

Penggunaan Golang dan *Framework* Gin untuk Aplikasi Web Steganografi LSB

Felix Limanta

Sekolah Tinggi Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
felixlimanta@gmail.com

Abstract—Bahasa pemrograman Go, atau Golang, adalah bahasa pemrograman *open-source* yang dikembangkan oleh Google. Dengan pengembangannya yang cepat dan resepsi yang baik dari komunitas pengembang web di seluruh dunia, bahasa pemrograman ini patut dicoba. Dasar-dasar penggunaan Golang serta contoh implementasinya dalam sebuah aplikasi web untuk steganografi LSB didokumentasikan pada makalah ini. Hasilnya menunjukkan bahwa Golang dapat dan baik untuk digunakan dalam pengembangan aplikasi web.

Keywords—Golang, Gin, LSB, steganografi

I. PENDAHULUAN

Sebuah bahasa pemrograman yang baik dibutuhkan untuk mengembangkan berbagai jenis perangkat lunak, seperti aplikasi web. Terkadang, orang lebih menyukai menggunakan aplikasi web dibandingkan aplikasi *desktop*. Aplikasi web tidak perlu di-*install* dan diperbarui, serta dapat digunakan selama koneksi Internet tersedia. Pengembangan aplikasi web dapat menjadi rumit karena sebuah aplikasi web akan diakses oleh berbagai pengguna dan pengembangan harus memikirkan aspek keamanan dan konkurensi.

Google, salah satu perusahaan teknologi terbesar di dunia, memutuskan untuk mengembangkan sebuah bahasa pemrograman baru pada 2007 silam. Bahasa ini dibuat agar memiliki model pengembangan yang terbuka, sehingga siapapun dapat berkontribusi dalam pengembangannya lebih jauh. Golang adalah bahasa mirip C, tetapi lebih kompleks. Golang tidak memaksakan paradigma prosedural seperti C atau paradigma pemrograman berorientasi objek seperti Java, tetapi memberikan kebebasan pada pengembang untuk memilih paradigma pemrograman yang ia sukai.

Karena pembuat Golang, Google, adalah salah satu perusahaan teknologi termapan, Golang dipercaya oleh banyak pengembang. Meskipun demikian, ini tidak secara otomatis membuat Golang bahasa pemrograman yang baik.

Alasan mengapa penulis memutuskan untuk mengeksplorasi bahasa ini adalah popularitasnya yang semakin berkembang dan ulasannya yang memujinya sebagai bahasa yang baik. Banyak orang telah mengembangkan aplikasi mereka dalam Golang. Makalah ini akan mendiskusikan proses pembelajaran menggunakan Golang dan pengembangan sebuah aplikasi steganografi LSB sederhana menggunakan Golang.

II. KONSEP DASAR GOLANG DAN *FRAMEWORK* GIN

Sebelum membuat sebuah aplikasi dalam Golang, sintaks dasar dan struktur program Golang sebaiknya dipelajari terlebih dahulu. Bagian berikut mendiskusikan secara singkat sintaks dasar dan struktur program Golang, serta cara kerja dasar *framework* Gin.

Sebagian besar hal yang didiskusikan di bab ini diambil dari pemahaman penulis mengenai tutorial yang disediakan di [1].

A. Struktur Program Golang

Sebagai bahasa turunan C, Golang memiliki struktur program yang hampir sama dengan bahasa turunan C lainnya (C, C++, Java, dll.), tetapi memiliki sintaks yang cukup berbeda. Sebuah program Golang terdiri atas minimal satu *package*.

```
package main
import (
    "fmt"
)

func main() {
    fmt.Println("Hello, World!")
}
```

Fig. 1. Contoh program “Hello World” dalam Golang

Sebuah program Golang harus mengandung setidaknya *package* “main”. Program tersebut dapat mengimpor *package* lain dan harus memiliki fungsi *main*. Sebuah fungsi dideklarasikan dengan kata kunci *func*.

B. Deklarasi Fungsi dan Prosedur

```
func multiply(x int, y int) int {  
    return x * y  
}  
  
func swap(x, y int) (int, int) {  
    return y, x  
}  
  
func split(sum int) (x, y int) {  
    x = sum / 10  
    y = sum % 10  
    return  
}
```

Fig. 2. Contoh deklarasi fungsi dalam Golang

Potongan kode di atas menunjukkan beberapa cara berbeda untuk mendeklarasikan fungsi.

- Fungsi pertama adalah deklarasi fungsi standar C di mana setiap parameter masukan memiliki sebuah tipe. Jika tipe semua parameter sama, hanya parameter terakhir yang perlu ditulis tipenya.
- Fungsi kedua menunjukkan kemampuan fungsi Golang untuk mengembalikan lebih dari satu variabel secara bersamaan, mirip dengan Python.
- Fungsi ketiga menunjukkan bahwa parameter keluaran dapat diberi nama variabel, sehingga perintah return di akhir fungsi akan secara implisit mengembalikan variabel yang bernama sama dengan variabel parameter keluaran.

C. Deklarasi Tipe dan Variabel

Tipe-tipe dasar yang umum digunakan adalah, di antaranya, bool, string, int, uint, byte, rune, float64, dan complex64. Terdapat beberapa cara untuk mendeklarasikan variabel pada Golang seperti berikut.

```
var a  
var b bool  
var c int = 1  
d := 1.5  
e := int(d)  
const Pi = 3.14
```

Fig. 3. Contoh deklarasi variabel dalam Golang

Kata kunci `var` digunakan untuk mendeklarasikan variabel, baik global maupun lokal. Jika tipe variabel tidak dituliskan, variabel tersebut akan diberi tipe saat di-assign sebuah nilai. Jika variabel memiliki tipe, tetapi tidak memiliki nilai awal, variabel tersebut akan diberi nilai awal berdasarkan tipenya.

Operator `:=` merupakan cara ringkas untuk mendeklarasikan variabel tanpa menggunakan kata kunci `var` dan tanpa mendeklarasikan tipenya. Operator ini dapat digunakan bersamaan dengan *type casting*, yang dilakukan dengan sintaks `type(variable)`. Deklarasi konstanta dilakukan dengan kata kunci `const`.

D. Analisis Kasus

Analisis kasus dilakukan dengan pernyataan `if-else` dan `switch`. Perilaku mereka sama dengan analisis kasus dalam C atau C++, serta *scope* variabel yang dideklarasikan di dalam blok kode tersebut hanya berlaku di dalamnya.

Satu hal yang berbeda pada `if-else` pada Golang adalah, pada Golang, pernyataan `if` dapat diawali dengan sebuah baris eksekusi program (mirip dengan klausa pertama pada `for`). C, C++, dan bahasa turunan C lainnya hanya memperbolehkan ekspresi *boolean* pada pernyataan `if`.

```
if v := math.Sqrt(x, n); v < lim {  
    return v  
} else {  
    return lim  
}
```

Fig. 4. Contoh analisis kasus dalam Golang

E. Pengulangan

Golang hanya mempunyai satu jenis pengulangan: `for`. Baik pengulangan `while-do` maupun `do-while` dilakukan menggunakan `for`.

```
for i := 1; i < 10; i++ {  
    fac *= i  
}  
  
for i < 10 {  
    fac += i  
    i++  
}
```

Fig. 5. Contoh pengulangan dalam Golang

F. Tipe Bentuk

```
type Point struct {  
    X int  
    Y int  
}  
  
P := Point(3, 4)  
fmt.Println(P.X)
```

Fig. 6. Contoh pembentukan tipe dalam Golang

Potongan kode di atas adalah cara mendeklarasikan tipe bentuk baru. Konstruksi tipe baru dilakukan dengan memasukkan semua anggota tipe ke dalam parameter. Pengaksesan anggota tipe dilakukan dengan operator `.` (titik).

G. Pointer

Golang memiliki *pointer*, tetapi tidak memiliki aritmetika *pointer*. Sama dengan C/C++, operator `&` mengembalikan alamat dari sebuah variabel, sedangkan operator `*` mengembalikan nilai dari alamat yang ditunjuk (*dereferencing*).

Golang tidak memiliki operator `->`. Pengaksesan anggota tipe bentuk dari *pointer* digunakan dengan operator `.`, sama seperti jika anggota tersebut diakses dari variabel pada *stack*.

H. Larik dan Slice

```
var T [10]int
var arr []int
a := make([]int, 5)
capacity := cap(T)
length := len(T)
```

Fig. 7. Contoh larik dan slice dalam Golang

Pernyataan pertama merupakan cara deklarasi larik dengan tipe tertentu dan ukuran tertentu. Pernyataan kedua dan ketiga merupakan cara deklarasi slice, yaitu larik bertipe tertentu dengan ukuran variabel. Fungsi `cap(T)` digunakan untuk menghitung kapasitas larik atau slice, sedangkan `len(T)` untuk menghitung panjang sebenarnya. Nilai `nil` adalah slice dengan kapasitas dan panjang nol.

I. Framework Gin

Salah satu fitur terbaik Golang adalah kakas `net/http` bawaannya, yang memungkinkan pembuatan server HTTP dengan mudah. Meskipun demikian, kakas ini tidak terlalu fleksibel dan membutuhkan sejumlah kode *boilerplate*.

Framework Gin merupakan sebuah *framework* web untuk Golang. Gin menawarkan sejumlah fungsionalitas yang umum digunakan, seperti *routing*, dukungan *middleware*, *rendering*, dll. yang mengurangi jumlah kode *boilerplate* dan mempermudah penulisan aplikasi web.

```
import (
    "github.com/gin-gonic/gin"
)

func main() {
    r := gin.Default()

    r.GET("/", func(c *gin.Context) {
        c.File("index.html")
    })

    r.POST("/upload", func(c *gin.Context) {
        ...
    })

    r.Run(":8000")
}
```

Fig. 8. Contoh aplikasi server sederhana menggunakan *framework* Gin

Deklarasi rute yang terdapat pada aplikasi, operasi HTTP yang diperbolehkan, halaman yang ditampilkan, serta aksi yang dilakukan pada rute tersebut ditulis seperti di atas.

III. KONSEP DASAR LAINNYA

A. Steganografi LSB

Steganografi adalah penyembunyian pesan tersembunyi pada sebuah pesan biasa dan mengekstraksinya ketika pesan tersebut sampai di tujuan. Steganografi merupakan bidang terkait dari kriptografi, di mana pesan yang telah dienkripsi disembunyikan, sehingga tidak ada yang tahu bahwa ada pesan

tersembunyi. Secara ideal, semua orang yang melihat data yang ditanami tidak tahu bahwa data tersebut mengandung pesan tersembunyi.

Untuk data multimedia, cara paling umum dan mudah untuk menanamkan pesan pada suatu data adalah substitusi LSB (*least significant bit*, bit paling tidak signifikan). Pada substitusi LSB, bit terkecil (paling tidak signifikan) dari setiap *byte* pada data yang ditanami diganti oleh data yang akan ditanamkan, bit per bit. Dengan hanya mengubah bit terkecil dari setiap *byte*, perubahan dari data semula yang belum ditanami hampir tidak dapat dipersepsi oleh indra manusia.

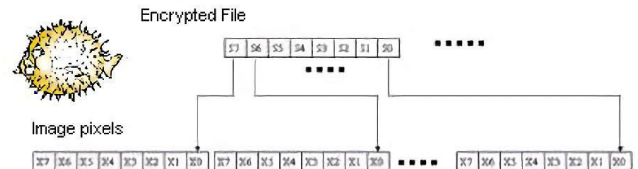


Fig. 9. Ilustrasi steganografi LSB

Sumber: codeproject.com

Meskipun sederhana, substitusi LSB dianggap tidak kokoh (*fragile*). Meskipun transformasi seperti pemotongan gambar dapat ditangani, penambahan *noise* yang tidak dikehendaki atau kompresi *lossy* dapat menghancurkan pesan yang ditanamkan.

B. Format Bitmap

Bitmap (BMP) adalah representasi citra grafis yang terdiri atas susunan titik yang tersimpan di memori komputer. BMP dikembangkan oleh Microsoft, yang memiliki kontrol penuh pada format tersebut, sehingga format BMP stabil sejak versi yang dirilis dengan Windows 3.0 (versi 3).

Bitmap adalah representasi dari citra grafis yang terdiri dari susunan titik yang tersimpan di memori komputer. Dikembangkan oleh Microsoft dan nilai setiap titik diawali oleh satu bit data untuk gambar hitam putih, atau lebih bagi gambar berwarna.

Kapabilitas format BMP dapat dilihat dari kapabilitas aplikasi Paint (aplikasi pengubah citra bawaan Windows). BMP versi 5.1 (yang datang bersama Windows XP) mendukung warna 1-bit, 4-bit, 8-bit, dan 24-bit. BMP mendukung kompresi sederhana, tetapi kompresi tersebut tidak pernah digunakan karena format PNG atau JPG jauh lebih baik untuk gambar terkompresi *lossless* atau *lossy*.

Struktur dasar BMP adalah berkas biner (bukan teks) dan dapat dibagi menjadi beberapa bagian seperti berikut.

- *Header* berkas (14 *byte*)
 - Berisi ukuran berkas, awal data citra
- *Header* citra (40 *byte*)
 - Berisi detail seperti dimensi citra, format yang digunakan (bits per pixel), kompresi, dll.
- Tabel warna (panjang variabel, tidak selalu ada)
 - Berisi palet warna (8-bit)
- Data piksel
 - Data piksel diurutkan baris per baris, kiri ke kanan

- Untuk citra 24-bit, kanal warna diurutkan sebagai *Red, Green, Blue*

Format BMP dipilih karena kemudahannya dimanipulasi piksel per pikselnya. Setiap piksel diurutkan secara teratur, sehingga pembacaan dan penyimpanan data setiap piksel langsung dilakukan dengan mengakses *byte* yang relevan. Bagian yang relevan pada masalah steganografi LSB hanya data piksel. *Header* berkas dan *header* citra diabaikan pada masalah ini.

IV. IMPLEMENTASI¹

A. Implementasi Algoritma Steganografi LSB

Algoritma steganografi LSB diimplementasikan dalam tiga fungsi sebagai berikut.

```
func encodeMessage(pic *bytes.Buffer, message
string) (bytes.Buffer, error) { ... }

func decodeMessage(pic *bytes.Buffer) string
{ ... }

func setLSB(b byte, val byte) byte { ... }
```

Fig. 10. *Header* fungsi-fungsi yang digunakan untuk implementasi algoritma substitusi LSB

Fungsi `encodeMessage` menerima parameter *buffer* dari larik *byte* citra pesan yang akan disisipi dan *string* pesan yang akan disisipkan, lalu mengembalikan *buffer* dari larik *byte* citra pesan yang sudah disisipi pesan, serta pesan kesalahan jika ada. Larik *byte* tidak dapat diterima dan dikembalikan secara mentah, sehingga harus dibungkus dalam sebuah *buffer* terlebih dahulu. Fungsi akan gagal jika jumlah bit pesan lebih besar dari jumlah *byte* pada citra, dikurangi 54 *byte* dari *header*. Jika pesan dapat disisipkan, setiap bit pesan akan disisipkan secara sekuensial pada setiap *channel* dari setiap *pixel* citra, secara berurutan. Ketika pesan selesai, akan disisipkan 8 bit 0 (0x00, *string terminator*) sebagai tanda bahwa pesan telah berakhir.

Fungsi `decodeMessage` menerima parameter *buffer* dari larik *byte* citra pesan yang telah disisipi. Fungsi akan menelusuri semua *byte* untuk bit terkecilnya, di mana setiap delapan bit akan digabung menjadi satu *byte* dan ditambahkan ke *string* pesan yang telah diekstrak. Fungsi berakhir ketika *byte* yang terekstrak bernilai 0.

Fungsi `setLSB` merupakan fungsi pembantu untuk mengubah nilai bit terkecil dari *byte* masukan.

B. Desain dan Implementasi Backend

Seluruh fitur *backend* berada pada fungsi *main*, di mana sebuah instansi Gin dibuat. Rute API yang dapat diterima *backend* adalah sebagai berikut.

Route	Metode	Deskripsi
/	GET	Mengembalikan laman utama <i>frontend</i>
/encode	POST	Menerima <i>form</i> berisi citra dan pesan, lalu mengembalikan citra hasil penyisipan
/decode	POST	Menerima <i>form</i> berisi citra hasil penyisipan, lalu mengembalikan <i>string</i> pesan yang disisipi

Rute *encode* menerima citra dan pesan, memanggil fungsi `encodeMessage`, lalu mengembalikan *response* berupa representasi base64 citra yang telah disisipkan. Rute *decode* menerima citra yang telah disisipkan, memanggil fungsi `decodeMessage`, lalu mengembalikan *response* berupa *string* pesan yang telah disisipkan.

C. Desain dan Implementasi Frontend

Frontend dibangun menggunakan HTML dan JavaScript. Desain CSS menggunakan kakas Bootstrap, sedangkan JavaScript diperkaya oleh kakas jQuery.

Encode

Choose File _test.bmp

Message:

Hello, World

Submit



Decode

Choose File download.dib

Submit

Message:

Hello, World

TABLE I. DAFTAR API YANG TERSEDIA

Fig. 11. Cuplikan layar *frontend*

¹ Kode program dapat diakses melalui pranala <https://github.com/felixlimanta/golang-steganography-lsb-bmp>

Frontend terdiri dari satu halaman HTML serta kode JavaScript. Bagian *encoding* dan *decoding* berada pada satu halaman yang sama, tetapi berada pada *form* yang berbeda.

Untuk meng-*encode* (menyisipkan pesan), pengguna mengunggah citra *bitmap* (BMP) dan mengetikkan pesan pada kotak teks yang tersedia. Citra dan pesan dikirim secara AJAX dan citra yang telah disisipkan ditampilkan di bawah tombol *submit*. Pengguna dapat menyimpan citra hasil penyisipan yang ditampilkan.

Untuk men-*decode* citra yang disisipi pesan, pengguna cukup mengunggah citra *bitmap* yang telah disisipi pesan. Citra dikirim secara AJAX dan pesan yang disisipkan pada citra akan ditampilkan pada kotak teks di bawah tombol *submit*.

Jika terjadi kegagalan, halaman akan mengeluarkan *alert* yang berisi pesan kegagalan.

V. PENGUJIAN DAN PEMBAHASAN

Pengujian dilakukan dengan mengubah dimensi citra, mengubah jumlah kanal warna pada citra, dan mengubah ukuran pesan yang disisipkan. Pengujian dianggap berhasil ketika pesan yang disisipkan dapat diekstrak dengan utuh.

Pesan hasil pengujian merupakan pesan *lorem ipsum* yang dibangkitkan sesuai jumlah *byte* yang dibutuhkan². Citra uji yang digunakan dikonversi terlebih dahulu ke format *bitmap*, lalu diubah dimensi dan dikonversi ke citra *greyscale* sesuai kebutuhan.



Fig. 12. Citra yang digunakan dalam pengujian
Sumber: [pixiv.net](https://www.pixiv.net)

Hasil pengujian dirangkum pada dua tabel berikut.

TABLE II. HASIL PENGUJIAN UNTUK CITRA BERWARNA 24-BIT

Ukuran Teks (byte)	Dimensi Citra (n x n piksel)				
	64	128	256	512	1000
64	V	V	V	V	V
4096	X	V	V	V	V
65536	X	X	X	V	V

TABLE III. HASIL PENGUJIAN UNTUK CITRA GREYSCALE

Ukuran Teks (byte)	Dimensi Citra (n x n piksel)				
	64	128	256	512	1000
64	V	V	V	V	V
4096	X	X	V	V	V
65536	X	X	X	X	V

Dari hasil pengujian di atas, dapat dilihat bahwa implementasi steganografi LSB sudah benar. Jumlah pesan yang dapat disisipkan pada suatu citra dapat dirumuskan sebagai berikut.

$$\text{byte pesan} \times 8 - 1 \leq \text{dimensi gambar}^2 \times \text{jumlah kanal} \quad (1)$$

VI. ANALISIS PENGEMBANGAN DENGAN GOLANG

Sebagai sebuah bahasa pemrograman, Golang dapat digunakan untuk keperluan umum, seperti implementasi steganografi LSB pada kasus ini. Meskipun demikian, kekuatan Golang dapat dilihat ketika membangun sebuah aplikasi web. Ini dapat dilihat dari kakas bawaan Golang, yang menyediakan fitur untuk menjadi *server*, sehingga tidak membutuhkan aplikasi *server* terpisah (Apache untuk PHP, Tomcat untuk Java, dsb.). Dibandingkan dengan bahasa lain yang menyediakan fitur serupa (NodeJS, misal), Golang menyediakan sintaks yang ringkas dan pengecekan tipe yang ketat (*strict typing*).

Meskipun demikian, Golang tergolong relatif muda, sehingga lingkungan pengembangan Golang masih tidak sedewasa lingkungan pengembangan bahasa lain. Kakas yang tersedia untuk Golang belum terlalu banyak, sehingga banyak fungsionalitas Golang yang harus dibuat dari awal. Belum terlalu banyak pula pakar Golang, sehingga mendapat bantuan untuk masalah Golang relatif lebih sulit.

Pengembangan aplikasi web yang mudah menggunakan Golang dibuat lebih mudah lagi dengan *framework* Gin. Tidak seperti *framework* pada umumnya, Gin tidak perlu struktur program khusus (seperti halnya Laravel atau *framework* lainnya); Gin hanya membutuhkan *import* saja. Penggunaan Gin mempermudah beberapa fungsionalitas server, seperti *routing* dan *middleware*.

Pengembang yang biasa menulis kode C, C++, atau Java akan sedikit kesulitan pada awalnya mengenai sintaks, tetapi kemiripannya dengan bahasa-bahasa tersebut serta paradigmanya yang masih prosedural membuat pembelajaran Golang tidak sesulit yang dibayangkan.

Ditambah lagi, Golang merupakan salah satu bahasa yang dibangun untuk konkurensi. Menurut [2], konkurensi pada Golang terjadi dengan kecepatan eksekusi statis setara C atau C++. Ini tidak mungkin pada bahasa *backend* lain.

² Pesan *lorem ipsum* dibangkitkan pada situs <https://www.lipsum.com/>

VII. KESIMPULAN

Golang bukan merupakan bahasa yang mudah untuk dipelajari, tetapi sekali dikuasai, pengembangan aplikasi web dengan Golang akan lebih mudah dibandingkan dengan bahasa *backend* lainnya, seperti PHP atau Java, terutama untuk pengembangan *server*. Akhir kata, Golang sulit dipelajari, tetapi patut dicoba.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih pada dosen mata kuliah IF3280 Socio-Informatika dan Profesionalisme: Ir. Windy Gambetta, ST., Dr. Eng. Ayu Purwarianti, ST., MT. and Dr. Dessi Puji Lestari, yang memacu penulis untuk terus belajar hal baru dan tidak bersantai-santai dengan pengetahuan yang sudah dimiliki. Ini memberi penulis kesempatan untuk penulis belajar Golang, yang akan digunakan penulis pada masa kerja praktiknya.

REFERENSI

- [1] "A Tour of Go," *A Tour of Go*. [Daring]. Tersedia: <https://tour.golang.org/list>. [Diakses: 01-Mei-2018].
- [2] Mariana, "9 Reasons to Choose Golang for your Next Web Application," *Letzgro*, 18-Jul-2016. [Daring]. Tersedia: <http://letzgro.net/blog/9-reasons-to-choose-golang-for-your-next-web-application/>. [Diakses: 01-Mei-2018].

- [3] K. Kabra, "Building Go Web Applications and Microservices Using Gin," *Semaphore*, 29-Jun-2017. [Daring]. Tersedia: <https://semaphoreci.com/community/tutorials/building-go-web-applications-and-microservices-using-gin>. [Diakses: 01-Mei-2018].
- [4] R. Munir, Diktat Kuliash IF5054 Kriptografi, Bandung: Departemen Teknik Informatika Institut Teknologi Bandung, 2005.
- [5] W. L. Bahn, "Simplified Windows BMP Bitmap File Format Specification," *DragonWins*, 27-Nov-2010. [Daring]. Tersedia: <http://www.dragonwins.com/domains/getteched/bmp/bmpfileformat.htm>. [Diakses: 01-Mei-2018].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 1 Mei 2018



Felix Limanta 13515065