

# Pengembangan Algoritma Steganografi pada Objek 3D

Fakhri

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Bandung, Indonesia  
13510048@std.stei.itb.ac.id

**Abstrak**—Pada makalah ini dibahas satu langkah pengembangan algoritma steganografi adaptif dengan menggunakan model atau objek 3D sebagai media penyimpanan pesan. Pengembangan ini bertujuan mendapatkan peningkatan steganografi pada objek 3D sehingga kapasitas penyimpanan dan *undetectability* dalam penyembunyian pesan rahasia meningkat tanpa mengurangi akurasi pesan sebelum dan sesudah steganografi. Pengembangan ini menggunakan beberapa algoritma serupa yang telah ada yang kemudian diambil beberapa komponen unggulan masing – masing algoritma tersebut lalu digabungkan ke dalam sebuah algoritma baru yang disebut algoritma gabungan. Algoritma – algoritma yang digabungkan antara lain adalah algoritma oleh Cheng dan Wang (2007), algoritma oleh Chao (2009) dan algoritma oleh Tsai (2012). Penggabungan ini menyebabkan algoritma gabungan memiliki keunggulan di segi kapasitas penyimpanan dan *undetectability*. Walaupun begitu terdapat kelemahan dari aspek akurasi hasil ekstraksi akibat keterbatasan penanganan bilangan berkoma.

**Keywords**— Steganografi, Objek 3D, *Vertex*, *Node*, koordinat.

## I. PENDAHULUAN

Steganografi adalah bentuk komunikasi tersembunyi sehingga hanya pengirim dan penerima saja yang menyadarinya (Cox, 2003). Steganografi adalah salah satu bidang yang telah sejak lama diteliti dan dikembangkan. Hal ini sejalan dengan peningkatan teknologi sehingga meningkatkan kebutuhan terhadap pemrosesan data. Komunikasi saat ini telah bergeser dari penggunaan media tulis menjadi media digital terutama menggunakan jaringan internet. Dengan adanya internet banyak kemudahan di banyak aspek yang dapat diperoleh. Kebutuhan terhadap data pada kehidupan sehari-hari dan mudahnya persebaran informasi atau data di internet ini menyebabkan privasi dan kerahasiaan dalam komunikasi serta penyimpanan data tertentu menjadi salah satu kebutuhan penting. Steganografi sebagai salah satu solusi kebutuhan ini dikembangkan mulai dari media tulis untuk bentuk komunikasi lama hingga beralih ke media digital. Bantu perkembangan steganografi pada media komunikasi digital terutama komputer salah satunya adalah pemanfaatan objek 3D. Pengembangan ini ditujukan untuk menemukan dua aspek utama dalam steganografi, yaitu kapasitas penyimpanan dan meminimalkan pendeteksian sekecil mungkin. Terdapat beberapa riset algoritma yang dapat

digunakan untuk melakukan steganografi pada objek 3D. Pengembangan algoritma steganografi pada objek 3D telah lama dikembangkan. Diawali oleh Cayre dan Macq (2003) yang melakukan steganografi 3D menggunakan *Quantization Index Modulation* (QIM), Cheng dan Wang (2006) dengan *Principal Component Analysis* (PCA), Cheng dan Wang (2007) yang memanfaatkan *contagious diffusion* disertai *sliding*, *extending*, dan *rotating*, lalu algoritma oleh Chao (2009) yang memanfaatkan *layer* dalam peningkatan kapasitas, dan algoritma oleh Tsai (2012) yang menggunakan metode *vertex decimation*.

Masalah utama yang dibahas dalam konteks algoritma steganografi adalah bagaimana caranya menemukan algoritma steganografi terlebih yang tergolong adaptif pada objek 3D yang lebih optimal dari pendahulunya. Jika dijabarkan lebih dalam akan muncul beberapa masalah spesifik, yaitu:

1. Bagaimana caranya memaksimalkan kuantitas pesan rahasia yang dapat disembunyikan pada media cover
2. Bagaimana caranya meminimalkan kemungkinan pendeteksian suatu stego objek
3. Bagaimana caranya memastikan hasil ekstraksi tetap akurat dan sama dengan pesan rahasia sebelum penyembunyian.

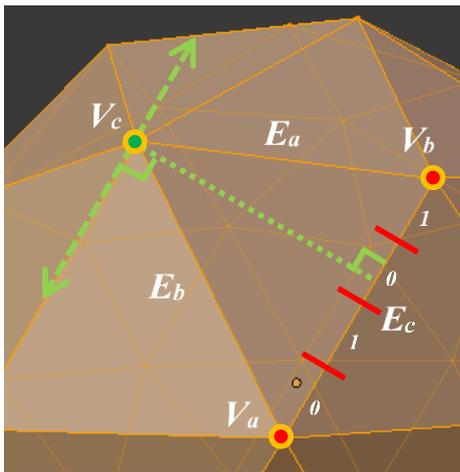
Untuk itu, diperlukan algoritma yang pada makalah ini diperoleh dari penggabungan beberapa algoritma steganografi yang telah ada dengan ekspektasi yang menjawab permasalahan. Sehingga tujuan dari penelitian ini adalah mengembangkan algoritma steganografi gabungan pada objek 3D, mengembangkan perangkat lunak implementasi algoritma gabungan, mengembangkan perangkat lunak implementasi algoritma - algoritma sumber gabungan, dan melakukan perbandingan performansi antara algoritma gabungan terhadap algoritma sumber penggabungan

## II. TEORI DAN PENELITIAN TERKAIT

### A. Steganografi pada Objek 3D

Elemen pada objek 3D yang dimanfaatkan dalam steganografi adalah koordinat *vertex* dari suatu *mesh* objek 3D. Pemanipulasian *vertex* ini perlu diatur agar memberikan efek

perubahan objek yang minimal dan mampu menampung pesan rahasia. Penyembunyian pesan ke dalam suatu *vertex* secara sederhana dilakukan dalam satuan bit dengan memproyeksikan suatu *vertex* pada cover objek 3D kepada sebuah garis yang memiliki indikator berupa zona – zona yang melambangkan nilai bit berbeda. Seperti yang terdapat pada gambar II-1, vektor  $V_c$  diproses ke sebuah garis, yaitu *edge*  $E_c$  yang telah dibagi ke dalam 4 zona nilai. Proyeksi tersebut terletak pada zona yang melambangkan nilai 0, sehingga kondisi  $V_c$  saat itu memiliki nilai 0 bit. Apabila bit yang ingin disembunyikan adalah 1, maka  $V_c$  digeser sejajar dengan  $E_c$  sehingga proyeksinya berada pada zona 1. Metode ini diterapkan untuk *vertex* – *vertex* yang terpilih dalam urutan tertentu, seperti menggunakan *pseudo random* dengan kunci atau pola lainnya hingga seluruh bit pesan rahasia tersisipkan. Penyisipan ini menyebabkan *vertex* penyusun  $E_c$ , yaitu  $V_a$  dan  $V_b$  tidak boleh berpindah posisi pada penyisipan di *vertex* selanjutnya. Ketika penyisipan atau penyembunyian pesan rahasia telah selesai, akan dihasilkan stego objek berupa sebuah model objek 3D yang menyerupai cover objek.



Gambar II-1 : Steganografi Pada Objek 3D Secara Sederhana

Ekstraksi dari sebuah stego objek dilakukan dengan cara membaca zona proyeksi masing – masing *vertex* lalu mengumpulkan seluruh nilai bit hingga menjadi pesan rahasia. Pemilihan *vertex* dilakukan dengan pola yang sama pola pemilihan *vertex* saat penyembunyian pesan. Penyamaan pola ini dapat dilakukan dengan banyak cara, salah satunya penggunaan kunci rahasia sebagai pembangkit bilangan *pseudo random*.

**B. Algoritma Steganografi pada Objek 3D sumber Penggabungan**

Dalam memecahkan permasalahan utama dalam makalah ini, pembentukan solusi dilakukan dengan cara menggabungkan fitur – fitur unggulan dari algoritma – algoritma steganografi pada model objek 3D yang telah ada. Algoritma tersebut adalah algoritma Cheng dan Wang (2007), Chao (2009), dan Tsai (2012). Katiga algoritma ini memiliki kelebihan dan kekurangan, selain itu terdapat pula kesamaan beberapa cara pengolahan data di dalam algoritmanya, sehingga penggabungan di antaranya lebih mudah dan dapat saling menutupi kekurangan masing – masingnya.

Algoritma Cheng dan Wang (2007) melakukan steganografi dengan pola pemilihan berbasis *face* yang disebut *contagious diffusion*. Sehingga setiap *face* yang terpilih akan dibagi menjadi dua atribut utama. Pertama *vertex* yang akan disisipkan pesan lalu yang kedua adalah *edge* pembanding, yaitu *edge* yang tidak tersusun atas *vertex* yang akan disisipkan pesan rahasia. Prinsip utama dalam penyembunyian atau penyisipannya adalah perbandingan posisi koordinat suatu *vertex* terhadap tiga indikator, pertama terhadap *edge* pembanding, kedua yaitu garis tinggi dengan *edge* pembanding sebagai alasnya, dan ketiga adalah sudut yang terbentuk antara *face* terpilih dengan *face* tetangga yang terhubung oleh *edge* pembanding. Ketiga indikator ini menyebabkan suatu *vertex* dapat berubah posisi sebanyak tiga kali dan menyebabkan *face* tersebut bergeser, diperluas, maupun berotasi. Pola pemilihan *face* yang digunakan (*contagious diffusion*) menyebabkan *vertex* yang telah disisipkan pesan, *vertex* penyusun *edge* pembandingnya, serta *face* terpilih ditandai sehingga tidak terjadi redundansi.

Algoritma Chao (2009) melakukan steganografi dengan pola pemilihan *vertex* yang belum tersisipkan pesan pada *face* tetangga secara *pseudo random* dengan kunci rahasia sebagai pembangkitnya. Prinsip steganografi yang digunakan adalah perbandingan posisi koordinat suatu *vertex* terhadap 3 sumbu koordinat baru yang dibentuk dari tiga *vertex* yang terpilih pertama kali. Tiga *vertex* ini tidak boleh diubah posisinya agar sumbu koordinat yang terbentuk pada proses penyisipan dan ekstraksi sama. Setelah dibandingkan, perubahan posisi *vertex* dilakukan berulang kali dalam sebuah vektor yang sejajar dengan masing – masing sumbu koordinat. Perubahan posisi ini dilakukan untuk *vertex* dengan proyeksi bernilai berbeda dengan bit pesan rahasia maupun sama. Jumlah perubahan posisi berulang kali ini disebut juga sebagai layer akibat terbentuknya lapisan kondisi perubahan posisi. Posisi yang berubah ini dapat dikembalikan tahap demi tahap hingga kembali ke posisi zona layaknya penyisipan belum dilakukan. Hal ini lah yang dimanfaatkan dalam proses ekstraksi. Ekstraksi dilakukan dengan pola pemilihan *vertex* yang sama dengan proses penyisipan, lalu tiap *vertex* dibaca zona proyeksinya, kemudian posisi *vertex* tersebut diubah layaknya satu tahap sebelum pada proses penyisipan, lalu dibaca kembali zona proyeksi barunya, dan kembali diubah posisinya dan dibaca sebanyak jumlah layer yang menjadi inputan pengguna. Proses ini dilakukan sebanyak jumlah layer untuk tiap satu sumbu koordinat.

Algoritma Tsai (2012) membentuk pola pemilihan *vertex* dengan *vertex* decimation berdasarkan tiga kategori input, yaitu layaknya format *vertex* yang tertulis dalam file objek, mengutamakan *vertex* dengan tetangga paling banyak, atau mengutamakan *vertex* dengan tetangga paling sedikit. Urutan yang terbentuk dari proses ini tergolong atas dua, yaitu *vertex* yang dapat disisipkan pesan dan tidak. Pesan yang tidak dapat disisipkan pesan rahasia digunakan untuk menghasilkan kunci rahasia. *Vertex* yang dapat disisipkan pesan memiliki atribut berupa *vertex* – *vertex* yang mereferensikannya. Prinsip steganografi yang digunakan adalah perbandingan posisi koordinat *vertex* terpilih terhadap titik koordinat baru. Titik koordinat baru dibentuk setiap kali suatu *vertex* akan disisipkan pesan. Titik ini adalah titik pusat dari *vertex* - *vertex* referensi atas *vertex* terpilih. *Vertex* terpilih kemudian digeser dalam satu arah yang sejajar dengan vektor yang terbentuk antara pusat

koordinat baru dengan *vertex* terpilih. Penyisipan dilakukan dengan menghitung kapasitas bit maksimum yang dapat disisipkan sebuah *vertex* dihitung dari jaraknya terhadap pusat koordinat baru. Perhitungan jarak ini tidak berubah setelah penyisipan dilakukan, sehingga hal ini dimanfaatkan dalam proses ekstraksi. Pada ekstraksi, urutan *vertex* dan koordinat baru dibentuk dengan pola dan nilai yang sama seperti pada proses penyisipan, begitu pula nilai kapasitas bit maksimum per *vertex*. Tiap bit yang dibaca kemudian dikumpulkan menjadi pesan rahasia.

### C. Distorsi

Nilai distorsi yang digunakan adalah hasil bagi dari *Root Mean Square Error* (RMSE) terhadap panjang diagonal dari *bounding box* model objek 3D keseluruhan. Rumus RMSE dapat dilihat pada persamaan (1).

$$RMSE = \sqrt{\frac{1}{|V|} \sum_{i=1}^{|V|} \|v_i - v'_i\|^2} \quad (1)$$

Dari persamaan (1), nilai  $v$  yang dimaksud adalah koordinat dari tiap *vertex* yang kemudian dalam rumus tersebut dihitung untuk seluruh *vertex* dalam objek 3D.

## III. ALGORITMA GABUNGAN

### A. Penyisipan atau Penyembunyian Pesan

Terdapat dua bagian, yaitu :

#### 1. Pre Proses

Pada tahap ini dilakukan pemrosesan variabel input, yaitu jumlah layer dan nilai *threshold*. Jumlah layer digunakan dalam penyisipan metode Chao (2009) dan *threshold* dalam penyisipan metode Tsai (2012). Selain itu dibuat tabel – tabel perhitungan yang dibutuhkan:

##### a. *Vertex Body Table* (VBT)

Berisi elemen sejumlah *face* yang tersusun atas *vertex-vertex* penyusun setiap *face* tersebut. Tabel ini sesuai dengan tabel VBT yang digunakan dalam algoritma Cheng dan Wang (2007).

##### b. *Polygon Neighbor Table* (PNT)

Berisi elemen sejumlah *face* yang tersusun atas beberapa *face* yang menjadi tetangga dari *face* yang ditunjuk oleh indeks. Tabel ini sesuai dengan tabel PNT yang digunakan dalam algoritma Cheng dan Wang (2007).

##### c. *Vertex Neighbor Table* (VNT)

Berisi elemen sejumlah *vertex* yang tersusun atas beberapa *vertex* yang menjadi tetangga dari *vertex* yang ditunjuk oleh indeks. Tabel ini sesuai dengan tabel VNT<sub>*v*</sub> yang digunakan dalam algoritma Tsai (2012).

##### d. NMS

Berisi sejumlah *vertex* yang tersusun atas beberapa *vertex* yang mereferensikan *vertex* yang ditunjuk oleh indeks. Tabel ini dihasilkan dari proses *vertex* decimation dan digunakan dalam menentukan suatu *vertex* dapat disisipkan atau tidak. *Vertex* yang dapat disisipkan hanyalah *vertex* yang memiliki *vertex* lain

yang merreferensikannya, dengan kata lain memiliki elemen NMS. Tabel ini sesuai dengan tabel NMS<sub>*v*</sub> yang digunakan dalam algoritma Tsai (2012).

##### e. *Vertex Selection Order* (VSO)

Berisi urutan penyisipan sesuai hasil *vertex* decimation. Dalam penyisipan, urutan yang digunakan adalah kebalikan dari urutan VSO. Tabel ini sesuai dengan tabel VSO yang digunakan dalam algoritma Tsai (2012).

### 2. Proses

Pada tahap ini dilakukan penyisipan data untuk seluruh *vertex* kecuali koordinat global. Penyisipan dilakukan dalam tiga bagian:

#### a. Penyisipan pada *vertex* tanpa elemen NMS.

*Vertex* yang termasuk kategori ini adalah *vertex* yang pada algoritma Tsai (2012) tidak termasuk *vertex* yang disisipkan pesan karena digunakan untuk menentukan koordinat global. Penyisipan dilakukan menggunakan algoritma Chao (2009) yang dimodifikasi. Perbedaannya adalah penyisipan dilakukan dengan menunjuk satu *vertex* sebagai acuan perubahan posisi untuk *vertex-vertex* lainnya dan tidak melakukan pemilihan *vertex* selanjutnya dengan menggunakan bilangan acak, melainkan menggunakan urutan VSO. Lalu interval yang digunakan bernilai sama dengan nilai *threshold* yang dikalikan dengan nilai absolut dari *vertex* acuan. Nilai ini digunakan dengan dasar yaitu menghasilkan sebuah nilai interval yang lebih kecil dari jarak antara dua *vertex* dalam proses penyisipan.

#### b. Pembentukan koordinat global.

Diperoleh melalui penentuan titik tengah pada *bounding box* yang terbentuk oleh *vertex-vertex* yang tidak memiliki elemen NMS. Selain itu dihasilkan juga sebuah kunci melalui pembagian nilai *threshold* terhadap diagonal dari *bounding box* tersebut. Kunci ini menjadi syarat dalam proses ekstraksi.

#### c. Penyisipan pada *vertex* berelemen NMS sesuai kebalikan urutan VSO.

Penyisipan ini dilakukan dengan menggunakan metode Tsai (2012) dan Chao (2009), yaitu sebagai berikut:

i. Diawali dengan penyisipan menggunakan algoritma Tsai (2012) Penyisipan dilakukan seperti normal, yaitu mengubah posisi *vertex* dengan acuan koordinat global dan beberapa *vertex* yang mereferensikannya dalam NMS.

ii. *Vertex* yang telah diubah posisinya disisipkan kembali dengan algoritma Chao (2009). Penyisipan dilakukan tepat setelah penyisipan dengan algoritma Tsai (2012) selesai dilakukan sehingga urutan pemilihan *vertex* tetap sama. Perubahan posisi *vertex* mengacu pada koordinat global dan tidak melakukan pemilihan *vertex* selanjutnya dengan bilangan acak. Arah perubahan posisi berlawanan dengan arah dari algoritma Tsai (2012) agar posisi *vertex* dapat sedekat mungkin dengan posisi awal sebelum penyisipan. Hal ini dilakukan dengan harapan meminimalkan distorsi.

Hal utama yang dimanfaatkan dalam algoritma gabungan ini adalah adanya sifat *reversible* dari algoritma Chao (2009) sehingga memungkinkan dilakukannya penumpukan dua algoritma steganografi sekaligus. Pada algoritma ini dapat terlihat bahwa dari ketiga algoritma yang digabungkan dua algoritma lebih banyak digunakan, yaitu algoritma Chao (2009) dan Tsai (2012). Pemilihan ini dilakukan dengan prioritas mendahulukan aspek kapasitas dan satu-satunya yang memilikisifat *reversible*, sehingga algoritma Chao (2009) menjadi algoritma yang utama.

Pemilihan kedua adalah algoritma dengan kemampuan adaptif dan berdistorsi rendah. Pada kategori ini secara teori lebih baik memilih algoritma Cheng dan Wang (2007). Namun algoritma melakukan pembacaan nilai dengan membandingkan posisi tiga *vertex* dalam satu *face*, sehingga variasi keterikatan antara suatu *vertex* subjek penyisipan dan pembandingnya sangat tinggi. Oleh karena itu, kemungkinan kesalahan akibat penumpukan dua algoritma lebih besar. Sedangkan pada algoritma Tsai (2012) digunakan koordinat global yang menyebabkan jumlah variasi keterikatan yang lebih rendah. Selain itu penggunaan koordinat global adalah suatu kesamaan yang dimiliki oleh algoritma Chao (2009). Algoritma Cheng dan Wang (2007) dimanfaatkan dalam pre proses akibat kelengkapan tabel yang memudahkan perhitungan.

### B. Ekstraksi

Seperti pada penyisipan terdapat dua tahapan:

#### 1. Pre Proses

Pada tahap ini dilakukan pembuatan beberapa table layaknya pada pra proses penyisipan.

#### 2. Proses

Ekstraksi dilakukan dengan melakukan invers dari proses penyisipan. Dilakukan dalam empat bagian:

##### a. Penentuan koordinat global.

Dilakukan dengan menentukan titik tengah *bounding box* yang dibentuk oleh seluruh *vertex* yang tidak memiliki *vertex* referensi. Selain itu ditentukan juga nilai *threshold* dengan mengalikan kunci pada masukan dengan panjang diagonal pada *bounding box*.

##### b. Ekstraksi pada *vertex* yang tidak memiliki elemen pada NMS.

Ekstraksi dilakukan dengan menggunakan algoritma Chao (2009) sesuai dengan jumlah layer masukan. *Vertex* acuan ekstraksi bukanlah koordinat global, melainkan sebuah *vertex* yang dipilih sesuai proses penyisipan di antara *vertex* yang tidak memiliki elemen pada NMS yang juga tergolong tidak memiliki *vertex* referensi.

##### c. Mengembalikan posisi *vertex* tanpa elemen pada NMS.

Posisi *vertex* ini dikembalikan menjadi posisi sebelum ekstraksi dilakukan. Hal ini dilakukan karena pada proses penyisipan untuk *vertex* berelemen pada NMS dilakukan terhadap *vertex* dengan posisi seperti ini, sehingga ekstraksi juga harus memiliki kondisi yang sama.

##### d. Ekstraksi pada *vertex* yang memiliki elemen pada NMS. Ekstraksi ini dilakukan sesuai urutan pemilihan *vertex* pada VSO. Dibagi atas tiga tahap:

- i. Ekstraksi menggunakan algoritma Chao (2009). Dilakukan dengan acuan koordinat global dan sebanyak layer yang menjadi masukan. Algoritma ini dilakukan terlebih dahulu untuk mengembalikan posisi *vertex* sesuai sifat *reversible*-nya.
- ii. Ekstraksi menggunakan algoritma Tsai (2012). Dilakukan dengan acuan koordinat global dan *vertex* yang menjadi referensinya sesuai elemen pada NMS.
- iii. Hasil ekstraksi algoritma Tsai (2012) diletakkan lebih awal dibandingkan hasil ekstraksi algoritma Chao (2009).

## IV. IMPLEMENTASI ALGORITMA GABUNGAN

Lingkungan implementasi utama atau pembuatan perangkat lunak dari algoritma ini menggunakan sistem operasi Windows 8.1 Pro 64-bit, prosesor AMD Phenom II X4 550 Processor (4 CPUs), ~3.1GHz, memori 4096MB RAM, dan DirectX 11. Antarmuka yang dirancang untuk algoritma sumber penggabungan dirancang sama dengan format input dan output masing – masingnya, sedangkan untuk algoritma gabungan antarmuka nya sama dengan antarmuka lagoritma Tsai, namun memiliki tambahan input berupa *layer*. Dari implementasi ini terdapat beberapa batasan, yaitu format file yang dapat diproses hanya \*.obj karena tidak terkompresnya format ini, lalu khusus untuk algritma Chao, seperti pada sumbernya, tidak dapat memproses objek 3D yang memiliki *face* penyusun berbentuk non-triangular.

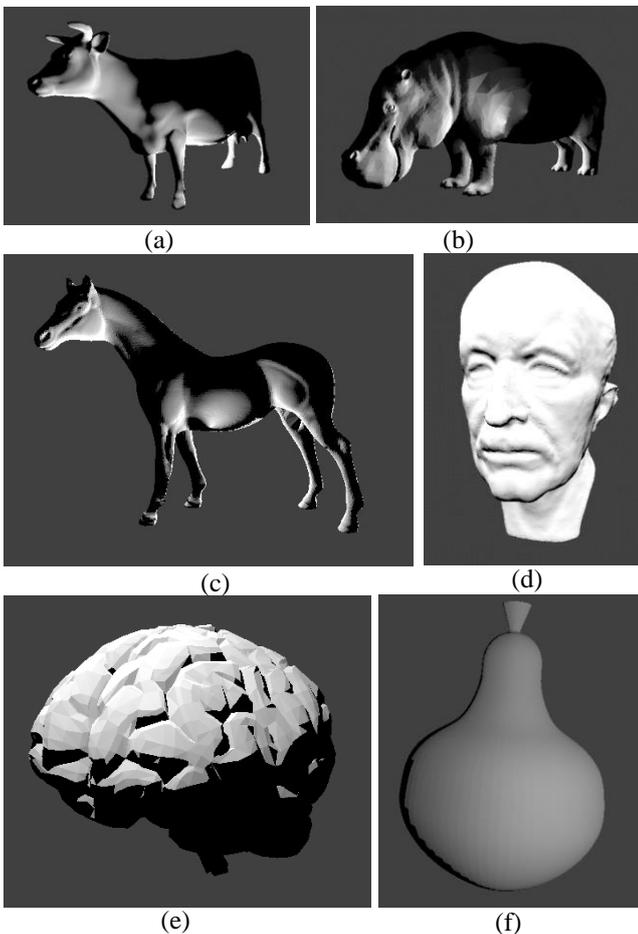
## V. PENGUJIAN ALGORITMA GABUNGAN

Pengujian yang dilakukan dibagi atas tiga bagian, yaitu pengujian aspek kapasitas, *undetectedability*, dan akurasi. Terdapat 6 model objek 3D yang digunakan dan dapat dilihat pada gambar V-1 serta tabel V-1. Konfigurasi input pada implementasi algoritma Cheng dan Wang (2007),  $\alpha$  bernilai 1024,  $\beta$  bernilai 0.0001,  $\gamma$  bernilai 0,035, dan  $\sigma$  bernilai 9. Pada implementasi algoritma Chao (2009), layer berjumlah 5 dan interval berjumlah 10. Pada implementasi algoritma Tsai (2012), urutan yang digunakan adalah Minimum First dan threshold bernilai 0.000003. Lalu untuk algoritma gabungan, urutan yang digunakan adalah minimum first, layer berjumlah 5, dan threshold bernilai 0.000003. Input pesan yang digunakan adalah file berukuran 6064 bits, lalu kunci berupa teks 2456 bits.

### A. Kapasitas

Pengujian ini dilakukan untuk mengetahui kapasitas maksimal tiap algoritma terhadap objek 3D yang diinputkan. Hasil uji dapat dilihat pada tabel V-2.

Algoritma Cheng dan Wang (2007), Chao (2009), serta Tsai (2012) memiliki jumlah yang sesuai dengan analisis kapasitas masing – masingnya. Sedangkan algoritma gabungan memiliki kapasitas yang lebih besar karena menggunakan algoritma Chao dan Tsai dalam satu penyisipan yang sama, sehingga memiliki kapasitas yang sama dengan penjumlahan kedua kapasitas algoritma tersebut.



Gambar V-1 : Objek 3D Pengujian, (a) cow.obj, (b) hippo.obj, (c) horse.obj, (d) maxplanck.obj, (e) brain.obj, dan (f) pear.obj

Tabel V-1: Spesifikasi Objek 3D

Nama Objek	Jumlah Vertex	Jumlah Face
pear.obj	10754	21504
brain.obj	18844	36752
cow.obj	46434	92864
hippo.obj	23105	46202
horse.obj	48484	96964
maxplanck.obj	49132	98260

Tabel V-2: Perbandingan Kapasitas Maksimal Objek dalam Bits dengan Tujuan Meningkatkan Kapasitas Algoritma Gabungan

Objek 3D	Cheng & Wang (2007)	Chao (2009)	Tsai (2012)	Gabungan
pear.obj	96768	161295	105720	267015
brain.obj	169578	282645	147561	430206
cow.obj	207927	346560	185721	532281
hippo.obj	417879	696480	437816	1134296
horse.obj	436338	727245	194757	922002
maxplanck.obj	442170	736965	522279	1259244

Pada implementasinya, data kapasitas pada tabel V-1 diperoleh dari proses analisis yang ditampilkan ketika tombol ‘analisis’ pada antar muka aplikasi ditekan. Proses

mendapatkan nilai kapasitas secara umum diperoleh dari langkah yang sama dengan algoritma masing – masingnya seperti yang dijelaskan pada Bab 2 dan Bab 3. Pada algoritma Cheng dan Wang (2007), akibat estimasi bit yang dapat disisipkan pada suatu vertex dilakukan tepat sebelum penyisipan vertex tersebut dan memiliki ketergantungan terhadap perubahan – perubahan yang telah terjadi di sekitar vertex tersebut, kalkulasi untuk ukuran bit maksimal tidak dapat dilakukan dengan akurat. Sehingga dilakukan kalkulasi yang mengasumsikan kondisi terbaik, yaitu setiap vertex dapat menyisipkan jumlah bit maksimal. Lalu untuk algoritma Chao (2009) kalkulasi yang dilakukan akurat akibat jumlah layer yang menentukan kapasitas tetap seperti yang diberikan oleh pengguna. Algoritma Tsai (2012) memiliki estimasi yang tergantung terhadap perubahan posisi vertex yang telah terjadi sehingga kalkulasi tidak dapat dilakukan dengan akurat, sehingga dilakukan kalkulasi berdasarkan kondisi tanpa adanya pergeseran vertex yang dengan kata lain tergolong kondisi terbaik. Untuk algoritma gabungan, kalkulasi dilakukan dengan menggabungkan proses kalkulasi untuk algoritma Chao (2009) dan Tsai (2012) dan keakuratan yang sesuai untuk masing – masing kalkulasi di algoritma tersebut.

### B. Undetectability

Pengujian dilakukan untuk membandingkan besar distorsi yang terjadi pada masing – masing algoritma untuk tiap objek. Distorsi pada pengujian ini dinyatakan dalam RMSE/diagonal bounding box dengan nilai yang lebih kecil menyatakan distorsi yang lebih kecil. Hasil uji dapat dilihat pada tabel V-3

Tabel V-3: Perbandingan Hasil Uji Undetectability dengan satuan RMSE/Diagonal Objek dengan Tujuan Meningkatkan Undetectability Algoritma Gabungan

Objek 3D	Cheng & Wang (2007)	Chao (2009)	Tsai (2012)	Gabungan
pear.obj	$7,82 \times 10^{-4}$	$2,41 \times 10^{-5}$	$4,54 \times 10^{-4}$	$4,96 \times 10^{-8}$
brain.obj	$2,71 \times 10^{-4}$	$9,60 \times 10^{-6}$	$3,81 \times 10^{-4}$	$1,93 \times 10^{-8}$
cow.obj	$7,43 \times 10^{-5}$	$4,21 \times 10^{-6}$	$1,44 \times 10^{-4}$	$2,38 \times 10^{-8}$
hippo.obj	$3,15 \times 10^{-5}$	$2,51 \times 10^{-6}$	$5,68 \times 10^{-5}$	$2,44 \times 10^{-8}$
horse.obj	$8,21 \times 10^{-5}$	$8,71 \times 10^{-6}$	$7,08 \times 10^{-5}$	$1,83 \times 10^{-8}$
maxplanck.obj	$5,99 \times 10^{-5}$	$1,10 \times 10^{-5}$	$6,54 \times 10^{-5}$	$9,78 \times 10^{-9}$

Konfigurasi yang dilakukan layaknya pengujian pada makalah masing – masing algoritma ternyata tidak seluruhnya dapat maksimal di variasi jenis inputan objek pada pengujian ini. Faktor penyebabnya dapat berasal dari ketidakakuratan pembuatan program dalam menyamai algoritma sesungguhnya dan dapat juga disebabkan konfigurasi masukan yang tidak selalu seimbang untuk objek 3D yang berbeda. Seperti hasil implementasi untuk algoritma Chao (2009) memiliki hasil yang sangat baik akibat jumlah layer dan interval yang kecil.

Selanjutnya untuk algoritma gabungan memiliki distorsi yang sangat kecil akibat menumpukkan dua algoritma dengan arah perubahan yang berbeda sehingga meminimalkan distorsi. Selain itu pada penyisipannya digunakan nilai interval yang menyesuaikan dengan nilai threshold sehingga penyisipan

dengan metode layer memberikan perubahan yang sesuai dengan posisi *vertex* terhadap keseluruhan objek 3D.

Pada implementasinya, data *undetectability* pada tabel V-2 diperoleh dari proses kalkulasi yang ditampilkan pada jendela konfirmasi keberhasilan penyisipan pesan. Kalkulasi yang dilakukan adalah menghitung RMSE dari tiap *vertex* yang ada dalam objek 3D, lalu membaginya dengan panjang diagonal *bounding box* objek 3D tersebut..

### C. Akurasi

Pengujian ini dilakukan untuk membandingkan pesan hasil ekstraksi dengan pesan sebelum disembunyikan. Hasil uji dapat dilihat pada tabel V-4.

Tabel V-4: Tabel V 3 : Perbandingan Hasil Uji Akurasi dengan Tujuan Mencapai Akurasi yang Sempurna pada Algoritma Gabungan

Objek 3D	Cheng & Wang (2007)	Chao (2009)	Tsai (2012)	Gabungan
pear.obj	100%	99,7%	100%	100%
brain.obj	98,9%	100%	100%	100%
cow.obj	100%	99,9%	100%	97%
hippo.obj	100%	100%	99,7%	100%
horse.obj	100%	100%	99,2%	99,9%
maxplanck.obj	100%	100%	100%	100%

Dari hasil pengujian ini, tidak seluruh algoritma memiliki akurasi 100%. Seperti pada pengujian *undetectability*, faktor yang dapat menyebabkan hal ini adalah hasil implementasi untuk algoritma – algoritma sumber tidak sepenuhnya sama dengan algoritma sesungguhnya, lalu keterbatasan algoritma dalam mengolah objek tertentu atau terhadap kombinasi masukan konfigurasi tertentu. Seperti pada algoritma Chao (2009), kekurangan algoritma ini adalah pengolahan bilangan pecahan akibat penggunaan operasi pembagian yang berkali lipat. Tipe data yang digunakan pada implementasi adalah double dan walaupun tipe bilangan ini besar, namun tetap memiliki batasan dan batas ini dapat terlewat apabila sebuah bilangan koordinat berkoma dibagi berulang kali. Hasil bagi tersebut tidak dapat mengembalikan nilai yang sama setelah dikalikan faktor pembagiannya sehingga menghasilkan nilai yang berbeda. Ditambah lagi apabila ukuran interval terlalu kecil akan menambah besar ketidakakuratan.

Kelemahan pada algoritma Chao (2009) memberikan dampak terhadap algoritma gabungan akibat penggunaan algoritma Chao (2009) di dalamnya. Sehingga semakin besar layer yang digunakan, semakin besar ketidakakuratan yang mungkin dihasilkan. Pada implementasinya, data akurasi pada tabel V-3 diperoleh dari *log console* yang ditampilkan setelah proses ekstraksi dilakukan. Prosesnya adalah membandingkan setiap *byte* karakter yang berbeda antara pesan rahasia dan hasil ekstraksi.

## VI. KESIMPULAN DAN SARAN

Terdapat beberapa kesimpulan dari pengembangan algoritma ini, yaitu:

1. Kuantitas pesan rahasia yang disembunyikan pada media cover dapat dimaksimalkan dengan memanfaatkan sifat *reversible* dari algoritma Chao (2009).
2. Kemungkinan pendeteksian suatu stego objek dilakukan dapat diminimalkan dengan memanfaatkan sifat *reversible* dari algoritma Chao (2009) juga, yaitu dengan menumpuk dua algoritma penyisipan namun dengan arah pergeseran yang berlawanan.
3. Hasil ekstraksi tetap akurat dan sama dengan pesan rahasia sebelum penyembunyian dapat dicapai dengan memastikan penggunaan tipe data yang tepat dan mampu menerima operasi pembagian pecahan – pecahan dengan akurat, terutama pada bilangan yang sangat kecil. Pada algoritma gabungan ini, dibutuhkan tipe data yang lebih besar daripada double akibat objek 3D yang banyak digunakan memiliki ukuran koordinat berkoma. Selain itu konfigurasi penyisipan juga harus diinput dengan pertimbangan ukuran data terkecil, sehingga tidak menimbulkan pembulatan pecahan yang jauh berbeda dengan nilai pecahan sesungguhnya.

Selain itu terdapat beberapa saran untuk pengembangan lebih lanjut, yaitu:

1. Dengan penyempurnaan sifat *reversible* pada algoritma Chao (2009) melalui penggunaan tipe data pecahan yang besar, lalu disertai pembelikan arah pergeseran seperti yang ada pada algoritma gabungan, banyak kombinasi steganografi pada objek 3D yang dapat dilakukan.
2. Optimalisasi pengkombinasian algoritma steganografi dapat dilakukan dengan memilih algoritma yang besar pergeserannya mirip dengan algoritma Chao(2009) perubahan posisi *vertex* minimal.

## REFERENSI

- [1] Cheng, Y.M., Wang, C.M. (2006). A high-capacity steganographic approach for 3D polygonal meshes. *Vis Comput* 22:845 -855.
- [2] Cheng, Y.M., Wang, C.M. (2007). An adaptive steganographic algorithm for 3d polygonal meshes. *Vis Comput* 23:721-732 .
- [3] Chao, M.W., Lin, C.H., Yu, C.W., Lee, T.Y. (2009). A high capacity 3D steganography Algorithm. *IEEE Transaction on Visualization & Computer Graphics* Vol 20 No. 3.
- [4] Tsai, Y.Y. (2012). An adaptive steganographic algorithm for 3D polygonal models using *vertex* decimation. *Multimedia Tools Appl* 69:859-876.
- [5] Wu, H.T., Dugelay, J.L. (2008). Steganography in 3D geometries and Images by adjacent bin mapping. *EURASIP Journal on Information Security* Volume 2009 Article ID 317165.
- [6] Cox, I.J., Miller, M.L., Bloom, J.A., Fridrich J, Kalker T. (2008). *Digital watermarking and steganography*, 2nd ed. Morgan Kaufmann, Burlington.
- [7] Dugelay, J.L., Baskurt, A., Daoudi, M. (2008). *3D Object processing : compression, indexing, and watermarking*. John Wiley & Sons, Chichester.
- [8] Cayre, F., Macq, B. (2003). Data hiding on 3-D triangle meshes. *IEEE Trans, Signal Process* 51(4) : 939-949.
- [9] Chopine, A. (2011). *3D art essentials*. Elsevier, Burlington.
- [10] Derakhshani, R.L., Derakhshani, R.L. (2003). *Autodesk 3DS Max 2014 essentials*. John Wiley & Sons, Indiana.
- [11] Cheng, Y.M., Wang, C.M. (2005). An Efficient Information Hiding Algorithm for Polygon Models. *Eurographics 2005* Vol.24-3: 591 -6