**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : Gregory Sukanto**

**NRP : 5110100162**

**DOSEN WALI : Ir. Muchammad Husni, M.Kom.**

**DOSEN PEMBIMBING : 1. Tohari Ahmad, S.Kom., MIT., Ph.D.  
 2. Hutan Studiawan, S.Kom., M.Kom.**

# JUDUL TUGAS AKHIR

“Pengembangan Metode Steganografi Menggunakan Emoticon Berbasis LZW pada PHP Free Chat”

# LATAR BELAKANG

Pada jaman informasi ini keamanan data adalah hal yang sangat penting dan mutlak untuk dimiliki. Keamanan data menjadi sesuatu yang sangat penting terutama jika pesan yang dikirimkan bersifat sangat rahasia seperti dokumen militer, password, informasi sidik jari, dan sebagainya. Berbagai usaha dilakukan untuk memembuat agar pesan rahasia yang dikirimkan tersebut tidak dapat diketahui oleh pihak yang tidak berwenang salah satu adalah dengan menciptakan metode untuk menyembunyikan pesan rahasia. Metode yang cukup umum digunakan untuk menyembunyikan pesan rahasia adalah kriptografi yaitu teknik menyamarkan arti dari suatu pesan, misalnya dengan mengacak atau menyandikan pesan tersebut, sehingga bila pesan tersebut diakses oleh pihak yang tidak berwenang maka diharapkan pihak tersebut tidak mendapatkan informasi pesan yang dirahasiakan. Akan tetapi, kriptografi tidak menyembunyikan bahwa ada suatu pesan karena sifatnya yang acak yang menyebabkan tidak dapat dimengerti oleh manusia sehingga menimbulkan kecurigaan. Untuk itulah steganografi dibutuhkan. Banyak metode steganografi yang ada, salah satunya adalah menggunakan *emotion* *icon* (*emoticon*) untuk menyembunyikan pesan. Akan tetapi jumlah *emoticon* yang digunakan terbatas dan pesan yang digunakan bisa jadi sangat panjang. Oleh karena itu penulis mengusulkan sebuah metode baru untuk meningkatkan jumlah pesan yang dapat disimpan oleh *emoticon*.

Menyisipkan pesan rahasia ke dalam teks tertulis adalah tugas yang menantang karena memerlukan teks yang tidak terlalu banyak pengulangan untuk menyisipkan pesan rahasia [1]. Secara umum steganografi dapat dikelompokkan menjadi substitusi sinonim, transformasi sintaksis, kesalahan penulisan, dan transformasi semantik [1]. Steganografi dengan *emoticon* diusulkan oleh Wang et al. [1] yaitu dengan mengubah *bit* dari pesan rahasia menjadi angka desimal kemudian mengambil *emoticon* sesuai angka dari daftar *emoticon* yang telah ditentukan. Penulis mengusulkan metode steganografi dengan membuat kamus *emoticon* yang dinamis dengan menggunakan algoritma LZW.

Hasil yang diharapkan penulis adalah membuat metode baru untuk steganografi dengan *emoticon* sehingga lebih sulit untuk dipecahkan / lebih aman. Penulis juga mengharapkan agar metode baru untuk steganografi dengan *emoticon* ini dapat menambah panjang *bit* dari pesan rahasia yang akan disisipkan pada *emoticon*.

# RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana cara menambah ukuran *bit* yang dapat disimpan pada *emoticon*?
2. Bagaimana cara mengimplementasi penyisipan pesan pada *emoticon*?
3. Bagaimana cara mengimplementasi pengekstrak pesan pada *emoticon*?
4. Bagaimana cara mengimplementasi steganografi pada PHP *Free Chat*?

# BATASAN MASALAH

Beberapa batasan masalah yang menjadi batas pada Tugas Akhir ini adalah:

1. Metode yang diterapkan adalah metode LZW.
2. Jumlah *emoticon* yang digunakan lebih dari 32.
3. Bahasa pemrograman yang digunakan adalah PHP dan python.
4. Teks yang digunakan untuk menyembunyikan steganografi *emoticon* menggunakan bahasa Indonesia.
5. Aplikasi ini menggunakan pustaka PHP *Free Chat.*
6. Perintah dasar pada aplikasi *chat* menggunakan bahasa Inggris.

# TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari pembuatan Tugas Akhir ini adalah:

1. Pengembangan metode steganografi dengan menggunakan *emoticon* berbasis LZW.
2. Pengembangan metode steganografi pada pustaka PHP *Free Chat.*
3. Mengevaluasi kinerja dari metode steganografi dengan *emoticon* berbasis LZW pada PHP *Free Chat*.

# MANFAAT TUGAS AKHIR

Hasil dari Tugas Akhir ini diharapkan dapat memberi manfaat pada bidang teknologi informasi dalam bidang keamanan data dengan cara mengembangkan sebuah metode steganografi baru dengan menggunakan *emoticon* dan LZW.

# TINJAUAN PUSTAKA

## 8.1. Steganografi

Steganografi adalah ilmu menulis pesan rahasia dengan menyembunyikan pesan mengunakan suatu cara sehingga selain pihak yang berwenang tidak ada pihak yang mengetahui atau menyadari bahwa ada suatu pesan rahasia. Steganografi berasal dari bahasa Yunani *steganos* yang artinya tersembunyi atau terselubung dan *graphein* yang artinya menulis [2].

Tujuan utama dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan rahasia dengan menggunakan kemampuan persepsi manusia sehingga tidak ada pihak tidak berwenang yang menyadari ada suatu pesan rahasia. Steganografi dapat diklasifikasikan menjadi substitusi sinonim, transformasi sintaksis, kesalahan penulisan, dan transformasi semantik [1]. Kelebihan steganografi jika dibandingkan dengan kriptografi adalah keluaran dari steganografi tidak bersifat acak sehingga tidak ada pihak yang menyadari keberadaan pesan rahasia. Keluaran dari kriptografi yang bersifat acak walaupun tidak dapat dipecahkan tetapi akan menimbulkan kecurigaan sehingga pesan dapat dimanipulasi oleh pihak yang tidak berwenang.

## 8.2. *Emoticon*

Menurut Ptaszynski et al [10] *emoticon* merupakan simbol yang menunjukkan bahasa tubuh kertika berkomunikasi melalui pesan tulisan. Keterbatasan dalam menyampaikan ekspresi muka, nada, dan gesture badan dalam pesan tulisan dapat diatasi dengan menggunakan *emoticon*. Salah satu hal yang membuat pesan tulisan berbeda dengan percakapan tatap muka adalah tidak adanya bahasa tubuh / kondisi emosi. Oleh karena itulah, *emoticon* menjembatani pemisah antara pesan tulisan dengan percakapan tatap muka dengan memberikan gambaran kepada pembaca tentang kondisi emosi penulis dengan menampilkan gambar yang sesuai.

## 8.3. Kompresi Data

Kompresi data adalah sebuah cara untuk memadatkan data sehingga ukuran penyimpanan lebih kecil dan efisien. Tujuan proses kompresi data adalah untuk mengurangi redundansi dari sebuah deskripsi data [2]. Biasanya kompresi data digunakan untuk menghemat penggunaan media penyimpanan ataupun mengurangi ukuran data yang dikirim. Pada umumnya, algoritma kompresi data dibedakan menjadi *lossless* atau *lossy*. Kompresi data *lossless* adalah jenis kompresi ketika data awal sebelum dikompresi identik dengan data akhir setelah didekompresi. Kompresi *lossless* biasanya digunakan saat berkas asli dan berkas hasil dekompresi harus sama persis (seperti kompresi teks, berkas yang dapat dijalankan, dll). Kompresi data *lossy* adalah jenis kompresi ketika data awal sebelum dikompresi berbeda dengan data akhir setelah didekompresi. Ukuran data dengan kompresi *lossy* lebih kecil / lebih bagus dari pada kompresi *lossless* akan tetapi data yang didapat tidak sebagus data awal. Kompresi *lossy* biasanya digunakan pada media yang berhubungan dengan persepsi manusia seperti suara, gambar, video, *streaming*, dll.

## 8.4. Algoritma LZW

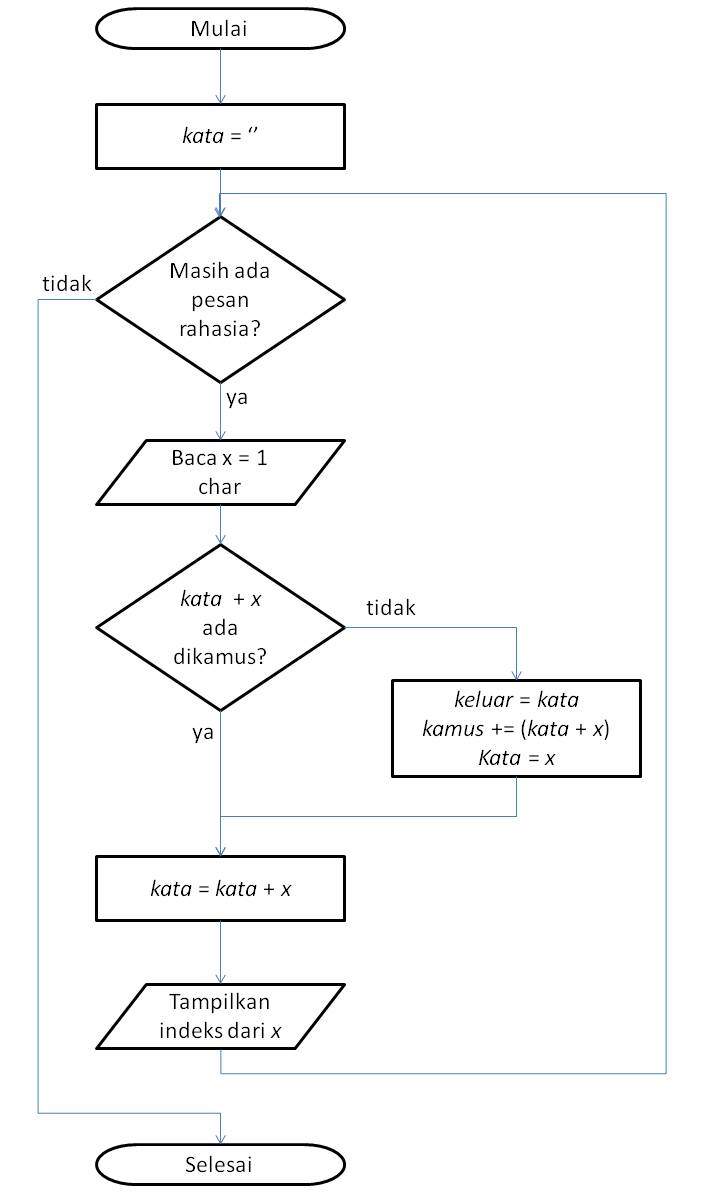
Algoritma LZW merupakan algoritma kompresi yang bersifat *lossless* dan menggunakan metode kamus yang dinamis. Algoritma ini ditemukan oleh Lemple, Ziv, dan Welch pada tahun 1984 [3]. Secara umum algoritma kompresi LZW akan membentuk kamus selama proses kompresinya belangsung kemudian setelah selesai maka kamus tersebut tidak ikut disimpan dalam berkas yang telah dikompresi.

*Pseudocode* dari algoritma *encode* dari LZW dapat dilihat pada Gambar 1 di bawah ini [5].

|  |  |
| --- | --- |
| Algoritma *encode* LZW | |
| 1: | *kata* 🡨 ‘’ |
| 2: | **while** bukan **EOF** lakukan |
| 3: | *x* 🡨 membaca\_karakter\_berikutnya() |
| 4: | **if** *kata* + *x* ada di kamus **then** |
| 5: | *kata* 🡨 *kata* + *x* |
| 6: | **else** |
| 7: | Keluaran indeks kamus dari kata |
| 8: | Tambahkan *kata* + *x* ke kamus |
| 9: | *kata* 🡨 *x* |
| 10: | **end if** |
| 11: | **end while** |
| 12: | Keluaran indeks kamus dari kata |

Gambar 1. *Pseudocode* *encode* LZW.

Untuk diagram alir dari algoritma *encode* LZW dapat dilihat pada Gambar 2 di bawah ini.



Gambar 2. Diagram alir e*ncode* LZW.

Dapat dilihat pada Gambar 2 bahwa hasil akhir dari algoritma LZW ini berupa indeks kamus yang menunjukkan letak kata di kamus. Setiap kali ditemukan gabungan kata yang belum ada di kamus maka algoritma ini akan menambahkan kata tersebut ke dalam kamus sehingga kamus dalam LZW ini bersifat dinamis. Algoritma LZW ini membaca masukan berupa satu karakter. Jika karakter tersebut ternyata ada di kamus maka algoritma ini akan membaca satu karakter lagi dan menggabungkannya dengan karakter sebelumnya dan begitu seterusnya sampai gabungan karakter tersebut tidak ditemukan di kamus. Setelah menemukan kata yang tidak ada di kamus maka algoritma ini akan menampilkan indeks kamus dari kata dikurangi dengan satu karakter. Hal ini dilakukan terus menerus hingga masukan habis.

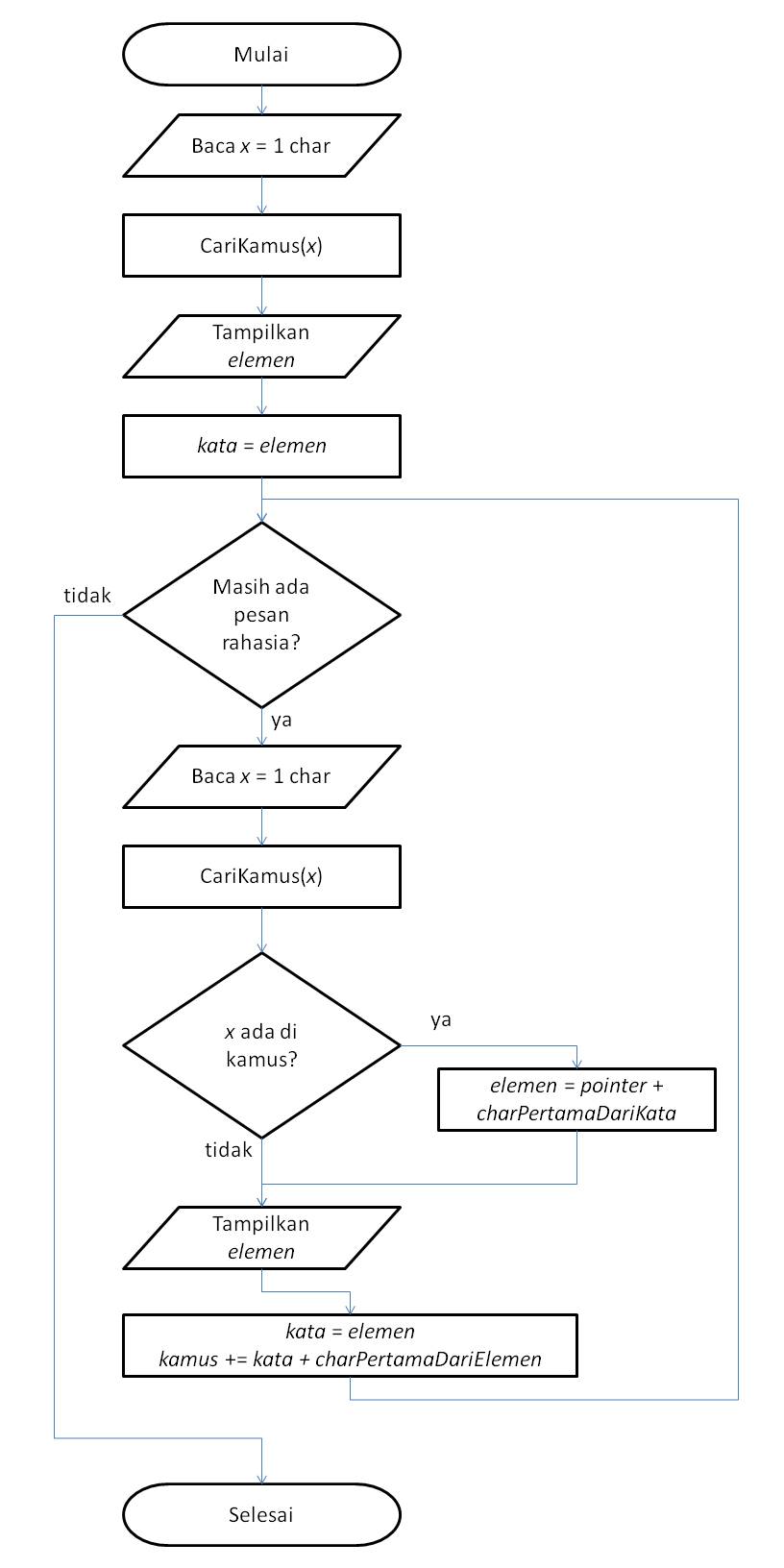
Sedangkan *pseudocode* untuk algoritma *decode* dari LZW dapat dilihat pada Gambar 3 berikut [5].

|  |  |
| --- | --- |
| Algoritma *decode* LZW | |
| 1: | Baca token *x* dari berkas terkompresi |
| 2: | Cari kamus untuk elemen *x* |
| 3: | Keluaran *elemen* |
| 4: | *kata* 🡨 *elemen* |
| 5: | **while** bukan **EOF** lakukan |
| 6: | Baca *x* |
| 7: | Cari kamus untuk elemen *x* |
| 8: | **if** tidak ada kamus untuk elemen *x* **then** |
| 9: | *elemen* 🡨 *kata* + *charPertamaDariKata* |
| 10: | **end if** |
| 11: | Keluaran *elemen* |
| 12: | Tambahkan *kata* + *charPertamaDariElemen* ke kamus |
| 13: | *kata* 🡨 *elemen* |
| 14: | **end while** |

Gambar 3. *Pseudocode* *decode* LZW.

Dapat dilihat pada *pseudocode* *decode* LZW di atas bahwa algoritma LZW akan membaca indeks sambil membuat kamus sendiri. Hasil kamus yang dibuat oleh algoritma *decode* LZW sama dengan hasil kamus yang dibuat oleh algoritma *encode* LZW karena metode pembuatan kamus sama asalkan inisialisasi awal kamus keduanya sama. Oleh karena itu inisialisasi kamus awal untuk algoritma *encode* LZW dan algoritma *decode* LZW harus sama.

Untuk memperjelas *pseudocode* *decode* diataas maka penulis membuat sebuah diagram alir. Diagram alir dari algoritma *decode* LZW dapat dilihat pada Gambar 4 berikut.



Gambar 4. Diagram alir *decode* LZW

Dapat dilihat pada Gambar 4 bahwa algoritma *decode* untuk LZW ini juga membuat kamus sendiri secara dinamis untuk men*decode* hasil kompresi dari LZW. Kamus yang dibuatpun bersifat dinamis sehingga hasil akhir kamus *encode* dan hasil kamus *decode* bersifat identik.

## 8.5. PHP *Free Chat*

PHP *Free Chat* adalah aplikasi *chatting* gratis yang cepat, bisa disesuaikan dan multi bahasa yang menggunakan *filesystem* sederhana untuk pesan dan penyimpanan *nickname* [7].

Arsitektur dari PHP Free Chat dibagi menjadi dua bagian [7]:

* Klien: *plugin* jQuery yang bertugas untuk menampilkan antarmuka *chat* dan untuk berkomunikasi dengan sisi server misalnya dengan menggunakan AJAX.
* Server: arsitektur *RESTful* yang dikodekan dalam PHP menggunakan *framework* Slim bertanggung jawab atas logika *chat*. Server menyimpan pesan dan mengirim pesan *update* ke klien menggunakan metode HTTP klasik (GET, POST, PUT, DELETE).

Berikut adalah contoh dari komunikasi dasar antara klien dan server [7]:

* Klien meminta server untuk otentikasi pengguna, server menyimpan pengguna dan mengembalikan sesi *id* ke klien.
* Klien bergabung ke kanal, server menyimpan bahwa pengguna ini bergabung dalam saluran dan mengirimkan pesan "bergabung" untuk setiap pengguna yang terhubung pada saluran ini.
* Klien mengirimkan pesan ke kanal ini, server mempublikasikan pesan ini ke dalam antrian untuk setiap pengguna terhubung dalam saluran ini.
* Klien membaca pesan yang tertunda, server membaca antrian user dan mengembalikan daftar pesan, klien menampilkan pesan pada antarmuka.

# RINGKASAN ISI TUGAS AKHIR

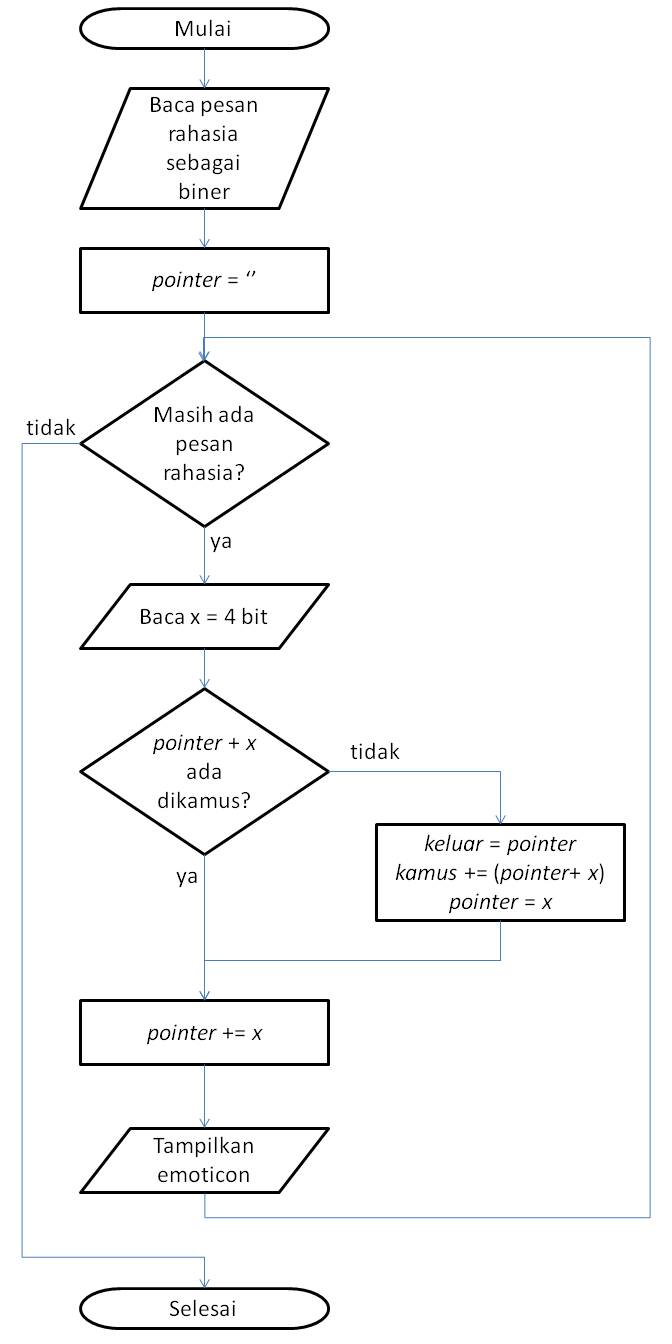
Pada Tugas Akhir ini penulis mengusulkan metode steganografi dengan menggabungkan *emoticon* dengan LZW untuk menyimpan pesan rahasia sehingga algoritma LZW pada kasus ini digunakan sebagai kamus dinamis untuk steganografi *emoticon*.

*Pseudocode* untuk algoritma menyisipkan pesan dapat dilihat pada Gambar 5 berikut.

|  |  |
| --- | --- |
| Algoritma menyisipkan pesan pada *emoticon* | |
| 1: | Inisialisasi indeks dari 1 hingga 16 pada susunan emoticon dalam biner |
| 2: | *buffer* 🡨 baca\_seluruh\_pesan\_asli\_sebagai\_biner() |
| 3: | *pointer* 🡨 ‘’ |
| 4: | **while** buffer bukan **EOF** lakukan |
| 5: | *x* 🡨 baca\_4bit\_berikutnya() |
| 6: | **while** selamanya |
| 7: | **if** *pointer* + *x* ada di kamus **then** |
| 8: | *pointer* 🡨 *pointer* + *x* |
| 9: | **else** |
| 10: | Keluaran emoticon dari kamus |
| 11: | Tambahkan *pointer* + *x* ke kamus |
| 12: | *pointer* 🡨 *x* |
| 13: | **end if** |
| 14: | **end while** |
| 15: | **end while** |
| 16: | Keluaran emoticon dari kamus |

Gambar 5. *Pseudocode* penyisipan pesan dari metode yang diusulkan.

Untuk diagram alir metode yang diusulkan dari menyisipkan pesan dapat dilihat pada Gambar 6 berikut.



Gambar 6. Diagram alir penyisipan pesan dari metode yang diusulkan

Jika melihat pada Gambar 6 maka dapat dilihat bahwa metode yang diusulkan ini akan selalu membaca masukan 4 *bit* hal ini tentunya sedikit berbeda dengan algoritma LZW yang menggunakan masukan satu karakter. Penggunaan masukan 4 *bit* dimaksudkan agar inisialisasi kamus lebih cepat dan juga memungkinkan kombinasi *bit* dalam satu *emoticon* lebih banyak. Hal lain yang membedakan metode ini dengan algoritma LZW adalah hasil keluaran yang berupa *emoticon*.

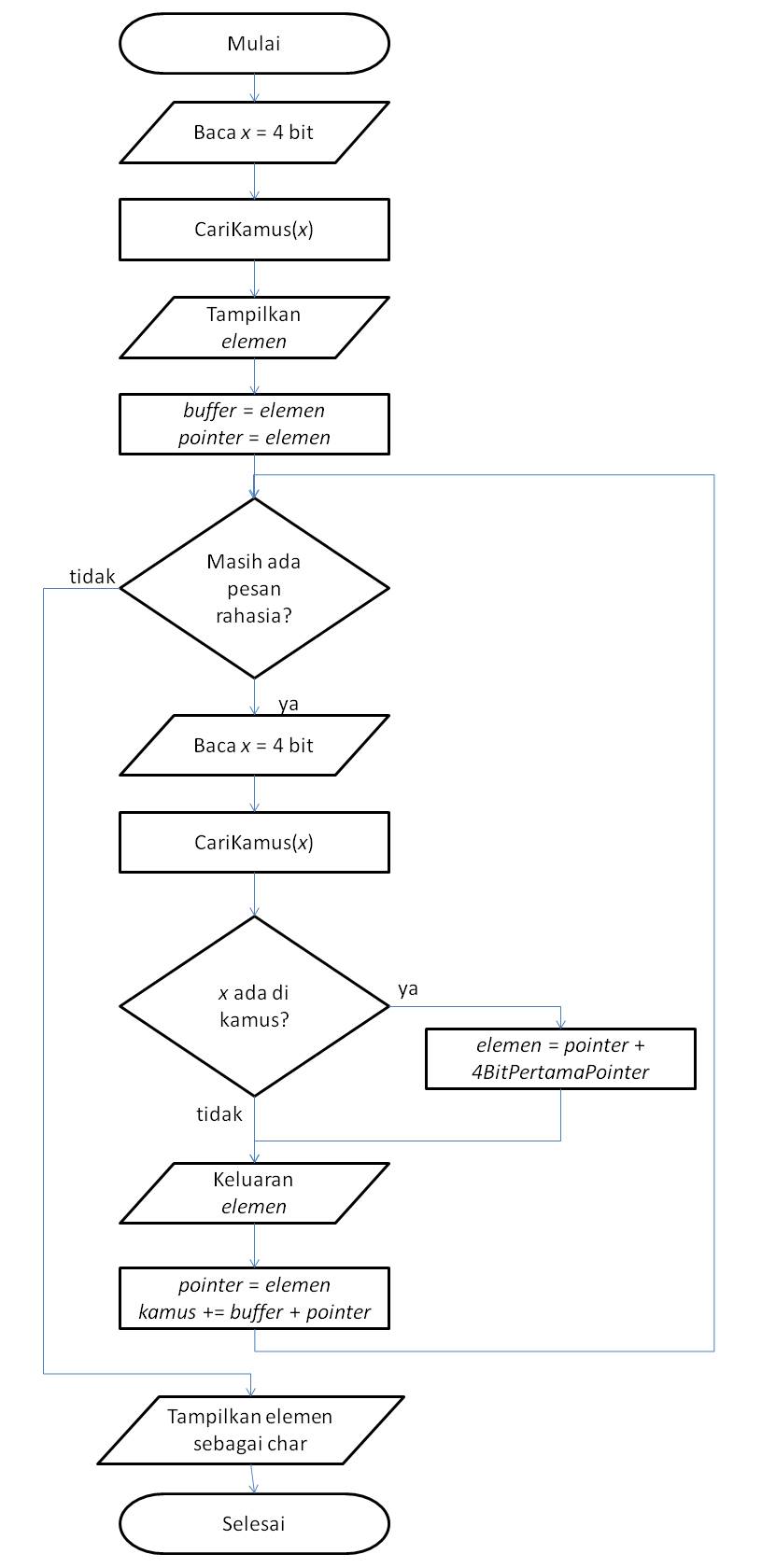
Sedangkan *pseudocode* untuk algoritma ekstraksi pesan dapat dilihat pada Gambar 7 berikut.

|  |  |
| --- | --- |
| Algoritma ekstraksi pesan dari *emoticon* | |
| 1: | Baca emoticon *x* |
| 2: | Cari kamus untuk elemen *x* |
| 3: | Keluaran *elemen* |
| 4: | *buffer* 🡨 *elemen* |
| 5: | *pointer* 🡨 *elemen* |
| 6: | **while** bukan **EOF** lakukan |
| 7: | Baca emoticon *x* |
| 8: | Cari kamus untuk elemen *x* |
| 9: | **if** tidak ada kamus untuk elemen *x* **then** |
| 10: | *elemen* 🡨 *pointer* + *4BitPertamaDariPointer* |
| 11: | **end if** |
| 12: | Keluaran *elemen* |
| 13: | Tambahkan *kata* + *4BitBerikutnyaDariElemen* ke kamus |
| 14: | *pointer* 🡨 *elemen* |
| 15: | *buffer* 🡨 *buffer* + *pointer* |
| 16: | **end while** |
| 17: | Tampilkan *elemen* sebagai char |

Gambar 7. *Pseudocode* ekstraksi pesan dari metode yang diusulkan.

Dapat dilihat dari Gambar 7 bahwa metode yang diusulkan penulis hamper sama dengan algoritma LZW. Hal yang membedakan metode yang diusulkan penulis dengan algoritma LZW adalah digunakannya *emoticon* sebagai pengganti indeks kamus dan juga hasil keluaran berupa biner dengan panjang yang tidak menentu tetapi panjang biner tetap kelipatan empat. Karena keluaran berupa biner maka keluaran harus ditampung terlebih dahulu dan setelah seluruh biner rahasia didapat barulah dirubah kebentuk karakter. Hal ini perlu dilakukan karena jika keluaran biner langsung dirubah ke dalam bentuk karakter maka pesan rahasia yang asli akan berbeda dengan pesan keluaran biner.

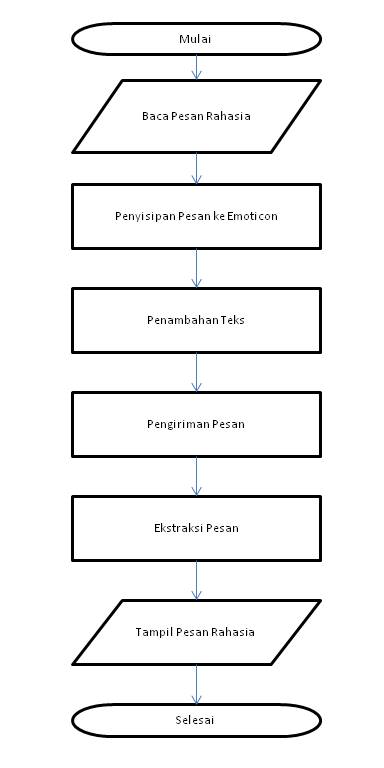
Untuk memudahkan membaca *pseudocode* ekstraksi pesan dari metode yang diusulkan di atas, maka penulis membuat sebuah diagram alir. Diagram alir metode yang diusulkan penulis dari ekstraksi pesan dapat dilihat pada Gambar 8 berikut.



Gambar 8. Diagram alir ekstraksi pesan dari metode yang diusulkan

Jika melihat pada Gambar 8 di atas maka dapat dilihat bahwa metode ekstraksi yang diusulkan sama dengan metode *decode* dari algoritma LZW. Yang membedakan adalah masukan yang dibaca merupakan *emoticon* dan keluarannya berupa biner. Dengan begitu setelah seluruh pesan rahasia diterima dalam bentuk biner maka dilakukan konversi kedalam bentuk karakter.

Untuk menyamarkan metode steganografi ini sehingga lebih sulit untuk disadari bahwa ada pesan rahasia maka ditambahkan teks berbahasa Indonesia sebelum *emoticon*. Dengan begitu diagram alir dari langkah metode ini dapat dilihat pada Gambar 9 berikut.



Gambar 9. Diagram alir urutan metode yang diusulkan

Gambar 9 di atas menjelaskan urutan / alir / langkah kerja dari metode yang diusulkan. Untuk penambahan teks, seperti yang terlihat di Gambar 5, digunakan teks berbahasa Indonesia pada setiap *emoticon* yang ada dengan terlebih dahulu membuat kamus.

# METODOLOGI

## Penyusunan proposal tugas akhir

Tahap pertama untuk memulai pengerjaan Tugas Akhir yaitu penyusunan proposal. Pada proposal ini, penulis mengajukan gagasan mengenai steganografi dengan menggunakan *emoticon* berbasis LZW.

## Studi literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur yang diperlukan. Informasi didapatkan dari buku dan materi-materi lain yang berhubungan dengan metode yang digunakan dalam pengerjaan Tugas Akhir ini dan juga didapat dari internet maupun buku acuan.

## Implementasi perangkat lunak

Implementasi merupakan tahap untuk membangun metode tersebut. Untuk mengimplementasikan metode tersebut digunakan kakas bantu yaitu gedit.

## Pengujian dan evaluasi

Pada tahap ini dilakukan pengujian dengan menggunakan beberapa teks *corpus canterbury* untuk mencoba aplikasi apakah telah sesuai dengan rancangan dan desain metode yang dibuat, serta mencari ketidaksesuaian yang ada pada program untuk selanjutnya dilakukan perbaikan dan penyempurnaan. Pengujian aplikasi ini dapat dilakukan antara lain sebagai berikut:

1. Aplikasi dapat menyisipkan pesan rahasia pada *emoticon*.
2. Aplikasi dapat mengirimkan *emoticon*.
3. Aplikasi dapat mengekstrak pesan rahasia dari *emoticon*.

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

Jadwal kegiatan pengerjaan Tugas Akhir ini dapat dilihat pada Tabel 1 berikut.

Tabel 1. Jadwal kegiatan pengerjaan Tugas Akhir.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | 2014 | | | | | | | | | | | | | | | | | | | | |
| Februari | | | | Maret | | | | April | | | | Mei | | | | | Juni | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | Wang, Z.-H., Chang, C.-C., Li, M.-C., & Kieu, T. (2009). Emoticon-based Text Steganography in Chat. *Second Asia-Pacific Conference on Computational Intelligence and Industrial Applications*, 457-460. |
| [2] | Listega : list-based steganography metodology. *Int J Inf Secur, vol. VIII,*, 247-261. |
| [3] | Salomon, D., & Motta, G. (2010). *Handbook of Data Compression (5th edition).* New York: Springer. |
| [4] | Sayood, K. (2000). *Introduction to Data Compression (3rd edition).* San Fransisco: Morgan Kaufmann Publisher. |
| [5] | Pu, I. M. (2006). *Fundamental Data Compression.* Oxford: Elsevier. A, D. (2009). |
| [6] | Lelewer, D., & Hirschberg, D. (1987). Data Compression. *Information and Computer Science*. |
| [7] | phpfreechat. (2013, Juny). *http://www.phpfreechat.net/documentation/*. Retrieved Feruary 2014, from http://www.phpfreechat.net/: http://www.phpfreechat.net/documentation/ |
| [8] | Salomon, D. (2007). *Data Compression, The Complete Reference (4th edition).* London: Springer-Verlag. |
| [9] | Ptaszynski, M., Rzepka, R., Araki, K., & Momouchi, Y. (2011). Research on Emoticons: Review of the Field and Proposal of Research Framework. *The Association for Natural Language Processing*, 1159-1162. |