TUGAS AKHIR

STUDI DAN EKSPLORASI CGAL

Reynard Rafferty Susilo

NPM: 2017730036

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2024

FINAL PROJECT

A STUDY AND EXPLORATION OF CGAL

Reynard Rafferty Susilo

NPM: 2017730036

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2024

LEMBAR PENGESAHAN

STUDI DAN EKSPLORASI CGAL

Reynard Rafferty Susilo

NPM: 2017730036

Bandung, 27 Juni 2024

Menyetujui,

Pembimbing

Digitally signed by Lionov

Lionov, Ph.D.

Ketua Tim Penguji

Digitally signed by Maria V. Claudia M.

Maria Veronica, M.T.

Anggota Tim Penguji

Digitally signed by Elisati Hulu

Elisati Hulu, M.T.

Mengetahui,

Ketua Program Studi

Digitally signed by Lionov

Lionov, Ph.D.

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa tugas akhir dengan judul:

STUDI DAN EKSPLORASI CGAL

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 27 Juni 2024

Reynard Rafferty Susilo
NPM: 2017730036

ABSTRAK

Komputasi geometris adalah studi yang mempelajari algoritma dan struktur data mengenai objek geometri, yang bertujuan untuk menyelesaikan permasalahan geometri. Komputasi geometris memiliki peran yang penting dalam berbagai macam teknologi maju pada saat ini. Banyak aplikasi modern yang memanfaatkan komputasi geometris, seperti grafika komputer, *Geographic Information System* (GIS), robotik, dan pencitraan medis.

Namun, terdapat banyak rintangan yang ditemukan jika algoritma dan struktur data komputasi geometris diimplementasikan dari awal. Salah satu rintangan yang sering dihadapi dalam proses pengimplementasian adalah presisi. Untuk mendapatkan solusi permasalahan geometri yang benar, implementasi harus memiliki presisi yang tinggi(atau cukup tinggi untuk menghasilkan output yang diinginkan). Di samping presisi, algoritma yang diimplementasi juga harus bisa mengatasi *degenerate cases* pada komputasi geometris. Selain memiliki beberapa rintangan, beberapa algoritma dan struktur komputasi geometris juga merupakan bagian dari algoritma komputasi geometris lainnya. Oleh dari itu, untuk mempermudah menyelesaikan pemasalahan geometri, penggunaan *library* yang terpercaya sangat dianjurkan. Salah satu *library* komputasi geometris adalah CGAL.

Computational Geometry Algorithms Library (CGAL) adalah sebuah proyek *open source* berbentuk C++ library, yang menyediakan berbagai algoritma dan stuktur data, untuk membantu menyelesaikan permasalahan komputasi geometris. Agar dapat memanfaatkan CGAL dengan baik, studi dan eksplorasi terhadap CGAL perlu dilakukan. Terdapat beberapa prasyarat yang dianjurkan sebelum studi dan eksplorasi dilakukan. Karena CGAL merupakan *library* bahasa pemgrograman C++ untuk permasalahan komputasi geometris, maka sangat disarankan untuk mengetahui dasar-dasar C++ dan komputasi geometris sebelum melakukan studi dan eksplorasi.

Pada penelitian ini, telah dilakukan studi dan eksplorasi dengan cara membangun dua buah program yang memanfaatkan CGAL. Program pertama adalah implementasi algoritma *divide and conquer convex hull*, untuk mencari *convex hull*. *Convex hull* adalah *convex set* terkecil yang melingkupi kumpulan titik. Sedangkan program kedua yang dibangun adalah *surface reconstruction from point clouds* (SRPC). SRPC adalah pembangunan model permukaan objek tiga dimensi yang didefinisikan oleh sekumpulan titik. Kedua program yang dibangun memanfaatkan struktur data dan algoritma yang didefinisikan oleh CGAL. Dengan membangun program yang memanfaatkan CGAL, fitur-fitur CGAL yang tersedia dapat dipelajari dan diketahui secara sekaligus. Karena untuk mempelajari sesuatu dalam *programming*, cara yang direkomendasikan pada umumnya adalah mengimplementasikan hal tersebut.

Setelah studi dan eksplorasi dilakukan, CGAL benar menyediakan fitur-fitur yang penting (umum digunakan pada implementasi permasalahan komputasi geometris). Klaim ini didasarkan oleh keberhasilan pembangunan kedua program yang telah dilakukan. CGAL berperan berat dalam pengimplementasian permasalahan-permasalahan geometri tersebut. Proses pembangunan program menjadi jauh lebih sederhana berkat pemanfaatan fitur-fitur CGAL. Selain kedua permasalahan yang diimplementasikan, CGAL juga menyediakan banyak algoritma untuk menyelesaikan permasalahan-permasalahan komputasi geometris lainnya yang relatif kompleks.

Kata-kata kunci: CGAL, Komputasi Geometris, C++, *Convex Hull*, *Surface Reconstruction from Point Clouds*

ABSTRACT

Computational geometry is the study of algorithms and data structures concerning geometric objects, aimed at solving geometric problems. Computational geometry plays a crucial role in the current advancing technologies. Many modern applications utilize computational geometry, such as computer graphics, Geographic Information System (GIS), robotics, and medical imaging.

However, there are many obstacles encountered if computational geometry algorithms and data structures are implemented from scratch. One of the common obstacle is precision. To obtain the correct solution of computational geometry problem, the implementation must have high precision (or sufficient precision to produce the desired output). Besides precision, the implemented algorithms must also be able to handle degenerate cases in computational geometry. In addition to having several obstacles, some computational geometry algorithms and structures are also parts of other computational geometry algorithms. Therefore, to simplify solving geometric problems, using a reliable library is highly recommended. One example of computational geometry library is CGAL.

Computational Geometry Algorithms Library (CGAL) is an open-source project in the form of a C++ library that provides various algorithms and data structures to help solving computational geometry problems. To be able to utilize CGAL, A study and exploration of CGAL need to be conducted. There are several prerequisites that are encouraged to be fulfilled before conducting the study and exploration. Since CGAL is a C++ library for computational geometry purpose, it is highly recommended to have basic understanding of C++ and computational geometry beforehand.

In this research, a study and exploration have been conducted by building two programs that utilize CGAL. The first program is the implementation of the divide and conquer convex hull algorithm, for finding a convex hull. Convex hull is the smallest convex set of given points. The second program is surface reconstruction from point clouds(SRPC). SRPC is the process of constructing surface model of a three-dimensional object which is defined by a set of points. Both programs utilize data structures and algorithms that are defined by CGAL. By developing programs that use CGAL, the available features of CGAL can be learned and understood simultaneously. This is because the recommended way to learn something in programming is generally to implement it.

After the study and exploration had been conducted, it was right that CGAL did indeed provide essential computational geometry features (commonly used in implementing computational geometry problems). This claim is based on the successful development of the two programs. CGAL played a significant role in the implementation of these geometric problems. The program development process became much simpler thanks to the utilization of CGAL features. Besides the two implemented problems, CGAL also offers many algorithms to solve other relatively complex computational geometry problems.

Keywords: CGAL, Computational Geometry, C++, Convex Hull, Surface Reconstruction from Point Clouds

Dipersembahkan kepada kedua orang tua dan kakak adik saya yang kasihnya tidak terhingga.

KATA PENGANTAR

Puji syukur dipanjatkan kepada Tuhan atas seluruh berkat, anugrah, dan mukjizat-Nya yang diberikan kepada penulis, sehingga penulis bisa menyelesaikan tugas akhir dengan judul "Studi dan Eksplorasi CGAL". Tugas akhir ini juga ditulis untuk memenuhi prasyarat kelulusan Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Sains, Universitas Katolik Parahyangan Bandung.

Penulis juga mendapatkan bantuan dan dukungan dari berbagai pihak, sehingga tugas akhir ini bisa selesai ditulis. Oleh karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya terhadap pihak yang bersangkutan, antara lain:

1. Keluarga penulis, yaitu mamih-papih berserta kakak-adik penulis, yang sudah bersusah payah memenuhi kebutuhan penulis, menemani, dan memberikan dukungan yang tidak dapat dibalaskan kepada penulis pada masa pengerjaan tugas akhir dan di masa penulis sedang sakit.
2. Dosen pembimbing penulis, Lionov, Ph.D., yang dengan sangat sabar dan cerdas telah memberikan bimbingan dan bantuan kepada penulis dari awal hingga akhir, serta memberikan ilmu-ilmu yang eksklusif.
3. Para dosen penguji, Maria Veronica, M.T.dan Elisati Hulu, M.T.yang memberikan berbagai saran dan perbaikan sehingga penulisan tugas akhir dapat menjadi lebih baik lagi.
4. Ibu Luciana, Ibu Mariskha, Ibu Vania, dan tim administrasi UNPAR, yang dengan berbaik hati telah memberikan keringanan dan membantu penulis di masa sakit.
5. Kepada seluruh dosen UNPAR, yang sudah memberikan pengetahuan akademik dan non-akademik kepada penulis, serta Pak Husnul, Ibu Joanna, dan tim Wombatify(Cristopher dan Henrico) yang menginspirasi penulis untuk berjuang di luar batas.
6. Kepada GRBLG, khususnya Steren, Reinalta, Dio, David, Rio, Fritz, Wang, Juan, Melody, yang telah menemani, menghibur, dan memberikan dukungan kepada penulis pada masa kuliah dan penulisan tugas akhir.
7. Kepada *CS50's instructor*, David J. Malan, yang telah membangkitkan *passion* penulis yang telah hilang.
8. Dan kepada seluruh pihak yang penulis tidak mampu sebutkan selengkapnya, yang telah membantu atau memberi dukungan kepada penulis sehingga penulisan tugas akhir ini bisa selesai.

Penulis menyadari bahwa penulisan tugas akhir dan penelitian ini masih jauh dari sempurna. Oleh karena itu, penulis meminta maaf, dan dengan berlapang dada akan menerima saran dan kritik membangun yang diberikan. Penulis juga berharap supaya penulisan ini dapat berguna bagi yang membaca atau membutuhkan.

Bandung, Juni 2024

Penulis

DAFTAR ISI

DAFTAR GAMBAR

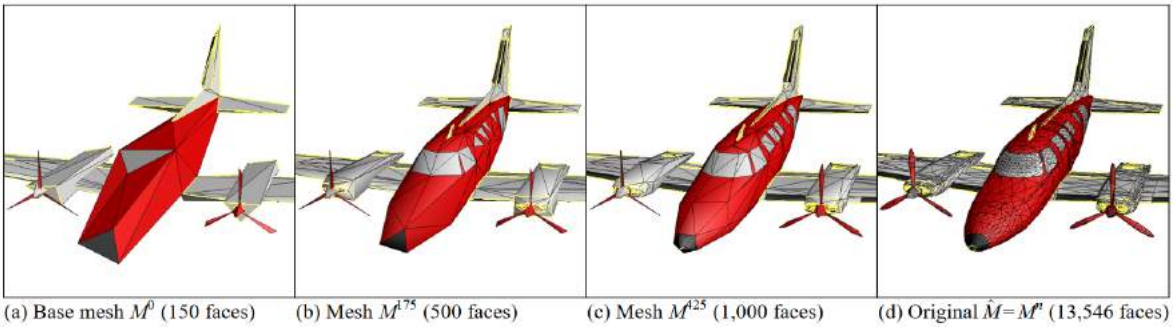
DAFTAR KODE PROGRAM

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Komputasi geometris adalah studi yang mempelajari algoritma dan struktur data mengenai objek geometri, yang bertujuan untuk menyelesaikan permasalahan geometri[1]. Komputasi geometris memiliki peran yang penting dalam aplikasi komputer pada zaman modern. Contoh aplikasi komputer modern yang memanfaatkan komputasi geometris antara lain: grafika komputer, *object rendering and simplification* pada video game, *Geographical Information System* (GIS), *medical imaging*, dan lain-lain. Berikut Gambar 1.1 yang menunjukkan *object simplification* pada video game.



Gambar 1.1: Ilustrasi *object simplification* pesawat pada video game.[1]

Object simplification atau juga dikenal dengan istilah *level of detail*(LOD) dalam video game, merupakan teknik untuk mengatur kompleksitas sebuah model berdasarkan *point of view*(POV). Pengubahan kompleksitas objek dilakukan dengan cara merubah jumlah poligon yang menyusun sebuah objek. Teknik ini dapat meringankan tenaga komputer yang digunakan untuk *rendering* ojek, sehingga pemain memiliki pengalaman yang lebih baik (tidak memiliki peforma yang relatif rendah). *Object simplification* umumnya dilakukan relatif terhadap POV pemain untuk meminimalisir tampak objek yang buruk dari perspektif pemain.

Pada GIS, komputasi geometris dapat dimanfaatkan untuk menghitung luas suatu daerah di peta. Berikut Gambar 1.2 yang menunjukkan penghitungan luas Gedung 10 Universitas Parahyangan (UNPAR) menggunakan *Google Earth*, dimana luas Gedung 10 adalah $2,038.71m^2$. Jika diperhatikan pada Gambar 1.2, bentuk gedung tersebut bukanlah persegi panjang sempurna. Hal tersebut membuat perhitungan luas gedung menjadi lebih kompleks dibandingkan dengan menghitung luas persegi panjang. Salah satu cara untuk menghitung luas tersebut adalah dengan membagi daerah yang dibentuk dengan urutan titik(poligon) menjadi sekumpulan segitiga(kemungkinan segitiga sembarang), dan menghitung total semua luas segitiga yang dihitung satu-persatu menggunakan algoritma komputasi geometris.

[1]gambar diambil dari Hoppe, H. (1996) *Progressive meshes. Microsoft Research*

Gambar 1.2: Menggunakan *Google Earth* untuk menghitung luas gedung 10 UNPAR[2]

Di samping memahami teori komputasi geometris, pengimplementasian komputasi geometris pada komputer bukanlah hal yang sederhana. Salah satu permasalahan yang sering ditemukan pada tahap implementasi adalah presisi. Untuk mendapatkan solusi permasalahan komputasi geometris yang benar, presisi program yang dibangun harus bersifat cukup(memiliki akurasi yang cukup tinggi untuk mendapatkan hasil yang diinginkan). *Floating point precision* merupakan permasalahan umum yang ditemukan permasalahan presisi. *Floating point precision* adalah akurasi yang dimiliki oleh penyimpanan atau operasi bilangan riil pada komputer. Contoh permasalahan *floating point precision* adalah perbandingan nilai desimal yang salah dapat menyebabkan solusi yang salah (contohnya operasi geometri yang salah). Maka dari itu, peimplementasian komputasi geometris dapat memanfaatkan *library* yang sudah ada. Menggunakan sebuah *library* dalam membangun sebuah program memiliki beberapa manfaat, antara lain[3]:

- Menghemat tenaga. Hal ini karena pengimplementasian suatu algoritma atau struktur data tidak perlu dilakukan dari dasar, sedangkan hanya perlu menggunakan algoritma atau struktur data yang sesuai dari *library*.
- Lebih teroptimisasi. *Library* yang baik umumnya memiliki peforma tinggi dan rentan *error* yang rendah karena sudah menjalani berbagai perbaikan, khususnya *open source libray*. Sifat *open source* terhadap suatu *library* memungkinkan berbagai kontributor untuk membantu memperbaiki *library*.
- Memiliki fitur lengkap. *Library* yang baik pada umumnya menyediakan fitur-fitur yang tidak disangka apabila fitur tersebut masuk akal untuk dimasukkan terhadap *library*. Dokumentasi *library* yang baik juga memungkinkan pengguna *library* untuk menemukan fitur-fitur yang tidak diduga.

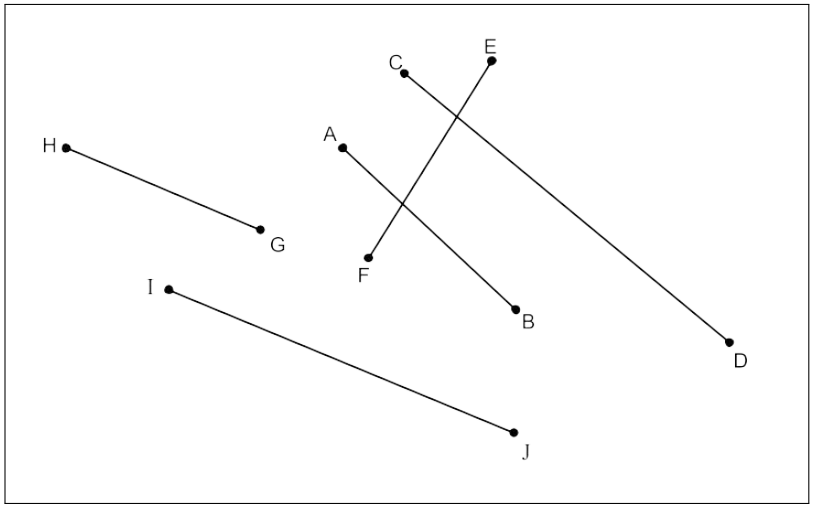
Salah satu *library* komputasi geometris adalah CGAL. *Computational Geometry Algorithms Library* (CGAL) adalah proyek *open source* berbentuk C++ *library* yang menyediakan akses mudah terhadap algoritma geometri[4]. Dengan memanfaatkan CGAL, pengguna bisa mengatasi permasalahan seperti presisi. Selain itu, CGAL juga menyediakan berbagai operasi geometri dan algoritma yang sudah diimplementasikan untuk menyelesaikan berbagai permasalahan komputasi geometris. Contoh operasi geometri yang sederhana antara lain: menentukan apakah 2 segmen

[2]gambar diambil dari *Google Earth*

[3]diambil dari (https://softwareengineering.stackexchange.com/questions/29513/is-reinventing-the-wheel-really-all-that-bad)

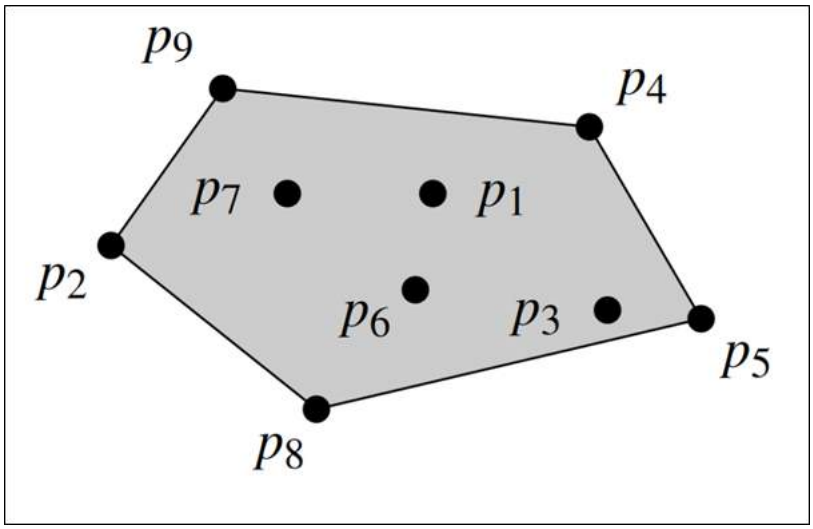
[4]deksripsi diambil dari website *official* CGAL (https://www.cgal.org/)

garis berpotongan (*segments intersection*), atau terdapat algoritma yang dapat mencari *convex hull*(pencarian poligon terkecil terhadap sekumpulan titik yang disediakan). Berikut Gambar 1.3 yang menunjukkan contoh *segments intersection*, dimana terdapat segmen garis AB dan EF, serta CD dan EF yang berpotongan.



Gambar 1.3: Menentukan apakah dua segmen garis berpotongan

Sedangkan Gambar 1.4 menunjukkan contoh *convex hull*. Terdapat 9 titik yang dimasukkan pada gambar $(p_1, p_2, p_3, ...p_9)$. *Convex hull* yang didapatkan dari titik yang dimasukkan adalah $(p_2, p_8, p_5, p_4, p_9)$.

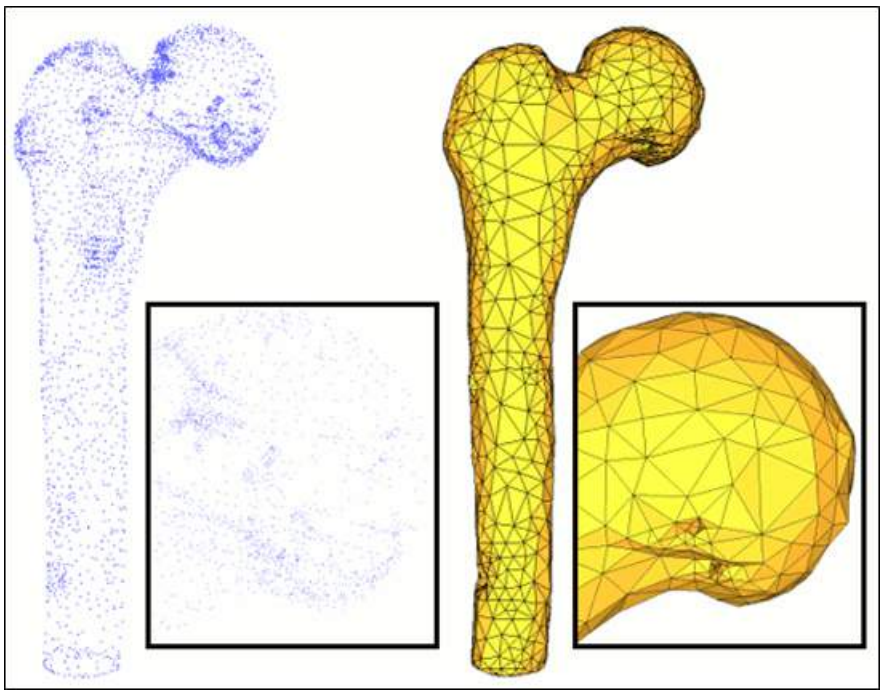


Gambar 1.4: Mencari *convex hull* dari sekumpulan titik.[5]

Tentu saja algoritma-algoritma untuk menyelesaikan permasalahan seperti *convex hull* dan *segments intersection* dapat diimplementasikan secara mandiri. Tetapi selain membuang tenaga, banyak juga optimisasi yang harus dilakukan seperti memastikan bahwa algoritma yang diimplementasikan memiliki peforma yang wajar untuk mengatasi permasalahan dengan input yang besar, atau algoritma yang diimplementasikan juga harus bisa mengatasi permasalahan-permasalahan seperti presisi. Melakukan implementasi secara mandiri juga akan bertambah sulit apabila permasalahan geometri yang ingin diselesaikan lebih kompleks. Berikut Gambar 1.5 yang merupakan salah satu

[5]gambar diambil dari [2]

permasalahan komputasi geometris, yaitu *surface reconstruction from point clouds*. *Surface reconstruction from point clouds* merupakan proses pembangunan permukaan model geometri, terhadap sebuah objek yang didefinisikan oleh sekumpulan titik [3]. Permasalahan tersebut merupakan permasalahan yang bersifat cukup kompleks. Oleh karena itu, pengimplementasian komputasi geometris dapat memanfaatkan CGAL demi mempermudah pengerjaan.



Gambar 1.5: *Surface reconstruction from point clouds*

Sebelum menggunakan CGAL untuk mengimplementasikan algoritma-algoritma untuk menyelesaikan permasalahan geometri, terdapat beberapa langkah atau prasyarat yang dianjurkan untuk dipenuhi. Dengan memenuhi langkah-langkah tersebut, akan terbangun fondasi yang diperlukan untuk memanfaatkan dan membangun program menggunakan CGAL.

Langkah pertama yang dilakukan adalah mempelajari komputasi geometris. Karena CGAL adalah *library* komputasi geometris, maka mempelajari komputasi geometris memiliki beberapa manfaat. Dengan mempelajari komputasi geometris, maka fitur-fitur yang disediakan oleh CGAL bisa dipahami dengan lebih baik. Pemahaman yang baik dapat membantu untuk menentukan fitur-fitur mana yang relevan untuk digunakan pada program yang akan dibangun. Studi komputasi geometris dilakukan dengan cara mempelajari komputasi geometris dari beberapa sumber, dan menyelesaikan contoh-contoh permasalahan komputasi geometris.

Langkah kedua yang diperlukan untuk membangun program CGAL yaitu mempelajari bahasa pemrograman C++. Karena CGAL merupakan *library* C++, maka sangat dianjurkan untuk mempelajari dasar-dasar C++ untuk memiliki fundamental yang baik dalam memahami CGAL. Dengan mempelajari C++, pemanfaatan CGAL bisa menjadi lebih maksimal. Selain itu, mempelajari C++ juga diperlukan untuk mengintegrasikan berbagai algoritma dan struktur data CGAL terhadap kode program yang dibangun secara mandiri. Di samping mempelajari C++ pada langkah kedua ini, akan dipelajari juga bagaimana cara untuk membangun (*build*) sebuah program yang memiliki dependensi CGAL (termasuk *library* yang dimanfaatkan CGAL sendiri).

Pada penelitian ini, telah dilakukan studi dan eksplorasi terhadap CGAL. Studi dan eksplorasi akan dilakukan dengan cara membangun program-program yang memanfaatkan CGAL. Dengan membangun beberapa program yang memanfaatkan CGAL, pengguna CGAL otomatis akan merasa lebih terbiasa untuk melakukan integrasi program atau fitur CGAL dengan program pengguna,

[5]gambar diambil dari https://www.cgal.org/

untuk membangun solusi yang ingin dicapai. Kesesuaian program-program dipilih dibatasi oleh beberapa faktor, yaitu:

- *Scope*. Fokus penelitian ini adalah studi dan eksplorisasi CGAL, maka beban program akan berada pada komputasi geometris, bukan di bidang informatika lainnya.
- *Resource*. Program yang dibangun tidak memerlukan *resource* dengan kemampuan sangat tinggi untuk menjalankan *software* yang dibangun.

Program pertama adalah pengimplementasian *convex hull*, dengan menggunakan metode *divide and conquer*, dimana pengimplementasian tersebut menggunakan *geometric kernel* CGAL. *Convex hull* seperti pada 1.4, adalah *convex set* terkecil yang bisa dibangun, sehingga menyeliputi seluruh titik yang diberikan. *convex set* sendiri merupakan urutan titik, dimana untuk segala kombinasi 2 titik di dalam set tersebut, jika direpresentasikan dengan segmen garis, maka segmen garis akan tertampung di dalam. Tujuan program pertama ini bukan menyelesaikan permasalahan *convex hull*, tetapi menunjukkan bahwa CGAL menyediakan fitur *geometric kernel*, dimana fitur tersebut bisa dimanfaatkan sebagai bagaian untuk memecahkan permasalahan komputasi geometris yang ada. *Geometric kernel* berupa objek-objek geometri dan operasi-operasi geometri dasar(*basic*) yang sudah teroptimisasi seperti: titik, *polygon*, operasi penentuan perpotongan, operasi menentukan arah orientasi.

Program kedua yang dibangun merupakan *surface reconstruction from point clouds*. Untuk menyelesaikan permasalahan ini, digunakan teknik *tetrahedra removal* untuk implementasinya. Sedangkan pemodelan tetrahedra yang dibentuk dari sekumpulan titik bisa dimodelkan dengan menggunakan *3D Delaunay Triangulation* CGAL.

Selain pembangunan program berfungsi untuk meningkatkan penguasaan terhadap CGAL, pembangunan program juga mendorong pembangun program untuk menelusuri fitur-fitur CGAL yang ada, sehingga bisa mengetahui lebih jauh kemampuan CGAL. Dengan demikian, pengguna dapat memanfaatkan CGAL dengan lebih baik lagi, yang mana akan berguna untuk keperluan akademik maupun untuk keperluan aplikasi di dunia nyata di masa depan nanti.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas di tugas akhir ini antara lain:

1. Apa saja prasyarat yang diperlukan supaya dapat memanfaatkan CGAL dengan maksimal?
2. Bagaimana cara menggunakan CGAL untuk mengimplementasikan algoritma-algoritma yang dapat menyelesaikan suatu permasalahan komputasi geometris?

1.3 Tujuan

Tujuan tugas akhir ini antara lain:

1. Mempelajari komputasi geometris, C++, dan cara membangun program C++ yang memiliki dependensi CGAL.
2. Memanfaatkan fitur-fitur CGAL yang relevan untuk menyelesaikan permasalahan geometri, dengan cara melakukan studi dan eksplorasi CGAL.

1.4 Batasan Masalah

Penelitian yang dilakukan dibatasi oleh beberapa hal seperti:

1. Penelitian dilakukan dalam jangka waktu terbatas, maka tidak semua fitur CGAL dipelajari secara keseluruhan dan mendalam.
2. Ruang lingkup geometri pada penelitian ini merupakan komputasi geometris diskrit, bukan geometri kontinu.

3. *Environment* penelitian dilakukan pada OS **Windows 11** menggunakan IDE **Microsoft Visual Studio 2022**, sehingga fokus pembahasan *environment* berada di *environment* tersebut.
4. Kompleksitas atau efisiensi perangkat lunak yang dibangun tidak akan dibahas secara mendalam selama masih dalam batas wajar(tidak lebih buruk dari *polynomial time*).

1.5 Metodologi

Metodologi yang digunakan untuk melakukan penelitian adalah sebagai berikut:

1. Melakukan studi literatur mengenai komputasi geometris dan menyelesaikan berbagai contoh permasalahan sederhana.
2. Mempelajari bahasa pemrograman C++.
3. Mempelajari bagaimana cara membangun program yang memiliki dependansi CGAL.
4. Menelusuri dan mempelajari dokumentasi CGAL, untuk mengetahui fitur-fitur CGAL lebih jauh.
5. Melakukan analisis masalah dan membuat perancangan perangkat lunak sesuai program yang dipilih.
6. Membangun program komputasi geometris yang ditentukan dengan memanfaatkan CGAL.
7. Melakukan *testing* dan *debugging* terhadap perangkat lunak yang dibangun.
8. Menganalisis hasil yang dihasilkan oleh perangkat lunak yang dibangun.
9. Menulis dokumen tugas akhir.

1.6 Sistematika Pembahasan

Penelitian ini jika digambarkan secara garis besar maka akan terstruktur sebagai berikut:

1. **Bab 1 Pendahuluan**
   Membahas latar belakang dan motivasi penggunaan CGAL, prasyarat, dan alasan dilakukan pembangunan program. Secara keseluruhan, bab ini membahas latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan.
2. **Bab 2 Landasan Teori**
   Membahas teori komputasi geometris, *convex hull divide and conquer*, *surface reconstruction*. Bab ini juga membahas dasar-dasar, *syntax*, dan *Standard Template Library*(STL) C++, juga dasar-dasar CGAL.
3. **Bab 3 Analisis**
   Membahas bagaimana mengintegrasi teori komputasi geometris, STL C++, dan fitur CGAL untuk menyelesaikan program, dan berisi contoh kasus dan *pseudocode* algoritma program yang dibangun.
4. **Bab 4 Perancangan**
   Membahas perancangan *workflow* untuk membangun sebuah program CGAL, dan membahas *data flow diagram* yang dibuat berdasarkan program yang dibangun.
5. **Bab 5 Implementasi**
   Membahas lingkungan implementasi, instalasi dan pembangunan CGAL, dan detail implementasi program-program yang disertakan dengan kode-kode program.
6. **Bab 6 Pengujian dan Eksperimen**
   Membahas pengujian yang berupa sumber dataset dan solusi yang diperoleh dari program yang dibangun, dan membandingkannya dengan solusi yang dihasilkan CGAL. Terdapat juga eksperimen yang dilakukan untuk memperbaiki program yang dibangun.
7. **Bab 7 Kesimpulan dan Saran**
   Membahas kesimpulan dari keseluruhan penulisan dan saran-saran untuk memperbaiki penelitian yang dilakukan.