengurutan yang sesuai, seperti pencarian linear, pencarian biner, pengurutan bubble sort, pengurutan quicksort, dan lainnya. Berikut contoh pencarian dan pengurutan yang dapat dilakukan dalam pemrograman Go :

1. Pencarian Linear

Dalam **linearsearch**, elemen dalam sebuah array dapat dicari secara berurutan dari awal hingga akhir array.



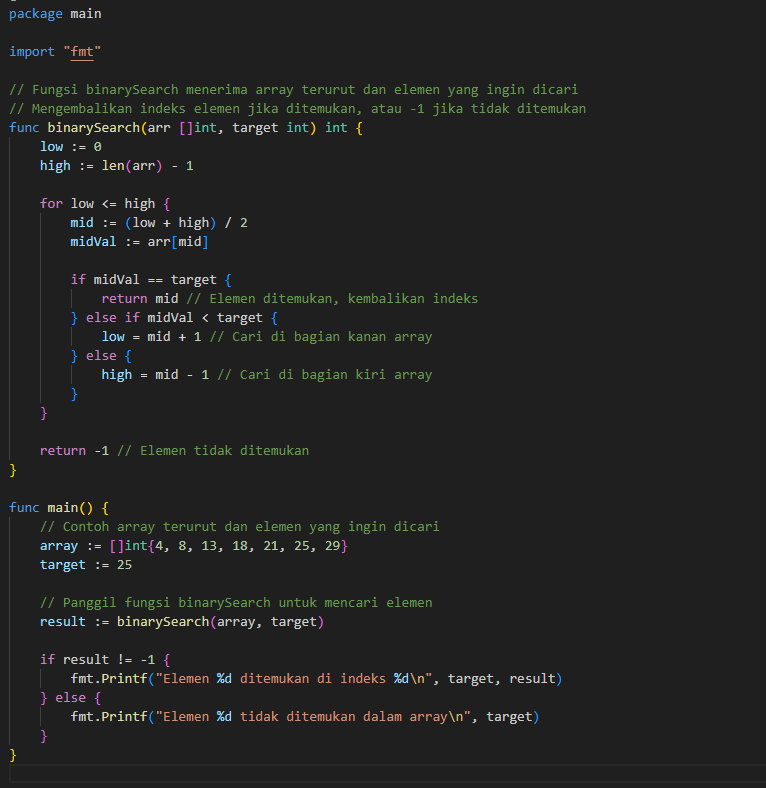
Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



Dalam contoh di atas, kita menggunakan fungsi **linearSearch** untuk mencari elemen **target** dalam array **arr**. Jika elemen ditemukan, fungsi akan mengembalikan indeksnya, dan jika tidak ditemukan, akan mengembalikan **-1**. Fungsi **linearSearch** menggunakan perulangan **for** untuk mengecek setiap elemen dalam array dan membandingkannya dengan **target**. Jika elemen ditemukan, fungsi akan langsung mengembalikan indeksnya. Jika tidak ada elemen yang cocok, fungsi akan mengembalikan **-1** sebagai penanda bahwa elemen tidak ditemukan.

1. Pencarian Binari

Dalam binary search, suatu elemen dalam array dapat dicari secara terurut dengan membagi array menjadi dua bagian dan mencari pada bagian yang sesuai berdasarkan perbandingan elemen tengah.

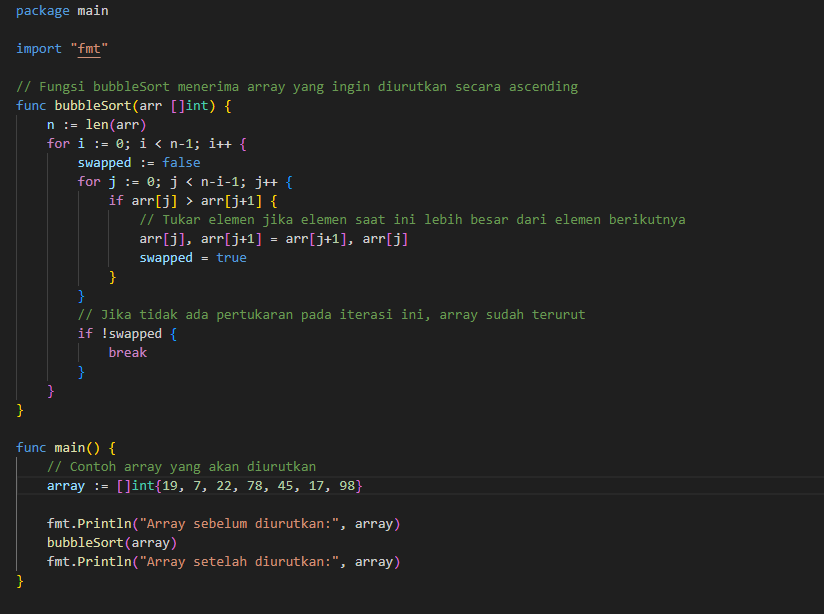


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



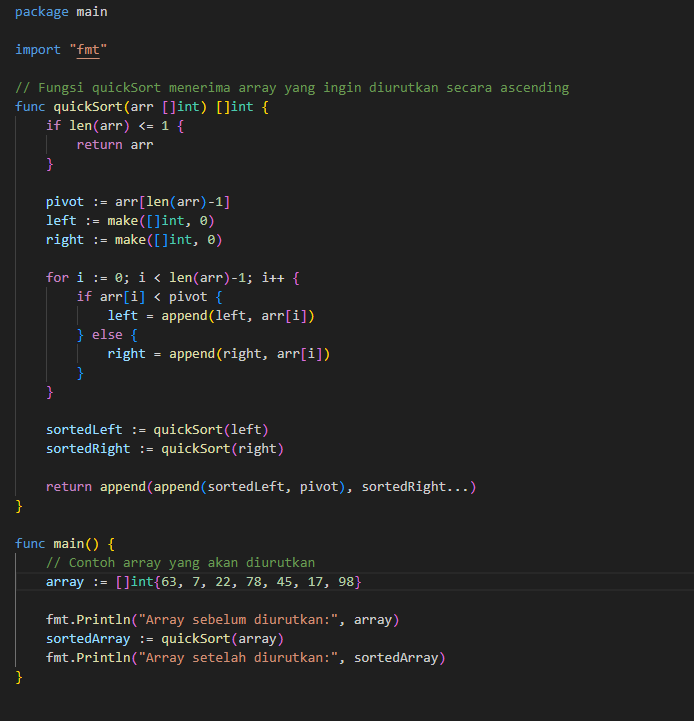
Dalam contoh di atas, fungsi **binarySearch** digunakan untuk mencari elemen **target** dalam array terurut **arr**. Fungsi tersebut menggunakan pendekatan pembagian dan pengurangan ukuran area pencarian untuk mencari elemen secara efisien. Jika elemen ditemukan, fungsi akan mengembalikan indeksnya. Jika tidak ada elemen yang cocok, fungsi akan mengembalikan -1 sebagai penanda bahwa elemen tidak ditemukan.

1. Bubble Sort



Dalam contoh di atas, kita menggunakan fungsi **bubbleSort** untuk mengurutkan array **arr** secara ascending. Bubble Sort bekerja dengan cara membandingkan pasangan elemen bersebelahan dan menukar posisi mereka jika urutan mereka tidak benar. Proses ini dilakukan berulang-ulang hingga seluruh array terurut dengan benar. Pada setiap iterasi, jika tidak ada pertukaran yang terjadi, ini berarti array sudah terurut dan proses pengurutan berhenti. Hal ini meningkatkan efisiensi Bubble Sort untuk array yang sudah hampir terurut. Namun, pada kasus terburuknya, yaitu saat array dalam urutan terbalik, **BubbleSort** membutuhkan banyak pertukaran dan memiliki kompleksitas waktu O(n^2).

1. Quick Sort



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :

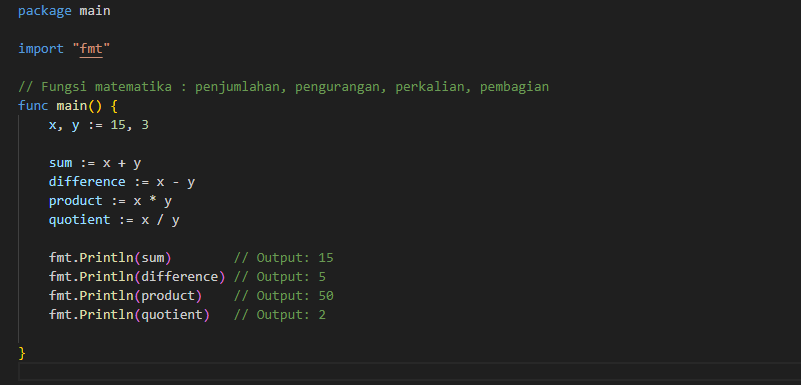


Pada contoh di atas, fungsi **quickSort** digunakan untuk mengurutkan array **arr** secara ascending. Quick Sort menggunakan pendekatan divide and conquer, di mana array dibagi menjadi dua bagian berdasarkan elemen pivot. Kemudian, kedua bagian tersebut diurutkan secara terpisah dan digabungkan kembali untuk menghasilkan array yang terurut. Fungsi **quickSort** mengambil elemen terakhir dari array sebagai pivot. Kemudian, elemen yang lebih kecil dari pivot ditempatkan di sebelah kiri, dan elemen yang lebih besar ditempatkan di sebelah kanan. Proses ini berulang secara rekursif hingga array terbagi menjadi bagian-bagian yang lebih kecil dan akhirnya diurutkan. Quick Sort memiliki kinerja yang baik untuk array dengan ukuran besar, tetapi pada kasus terburuknya ketika pivot selalu merupakan elemen terbesar atau terkecil, waktu eksekusi dapat mencapai O(n^2). Namun, rata-rata kompleksitas waktu Quick Sort adalah O (n log n) dengan rata-rata kasus terbaik yang sangat efisien.

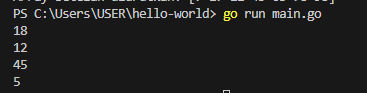
## OPERASI MATEMATIKA DAN STATISTIK

Menggunakan package math untuk melakukan perhitungan matematika dan statistik pada data numerik. Termasuk fungsi-fungsi matematika seperti penghitungan nilai mutlak, eksponensial, logaritma, trigonometri, statistik deskriptif, dan lainnya. Berikut ini adalah contoh proses operasi matematika dan statistic yang dapat dilakukan dalam pemrograman Go :

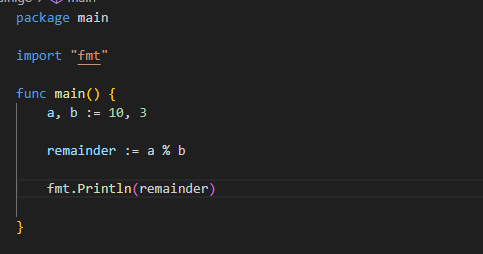
1. Operasi Penjumlahan, Pengurangan, Perkalian, Pembagian



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



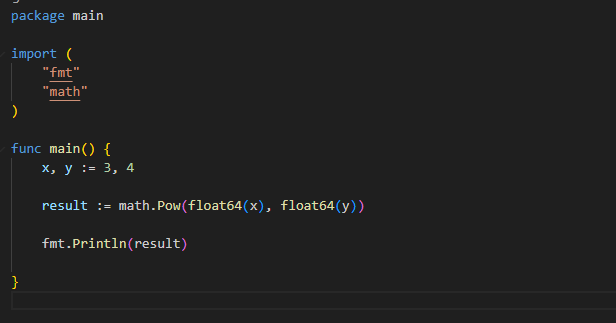
1. Modulo dan Pembagian Sisa



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



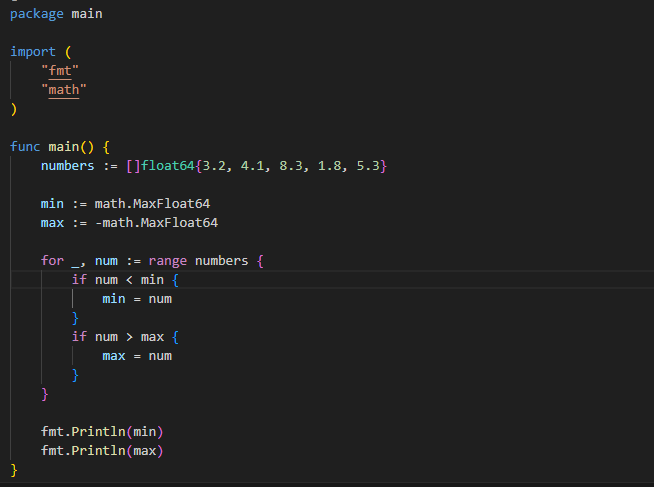
1. Operasi Pangkat



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



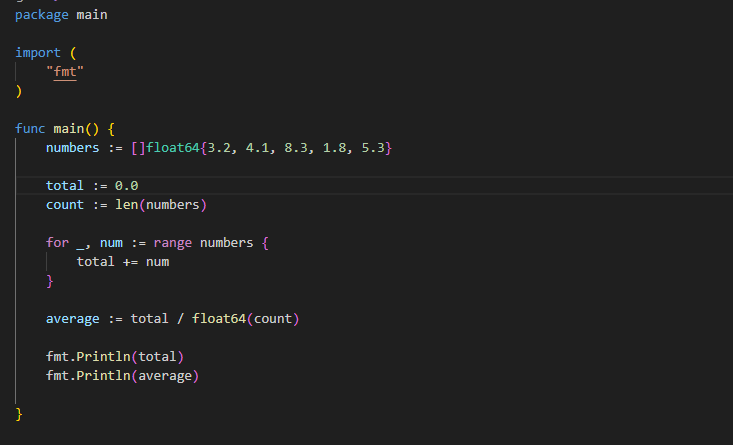
1. Minimum dan Maximum



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



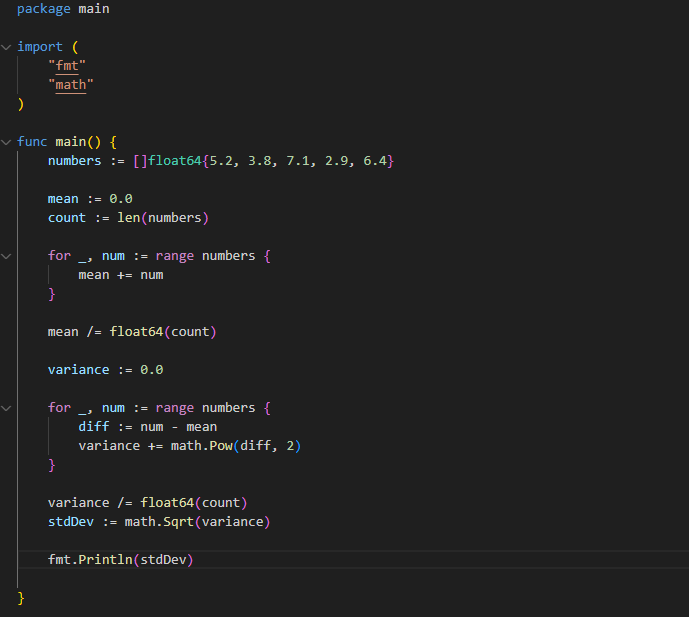
1. Rata-rata dan Total



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



1. Standar Deviasi

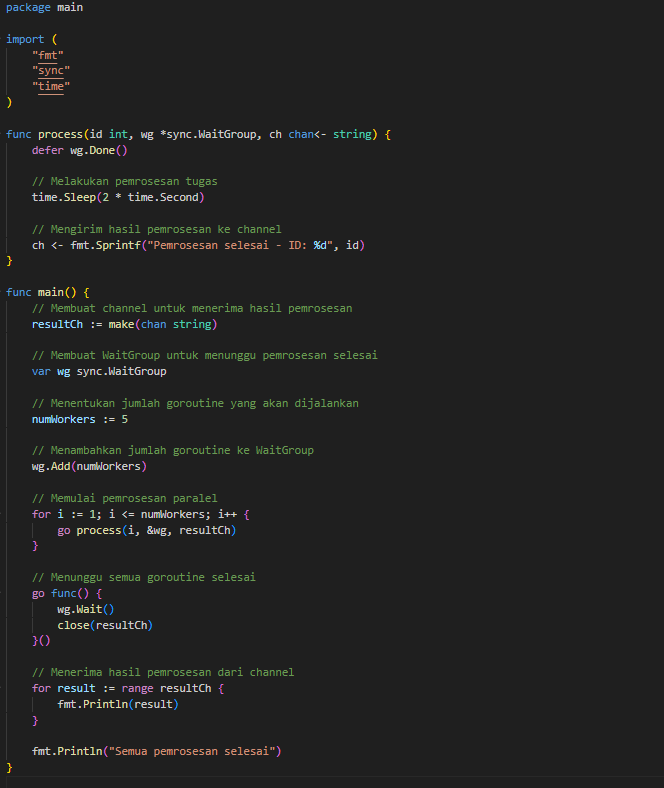


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :

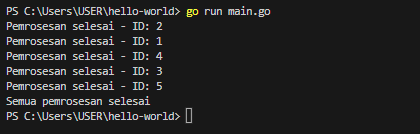


## PEMROSESAN PARALEL

Memanfaatkan fitur goroutine dan channel dalam Go untuk mendukung pemrograman paralel. Melibatkan menjalankan tugas secara konkuren dengan goroutine dan mengirim/menerima data antara goroutine menggunakan channel.

****

Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :

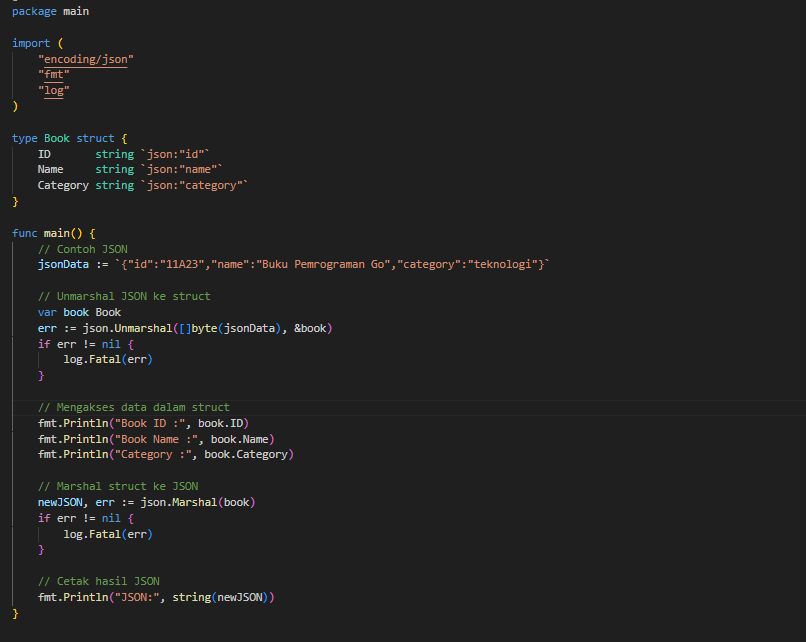
****

Pada contoh tersebut, goroutine digunakan untuk menjalankan pemrosesan paralel. Fungsi **process** dijalankan sebagai goroutine dengan melakukan pemrosesan tugas dan mengirimkan hasilnya ke channel **resultCh**. Kemudian, dalam fungsi **main**, kita membuat channel **resultCh** untuk menerima hasil pemrosesan, membuat **WaitGroup** untuk menunggu pemrosesan selesai, dan menentukan jumlah goroutine yang akan dijalankan. Setelah itu, kita memulai pemrosesan paralel dengan menjalankan goroutine process sebanyak jumlah goroutine yang ditentukan, dan menambahkan jumlah goroutine ke **WaitGroup**. Selanjutnya, kita menggunakan goroutine anonim untuk menunggu semua goroutine selesai dengan menggunakan WaitGroup, dan menutup channel **resultCh** setelah pemrosesan selesai. Terakhir, kita menerima hasil pemrosesan dari channel **resultCh** menggunakan loop for range dan mencetak hasilnya. Setelah semua hasil pemrosesan diterima, kita mencetak pesan "Semua pemrosesan selesai". Dengan menggunakan goroutine dan channel, pemrosesan paralel dapat dijalankan secara efisien dan melakukan komunikasi antar goroutine dengan menggunakan channel. Hal ini memungkinkan pemrosesan yang lebih cepat dan meningkatkan kinerja aplikasi.

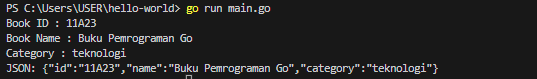
## PEMROSESAN JSON/XML

Membaca dan menulis data dalam format JSON (JavaScript Object Notation) dan XML (eXtensible Markup Language) menggunakan package encoding/json dan encoding/xml.

1. Pemrosesan JSON

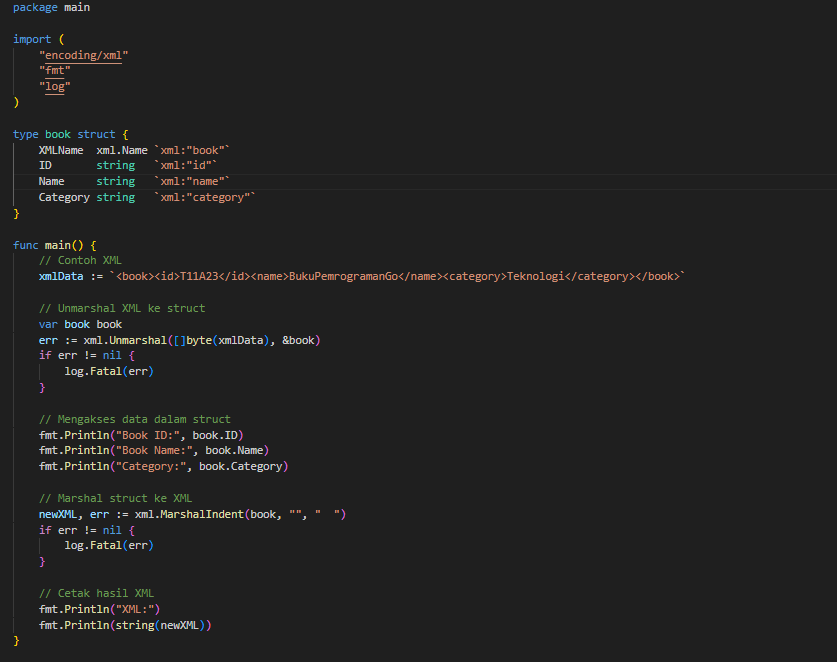


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :

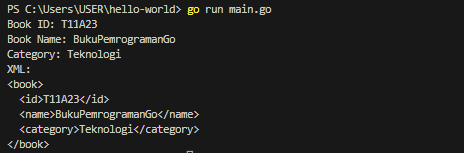


Pada pemrosesan JSON digunakan package **encoding/json.** Untuk pemrosesan JSON, kita menggunakan fungsi **json.Unmarshal** yang digunakan mengubah data JSON menjadi struct Go (Book dalam contoh tersebut) dan fungsi **json.Marshal** untuk mengubah struct menjadi data JSON. Kita juga dapat menggunakan tag **struct json:"..."** untuk menentukan nama field saat melakukan pemrosesan JSON.

1. Pemrosesan XML



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :

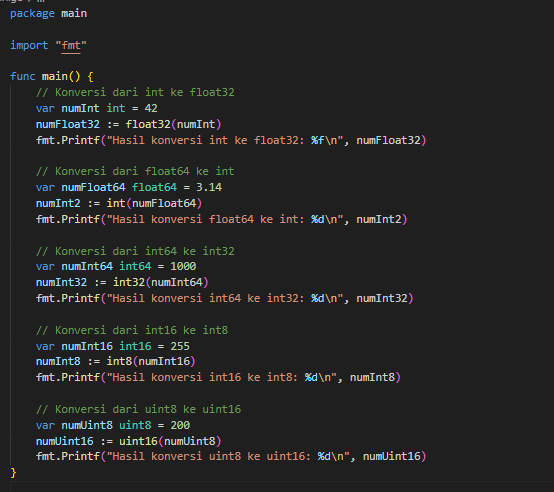


Pada pemrosesan JSON digunakan package **encoding/xml.** Untuk pemrosesan XML, kita menggunakan fungsi **xml.Unmarshal** untuk mengubah data XML menjadi struct Go (book dalam contoh tersebut) dan fungsi xml.MarshalIndent untuk mengubah struct menjadi data XML. Kita juga dapat menggunakan tag struct **xml:"..."** untuk menentukan nama elemen saat melakukan pemrosesan XML.

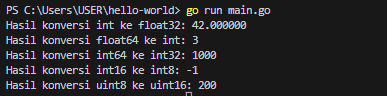
## OPERASI KONVERSI TIPE DATA

Mengubah tipe data dari satu bentuk ke bentuk lain menggunakan konversi tipe data. Misalnya, mengkonversi int menjadi float, string menjadi int, dan sebagainya. Berikut contoh pemrosesan konversi tipe data pada pemgrograman Go :

1. Konversi dari satu tipe numerik ke tipe numerik lainnya

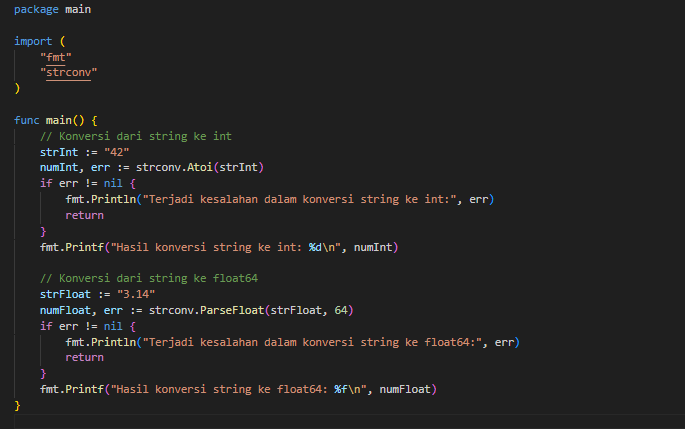


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



Pada contoh di atas, dilakukan beberapa konversi tipe numerik antara int, float32, float64, int8, int16, int32, dan uint8. Agar diperhatikan bahwa saat melakukan konversi, pastikan untuk memperhatikan batasan nilai dari tipe data yang akan dikonversi agar tidak terjadi kehilangan informasi atau overflow. Selain itu, pastikan nilai yang akan dikonversi sesuai dengan tipe data yang dituju, agar hasilnya benar dan sesuai dengan yang diharapkan.

1. Konversi dari tipe string ke tipe numerik

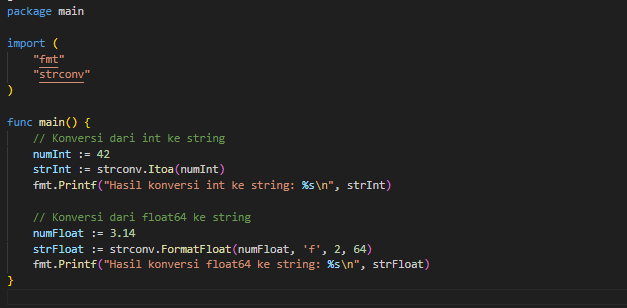


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



Pada contoh di atas, digunakan fungsi **strconv.Atoi** untuk mengkonversi string menjadi tipe data int dan fungsi **strconv.ParseFloat** untuk mengkonversi string menjadi tipe data float64. Perlu dicatat bahwa kedua fungsi tersebut mengembalikan dua nilai, nilai yang pertama adalah hasil konversi, dan nilai kedua adalah error jika terjadi kesalahan dalam konversi. Oleh karena itu, sebaiknya selalu melakukan penanganan error saat melakukan konversi dari tipe data string ke tipe data numerik untuk menghindari potensi masalah saat eksekusi program.

1. Konversi dari tipe numerik ke tipe string

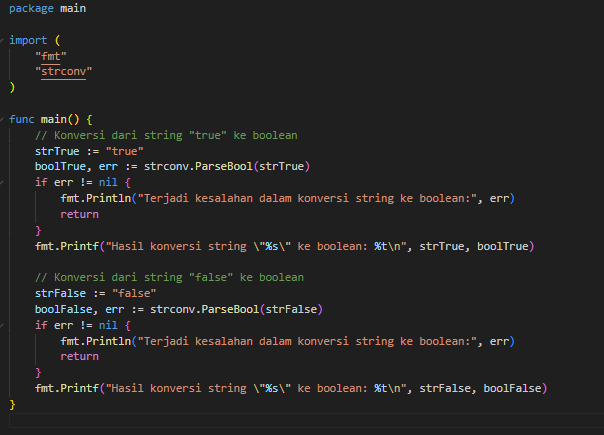


Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



Pada contoh di atas digunakan fungsi **strconv.Itoa** untuk mengkonversi tipe data int menjadi string dan **strconv.FormatFloat** untuk mengkonversi tipe data float64 menjadi string. Fungsi **strconv.FormatFloat** menerima beberapa argumen, di antaranya adalah **fmt**, **prec**, dan **bitSize**. Pada contoh di atas, kita menggunakan **f** untuk **fmt** yang mengindikasikan format floating-point, **2** untuk **prec** yang merupakan jumlah digit desimal yang ingin ditampilkan, dan **64** untuk **bitSize** yang menunjukkan bahwa tipe float yang digunakan adalah float64. Perhatikan bahwa dalam konversi dari tipe numerik ke string, kita bisa mengatur format tampilan sesuai kebutuhan, seperti menentukan jumlah digit desimal atau format bilangan floating-point.

1. Konversi dari tipe string ke tipe Boolean dapat dilakukan dengan menggunakan fungsi **strconv.ParseBool**. Fungsi ini akan mengembalikan dua nilai, yaitu nilai boolean hasil konversi dan error jika terjadi kesalahan dalam konversi.



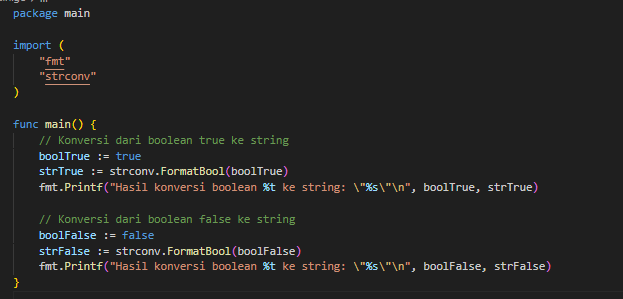
Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



Fungsi ini akan mengembalikan dua nilai, yaitu nilai boolean hasil konversi dan error jika terjadi kesalahan dalam konversi.

Pada contoh di atas, digunakan fungsi **strconv.ParseBool** untuk mengkonversi string **"true"** dan **"false"** menjadi nilai **boolean**. Fungsi ini mengenali string **"true"** (dalam huruf besar atau kecil) sebagai **true**, dan string **"false"** (dalam huruf besar atau kecil) sebagai **false**. Jika string yang diberikan tidak cocok dengan nilai boolean, maka fungsi akan mengembalikan error. Oleh karena itu, selalu penting untuk menangani error saat melakukan konversi tipe data.

1. Konversi dari tipe Boolean ke tipe string dapat dilakukan dengan menggunakan fungsi **strconv.FormatBool**. Fungsi ini akan mengembalikan string "true" jika nilai boolean adalah true, dan string "false" jika nilai boolean adalah false.



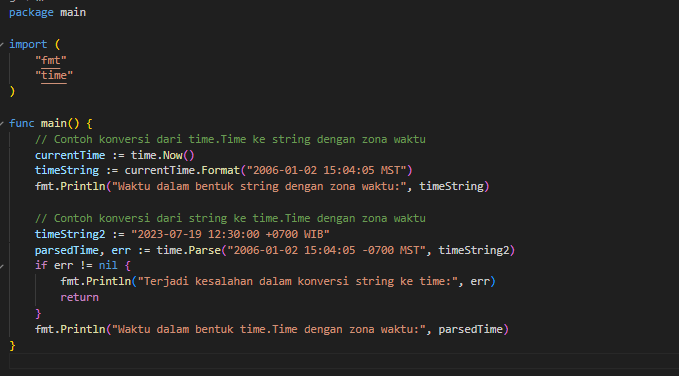
Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



Dalam contoh di atas, digunakan fungsi **strconv.FormatBool** untuk mengkonversi nilai boolean menjadi string "true" atau "false". Fungsi ini sangat berguna ketika kita perlu menyimpan nilai boolean dalam bentuk string atau mengirimkan nilai boolean sebagai parameter dalam URL atau data tipe string lainnya.

1. Konversi antara tipe data waktu (time)

Pada bahasa Go (Golang), konversi antara tipe data waktu (**time.Time**) dan string dapat dilakukan dengan menggunakan metode Time.Format untuk mengubah objek time.Time menjadi string dan fungsi time.Parse untuk mengubah string menjadi objek **time.Time**. Berikut adalah contoh konversi antara tipe data waktu (time.Time) dan string pada kode program bahasa Go :



Jika kode program tersebut dijalankan akan menampilkan output sebagai berikut :



Pada contoh di atas, digunakan metode **Time.Format** untuk mengonversi objek currentTime (tipe time.Time) menjadi string dengan format "2006-01-02 15:04:05 MST". Format "2006-01-02 15:04:05 MST" menyertakan zona waktu dalam format "MST" (misalnya, "WIB" untuk Waktu Indonesia Barat). Selanjutnya, kita menggunakan fungsi **time.Parse** untuk mengonversi **string timeString2** menjadi objek **parsedTime** (tipe time.Time) dengan zona waktu yang disertakan dalam format "2006-01-02 15:04:05 -0700 MST". Fungsi ini memerlukan format string yang sesuai dengan zona waktu yang ingin di-parse, dan string waktu itu sendiri. Jika berhasil, fungsi akan mengembalikan objek time.Time dengan zona waktu yang sesuai, jika gagal (misalnya, format tidak sesuai), maka akan mengembalikan error.

# BAB 5 MACHINE LEARNING PADA BAHASA PEMROGRAMAN GO

## KONSEP MACHINE LEARNING

Machine learning memungkinkan komputer untuk mengatur program secara otomatis. Ini mencapai otomatisasi dalam proses pemrograman. Namun, salah satu tantangannya adalah kekurangan pengembang berpengalaman dalam pembuatan perangkat lunak. Dengan adanya machine learning, masalah ini dapat diatasi, sehingga memungkinkan pembuatan perangkat lunak menjadi lebih terjangkau dan mudah dilakukan.

Tidak seperti pemrograman tradisional yang menjalankan data dan program di komputer untuk menghasilkan output, machine learning memungkinkan data dan output dijalankan di komputer untuk menciptakan program. Program ini kemudian dapat digunakan dalam pemrograman tradisional.

Machine learning adalah bidang ilmu komputer yang berfokus pada pengembangan algoritma dan teknik yang memungkinkan sistem komputer untuk belajar dari data tanpa harus diprogram secara eksplisit. Tujuan utamanya adalah untuk membuat model atau algoritma yang dapat mengidentifikasi pola dan menerapkan pengetahuan dari data pelatihan ke data baru yang belum pernah dilihat sebelumnya.

Machine learning memiliki banyak aplikasi praktis. Berikut adalah contoh-contoh penerapan machine learning :

* Pencarian web : Pengurutan halaman berdasarkan tingkat kemungkinan user mengkliknya.
* Biologi komputasi : Pengembangan/desain obat di komputer berdasarkan pengalaman eksperimen sebelumnya.
* Keuangan : Menentukan calon penerima penawaran kartu kredit, menilai risiko dalam tawaran kredit, Pengambilan keputusan tentang alokasi investasi keuangan.
* E-commerce : Mengantisipasi perpindahan pelanggan (customer churn), Mendeteksi keaslian suatu transaksi.
* Eksplorasi ruang angkasa : pesawat luar angkasa dan radio astronomi.
* Robotika : Cara menghadapi ketidakpastian dalam lingkungan yang baru, kendaraan otonom, mobil tanpa pengemudi.
* Ekstraksi informasi : Ajukan pertanyaan melalui database di seluruh web, mengajukan pertanyaan melalui basis data di seluruh web.
* Jejaring social : Informasi mengenai hubungan dan preferensi, penerapan pembelajaran mesin untuk menggali nilai dari data tersebut.
* Debugging : Diterapkan dalam isu-isu ilmu komputer seperti debugging, proses yang membutuhkan usaha dan pekerjaan intensif, mampu memberikan saran mengenai kemungkinan lokasi bug tersebut.

Pada machine learning terdapat komponen-komponen yang dibutuhkan. Semua komponen bekerja bersama-sama untuk menciptakan sistem machine learning yang dapat mengenali pola, memahami data, dan memberikan solusi atau prediksi yang berharga dalam berbagai bidang dan aplikasi. Komponen-komponen tersebut meliputi :

1. Data : Data adalah elemen paling krusial dalam machine learning. Ini berfungsi sebagai bahan bakar untuk melatih model dan mempengaruhi kualitas prediksi. Data pelatihan berisi contoh-contoh yang digunakan untuk mengajari model, sedangkan data uji digunakan untuk menguji kinerja model pada data yang belum pernah dilihat sebelumnya.
2. Model : Model adalah representasi matematis dari algoritma machine learning. Model ini mempelajari pola dan hubungan dalam data pelatihan dan menggunakannya untuk membuat prediksi pada data baru. Jenis model beragam, seperti regresi linear, jaringan saraf tiruan, pohon keputusan, dan masih banyak lagi.
3. Algoritma : Algoritma machine learning adalah kumpulan aturan dan prosedur matematis yang digunakan untuk melatih model dan membuat prediksi. Algoritma ini berperan penting dalam mencari pola dalam data dan mengoptimalkan kinerja model.
4. Fitur (Features) : Fitur adalah variabel atau atribut dalam data yang digunakan oleh model untuk mengidentifikasi pola. Memilih fitur yang relevan dan representatif adalah bagian penting dalam proses machine learning.
5. Pelatihan (Training) : Pelatihan adalah tahap di mana model belajar dari data pelatihan untuk mengenali pola dan melakukan penyesuaian dalam algoritmanya. Proses ini melibatkan mencocokkan model dengan data pelatihan dan mengoptimalkannya.
6. Validasi (Validation) : Validasi digunakan untuk mengukur sejauh mana model mampu melakukan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya. Hal ini memungkinkan evaluasi kinerja model sebelum digunakan pada data baru.
7. Uji Coba (Testing) : Setelah model dilatih dan divalidasi, langkah selanjutnya adalah mengujinya pada data uji yang belum pernah dilihat sebelumnya. Ini memberikan gambaran tentang seberapa baik model dalam menggeneralisasi pola dan melakukan prediksi pada data baru.
8. Optimisasi : Optimisasi melibatkan penyesuaian parameter atau hiperparameter model untuk mencapai performa terbaik. Tujuannya adalah untuk meningkatkan kualitas prediksi dan mengurangi kesalahan.
9. Prediksi dan Keputusan : Setelah melalui tahap pelatihan, model dapat digunakan untuk melakukan prediksi atau membuat keputusan berdasarkan data baru yang diberikan kepadanya.
10. Evaluasi : Evaluasi kinerja model dilakukan dengan menggunakan metrik tertentu, seperti akurasi, presisi, recall, atau F1-score. Ini membantu menilai seberapa baik model dalam memecahkan masalah yang diberikan.
11. Deployment (Penyajian) : Tahap terakhir adalah deployment, di mana model yang telah dilatih dan diuji diimplementasikan dalam aplikasi nyata atau lingkungan produksi untuk menyelesaikan tugas yang ditetapkan.

## PERAN SERTA KELEBIHAN DAN KEKURANGAN BAHASA GO UNTUK MACHINE LEARNING

Menerapkan AI memungkinkan perusahaan untuk mengembangkan beragam aplikasi inovatif yang dapat disesuaikan dengan kebutuhan mereka. Meskipun begitu, kerangka kerja web harus mendukung AI. Namun, apa yang membuat Go istimewa? Popularitas bahasa Go dalam pengembangan AI karena sifat Bahasa Go yang cepat dan dapat diskalakan, aplikasi yang dibangun di atasnya menjadi sangat diminati oleh banyak pengembang.

Identifikasi masalah dalam Golang sangatlah mudah karena tipe data pada Golang ditentukan secara statis saat diketik dan dikompilasi. Selain itu, fitur kompilasi kode mesin secara real-time dalam Bahasa Go memungkinkan siklus pengeditan dan penyegaran yang cepat, sambil tetap menghasilkan kode mesin yang efisien. Karena alasan ini, bahasa pemrograman Go sangat populer untuk berbagai aplikasi pembelajaran mesin.

Bahasa Go menjadi pilihan yang menarik bagi para pengembang karena kemampuannya dalam menulis program yang sangat konkuren. Meskipun dalam Go, fungsi pembelajaran terbatas, namun komunitas Go terus berkembang dari waktu ke waktu. Infrastruktur yang mendukung Go mencakup berbagai fitur pengujian yang memungkinkan pendekatan teknik yang lebih baik dengan kemudahan mengidentifikasi dan menguji modul.

Keuntungan utama dari Go adalah kemampuannya untuk berjalan di berbagai platform, sehingga menjadikannya bahasa yang ideal untuk melakukan migrasi perangkat lunak. Selain itu, kompiler Go memberlakukan penggunaan variabel yang tidak digunakan dan memiliki sistem pemformatan bawaan yang mempermudah penulisan kode.

Go juga memiliki keunikan dalam kemampuan konkurensi dan paralelisme bawaannya. Inilah yang membuat Golang menjadi platform yang ideal untuk mengembangkan aplikasi yang membutuhkan intensitas konkurensi tinggi, seperti aplikasi jaringan, data besar, dan lain-lain. Secara khusus, Go sangat cocok untuk aplikasi yang beroperasi di lingkungan cloud. Selain itu, ekosistem Golang terus berkembang dan mendukung pengembangan aplikasi pembelajaran mesin, membuatnya menjadi pilihan yang menarik bagi para pengembang. Kemampuan Go dalam otomatisasi browser juga membuatnya menjadi pilihan utama bagi pengembang. Dengan kecakapannya dalam membangun aplikasi cloud-native dan mengembangkan CLI (Command Line Interface) yang efisien dan elegan, serta mendukung DevOps dan SRE (Site Reliability Engineering), dan mampu mengimplementasikan database dan membangun situs web, Go sangat cocok untuk proyek-proyek semacam itu.

Bahasa pemrograman Go (biasa disebut Golang) telah mendapatkan popularitas dalam lingkungan machine learning karena beberapa kelebihan diantaranya :

* Kinerja

Bahasa Go diketahui memiliki kinerja yang baik dan performa yang tinggi. Kode Bahasa Go dieksekusi secara cepat dan efisien, sehingga cocok untuk pemrosesan data dan perhitungan yang intensif dalam machine learning.

* Efisiensi Penggunaan Sumber Daya

Go dirancang dengan manajemen memori yang efisien dan sistem pengelolaan tata letak (garbage collection) yang canggih. Hal ini memungkinkan penggunaan sumber daya komputer yang lebih efisien.

* Kemudahan Penggunaan

Bahasa Go dirancang dengan sintaks yang sederhana dan mudah dipahami. Hal ini memudahkan pengembang untuk menulis kode yang lebih bersih dan ringkas dalam proyek machine learning.

* Konkurensi

Bahasa Go memiliki fitur konkurensi yang bawaan, seperti goroutine, yang memungkinkan pengembang untuk dengan mudah mengelola tugas-tugas paralel. Kemampuan ini bermanfaat dalam mengolah data secara paralel, yang sering terjadi dalam machine learning.

* Pengemasan

Bahasa Go memiliki alat bawaan untuk pengemasan (packaging) kode, seperti Go Modules, yang memudahkan manajemen dependensi dan distribusi proyek machine learning.

* Open Source Library

Meskipun Bahasa Go bukan bahasa pemrograman utama dalam ekosistem machine learning seperti Python atau R, ada sejumlah pustaka open source yang tersedia untuk Bahasa Go yang mendukung pengembangan aplikasi machine learning, seperti Gorgonia, TensorFlow Go, dan Pachyderm.

* Produksi

Bahasa Go dikenal karena kestabilan dan kehandalan, sehingga cocok untuk pengembangan sistem machine learning yang diproduksi dan digunakan secara luas.

* Kestabilan dan Keandalan

Go dikenal karena kestabilan dan kehandalan. Bahasa ini telah digunakan dalam berbagai proyek produksi yang menunjukkan kemampuan untuk bekerja dengan baik dalam lingkungan yang kritis dan tuntutan skalabilitas tinggi.

Meskipun Bahasa Go telah menjadi pilihan yang menarik bagi beberapa pengembang dalam lingkungan machine learning, penting untuk diingat bahwa bahasa lain seperti Python masih mendominasi dalam ekosistem machine learning karena dukungan pustaka yang lebih luas dan komunitas yang besar. Pilihan bahasa pemrograman untuk proyek machine learning harus dipertimbangkan dengan matang berdasarkan kebutuhan proyek, kemampuan tim pengembang, dan ekosistem yang tersedia. Adapun kekurangan Bahasa Go dalam machine learning diantaranya :

* Ekosistem Machine Learning yang Terbatas

Ekosistem pustaka dan alat untuk machine learning di Go jauh lebih terbatas dibandingkan dengan bahasa pemrograman seperti Python. Bahasa Python memiliki banyak pustaka terkenal seperti TensorFlow, PyTorch, dan scikit-learn yang mendukung pengembangan machine learning dengan berbagai fungsi dan model.

* Kurangnya Pustaka Populer

Karena ekosistem yang terbatas, beberapa pustaka machine learning di Go mungkin belum seoptimal atau sepopuler pustaka yang ada dalam bahasa lain. Ini dapat membatasi akses ke teknologi dan inovasi terbaru dalam machine learning.

* Keterbatasan Komunitas

Go tidak memiliki komunitas machine learning sebesar bahasa seperti Python. Sebagai hasilnya, mungkin lebih sulit untuk menemukan dukungan, panduan, atau solusi untuk masalah yang khusus terkait dengan pengembangan machine learning dalam Go.

* Keterbatasan Algoritma

Beberapa algoritma machine learning cenderung lebih rumit dan memerlukan lebih banyak sumber daya komputasi daripada yang dapat ditangani dengan baik oleh Go. Karena itu, bahasa ini mungkin kurang cocok untuk pengembangan model machine learning yang kompleks.

* Perkembangan Lambat

Meskipun Go terus berkembang dan mendapatkan popularitas dalam berbagai bidang, perkembangan bahasa ini dalam konteks machine learning tidak setinggi bahasa pemrograman lain seperti Python yang telah lama menjadi bahasa utama untuk pengembangan machine learning.

## LIBRARY BAHASA GO UNTUK MEMBANGUN MESIN LEARNING

Pada saat ini, bahasa Go belum memiliki ekosistem pustaka machine learning sebesar bahasa-bahasa seperti Python atau R. Meskipun begitu, beberapa pustaka dan proyek open-source dalam bahasa Go sedang berkembang untuk membangun aplikasi machine learning. Meskipun belum sekomprehensif bahasa-bahasa lain, pustaka-pustaka tersebut menyediakan beberapa fitur yang dapat mendukung pengembangan proyek machine learning dalam bahasa Go. Berikut ini adalah beberapa pustaka yang patut dipertimbangkan :

1. GoLearn

GoLearn adalah pustaka yang sangat penting bagi bahasa Golang. Pustaka ini menyediakan berbagai algoritma Machine Learning yang dapat digunakan untuk berbagai keperluan. Density Based Spatial Clustering (DBSCAN), Random Forest (RF), K-Nearest Neighbors (KNN), Naïve Bayes (NB), Neural Network (NN) dan Component Analysis (PCA) adalah algoritme Machine Learning utama dari library ini. Berikut adalah langkah-langkah untuk menginstall library GoLearn :

* Buka Terminal atau Command Prompt
* Jalankan perintah berikut untuk menginstall Golearn



Perintah tsb akan mengunduh dan menginstall library Golearn ke direktori Go workspace pada computer yang digunakan.

* Berikut ini adalah contoh program Go menggunakan library GoLearn :



1. Tensorflow Go

TensorFlow adalah salah satu kerangka kerja machine learning yang paling terkenal dan mendukung bahasa Go. Dalam TensorFlow, bahasa Go dapat digunakan untuk membangun model dan melakukan pelatihan pada data. Berikut adalah langkah-langkah untuk menginstall library Tensorflow Go :

* Buka Terminal atau Command Prompt
* Jalankan perintah berikut untuk menginstall TensorFlow Go



Perintah tsb akan mengunduh dan menginstall library TensorFlow Go ke direktori Go workspace pada computer yang digunakan.

1. Gorgonia

Gorgonia adalah sebuah library yang secara khusus dikembangkan untuk bahasa Go. Library ini memungkinkan pengguna untuk membangun model machine learning dengan menggunakan algoritma deep learning seperti neural network dan convolutional neural network. Berikut adalah langkah-langkah untuk menginstall library Gorgonia :

* Buka Terminal atau Command Prompt
* Jalankan perintah berikut untuk menginstall Gorgonia



Perintah tsb akan mengunduh dan menginstall library Gorgonia ke direktori Go workspace pada computer yang digunakan.

* Berikut ini adalah contoh program Go menggunakan library Gorgonia :





1. Goml

Goml adalah proyek open-source yang menyediakan implementasi algoritma machine learning dalam bahasa Go (Golang). Proyek ini bertujuan untuk memberikan pilihan bagi pengembang yang ingin membangun aplikasi machine learning dengan menggunakan bahasa Go. Goml menyediakan berbagai algoritma machine learning, termasuk klasifikasi, regresi, clustering, dan lainnya, sehingga memungkinkan pengguna bahasa Go untuk melakukan analisis data dan pembelajaran mesin dengan lebih mudah dan efisien. Berikut adalah langkah-langkah untuk menginstall library goml :

* Buka Terminal atau Command Prompt
* Jalankan perintah berikut untuk menginstall Goml



Perintah tsb akan mengunduh dan menginstall library Goml ke direktori Go workspace pada computer yang digunakan.

* Berikut ini adalah contoh program Go menggunakan library Goml :



## TEKNIK PREPROCESSING DATA PADA BAHASA GO

Preprocessing adalah teknik yang digunakan untuk mengubah data mentah menjadi format yang lebih berguna dan efisien. Hal ini diperlukan karena data mentah seringkali tidak lengkap dan memiliki format yang tidak konsisten. Kualitas data memiliki dampak langsung pada keberhasilan setiap proyek yang melibatkan analisis data. Preprocessing melibatkan validasi dan imputasi data. Tujuan dari validasi adalah untuk mengevaluasi tingkat kelengkapan dan keakuratan data yang telah disaring.

Di sisi lain, tujuan dari imputasi adalah untuk memperbaiki kesalahan dan mengisi nilai yang hilang, baik secara manual atau otomatis melalui program business process automation (BPA). Data preprocessing umumnya diterapkan pada aplikasi yang berbasis aturan (rule-based) dan berbasis database. Dalam machine learning, langkah ini sangat penting untuk memastikan bahwa big data telah diformat dengan benar dan informasinya dapat diinterpretasikan dan digunakan oleh algoritma perusahaan.

Dalam pemrograman Go (Golang), konsep preprocessing data tidak umum digunakan seperti pada beberapa bahasa pemrograman lain seperti C atau C++. Namun, dalam konteks pengolahan data sebelum pemrosesan lebih lanjut, beberapa proses yang sering dilakukan sebagai bagian dari persiapan data dalam Go meliputi :

1. Membersihkan Data

Proses ini melibatkan pembersihan dan penghapusan karakter atau elemen yang tidak diinginkan atau tidak valid dari data. Misalnya, menghapus spasi ekstra, karakter khusus, atau tanda baca yang tidak relevan.

1. Pemformatan Data

Pemformatan data melibatkan mengatur data dalam format yang konsisten dan sesuai dengan kebutuhan aplikasi. Misalnya, mengubah format tanggal, angka, atau string menjadi format yang diinginkan.

1. Validasi Data

Validasi data melibatkan memeriksa apakah data memenuhi aturan atau kriteria tertentu. Misalnya, memeriksa apakah data numerik berada dalam rentang yang valid atau memverifikasi keabsahan format alamat email.

1. Normalisasi Data

Normalisasi data melibatkan mengubah data ke dalam bentuk yang standar atau normal. Misalnya, mengubah huruf besar menjadi huruf kecil, mengganti sinonim atau kata serupa menjadi bentuk yang seragam, atau menyederhanakan format data seperti penggabungan format telepon yang berbeda.

1. Pemfilteran Data

Proses pemfilteran data melibatkan menghilangkan atau menyaring data yang tidak relevan, duplikat, atau tidak diinginkan. Ini sering melibatkan penggunaan kondisi atau kriteria untuk memutuskan data mana yang akan diterima dan mana yang akan dihapus.

1. Penggabungan atau Penggantian Data

Proses ini melibatkan penggabungan data dari beberapa sumber atau penggantian nilai-nilai data dengan nilai yang baru atau direvisi. Misalnya, menggabungkan data dari beberapa file atau mengganti nilai khusus dengan nilai yang lebih bermakna.

1. Penyusutan atau Pengumpulan Data

Proses ini melibatkan pengurangan ukuran data atau penyusutan data menjadi format yang lebih efisien. Misalnya, menggabungkan atau mengelompokkan data menjadi kelompok yang lebih kecil atau lebih terorganisir.

1. Pembuatan Indeks atau Struktur Data Tambahan

Proses ini melibatkan pembuatan indeks atau struktur data tambahan untuk mempercepat akses atau pemrosesan data di kemudian hari. Misalnya, membuat indeks untuk pencarian cepat atau membangun struktur data yang lebih efisien.

## ALGORITMA MACHINE LEARNING PADA BAHASA GO

Walaupun bahasa Go semakin banyak digunakan, jumlah pustaka machine learningnya masih kurang dibandingkan dengan bahasa seperti Python atau R. Meskipun begitu, telah ada beberapa contoh algoritma machine learning yang berhasil diimplementasikan dalam bahasa Go, antara lain :

1. **Regresi Linear** : Algoritma regresi linear digunakan untuk memprediksi nilai numerik berdasarkan hubungan linier antara variabel input dan output.
2. **K-Means Clustering** : Algoritma k-means clustering digunakan untuk mengelompokkan data ke dalam k kelompok yang berbeda berdasarkan kesamaan atribut.
3. **Naive Bayes Classifier** : Algoritma Naive Bayes digunakan untuk klasifikasi teks atau data berlabel berdasarkan probabilitas dari fitur-fitur yang ada.
4. **Decision Tree** : Algoritma decision tree digunakan untuk membuat model prediksi berbasis pohon keputusan, yang dapat digunakan untuk klasifikasi atau regresi.
5. **K-Nearest Neighbors (KNN)** : Algoritma KNN digunakan untuk klasifikasi data berdasarkan kesamaan dengan tetangga terdekatnya.
6. **Principal Component Analysis (PCA)** : Algoritma PCA digunakan untuk mengurangi dimensi data dan membantu visualisasi data yang kompleks.
7. **Support Vector Machine (SVM)** : Algoritma SVM digunakan untuk klasifikasi data dengan mencari hyperplane yang optimal untuk memisahkan kelas data yang berbeda.
8. **Gradient Boosting** : Algoritma gradient boosting digunakan untuk menggabungkan beberapa model lemah menjadi model yang lebih kuat dan akurat.
9. **Neural Networks** : Beberapa pustaka dalam bahasa Go seperti Gorgonia dan Tensorflow Go memiliki implementasi untuk neural networks, yang digunakan untuk membangun model pembelajaran dalam jaringan saraf tiruan.
10. **Random Forest** : Algoritma random forest adalah algoritma ensemble yang menggabungkan beberapa pohon keputusan untuk klasifikasi atau regresi. Implementasi random forest dalam Go dapat membantu dalam pembuatan model yang lebih kuat dan presisi.

## KIAT MEMBANGUN MACHINE LEARNING MENGGUNAKAN GOLANG

Untuk membangun machine learning menggunakan bahasa Go, terlebih dahulu kita memerlukan pemahaman tentang bahasa tersebut dan konsep machine learning sendiri. Berikut ini adalah kiat-kiat untuk membangun machine learning menggunakan bahasa Golang :

* Memahami dasar-dasar bahasa Golang. Sebelum memulai proyek AI, pastikan Anda memahami dasar-dasar bahasa Golang seperti tipe data, struktur kontrol (if-else, switch), pemrograman berorientasi objek, penggunaan goroutines, dan channel.
* Memahami konsep machine learning. Sebelum mulai membangun model machine learning, penting untuk memahami konsep dasar seperti supervised learning, unsupervised learning dan algoritma-algoritma machine learning.
* Menggunakan library Go yang tepat. Ada beberapa pustaka ML yang tersedia di Golang. Contohnya, Anda dapat menggunakan Gorgonia untuk pembelajaran mesin dan komputasi numerik, GoLearn untuk pembelajaran mesin, dan GoCV untuk pengolahan citra. Pilih library yang sesuai dengan kebutuhan Anda dan pelajari dokumentasinya.
* Melatih dan mengevaluasi model. Sebagian esensial dalam machine learning adalah melakukan pelatihan model menggunakan data yang tersedia. Anda perlu memisahkan dataset menjadi dua bagian, yaitu data pelatihan dan data pengujian. Dengan menggunakan data pelatihan, Anda melatih model dan kemudian mengukur kinerjanya dengan data pengujian.
* Optimalisasi dan setting parameter. Setelah berhasil memiliki model yang berfungsi, terdapat peluang untuk melakukan setting parameter guna meningkatkan kinerjanya. Anda dapat mencoba variasi parameter seperti learning rate, jumlah iterasi, atau jumlah neuron dalam jaringan saraf, dan mengamati bagaimana hal tersebut mempengaruhi kinerja model.
* Menggunakan model pada data baru. Setelah model telah disiapkan, Anda dapat mengaplikasikannya untuk memprediksi atau mengklasifikasikan data baru. Pastikan model tersebut dapat dengan baik mengatasi data yang belum pernah dilihat sebelumnya.
* Memperbaharui Model. Machine learning merupakan proses yang berlangsung terus-menerus. Teruslah melakukan pembaruan dan peningkatan pada model Anda seiring dengan munculnya data baru atau sebagai respons terhadap perubahan kebutuhan bisnis.

# DAFTAR PUSTAKA

Agung, Noval. Dasar Pemrograman Golang. Diakses pada 15 Juli 2023 dari <https://dasarpemrogramangolang.novalagung.com/A-hello-world.html.>

Bahasa Golang dalam membangun Machine Learning. Diakses pada 21 Juli 2023 dari <https://technobrains.io/golang-best-language-for-ai-development/>

Bekerja dengan file xml di go. Diakses pada 20 Juli 2023 dari https://www.developer.com/languages/xml-files-golang/

Fungsi dari Golang. Diakses pada 16 Juli 2023 dari <https://www.binaracademy.com/blog/apa-itu-golang-dan-fungsinya>

How To Use JSON in Go - Digital Ocean. Diakses pada 20 Juli 2023 dari <https://www.digitalocean.com/community/tutorials/how-to-use-json-in-go>

Installasi Golang. Diakses pada 18 Juli 2023 dari <https://www.tutorialpemrograman.com/others/cara-install-golang-di-windows-10/>

JSON - Go by Example. Diakses pada 20 Juli 2023 dari https://gobyexample.com/json

Keunggulan Golang. Diakses pada 15 Juli 2023 dari <https://www.gamelab.id/news/1908-kenalan-dengan-golang-bahasa-pemrograman-yang-dikembangkan-oleh-google>

Konsep Dasar Machine Learning. Diakses pada 21 Juli 2023 dari <https://medium.com/@ph.harmony1/konsep-dasar-dalam-machine-learning-8ceb22157ef2>

Langkah-langkah installasi Golang. Diakses pada 15 Juli 2023 dari https://go.dev/doc/

Library golang untuk membangun machine learning. Diakses pada 21 Juli 2023 dari https://golangdocs.com/golang-machine-learning-libraries

Mengenal Tipe Data pada Golang. Diakses pada 19 Juli 2023 dari <https://www.santekno.com/mengenal-tipe-data-pada-pemrograman-golang/>

Pengenalan Golang. Diakses pada 15 Juli 2023 dari <https://www.sekawanmedia.co.id/blog/belajar-golang/>

Pengolahan Data pada Golang. Diakses pada 21 Juli 2023 dari <https://www.tutorialspoint.com/golang-program-to-write-into-a-file>

Preprocessing dalam Data Mining. Diakses pada 21 Juli 2023 dari <https://www.trivusi.web.id/2022/09/data-preprocessing.html>

String Manipulasi menggunakan Golang. Diakses pada 21 Juli 2023 dari <https://eternaldev.com/blog/5-ways-to-concatenate-string-in-golang-with-reasons>

Tipe Data Float pada Golang. Diakses pada 19 Juli 2023 dari <https://medium.com/clean-code-62/golang-bagian-4-e148cb3614f3>