Penerapan Algoritma Divide and Conquer pada Algoritma Convex Hull

Jessica Andjani / 13513086
Program MagisterInformatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13513086@std.stei.itb.ac.id

***Abstract*—Pengelompokkan suatu titik – titik dapat membuat suatu pola tertentu. Pembentukkan pola akan terjadi dengan menghubungkan antara titik terluar pada kumpulan titik – titik tersebut. Pada awalnya kelompok dari titik – titik tersebut akan dibagi menjadi dua area dan pada akhirnya titik – titik tersebut akan kembali digabungkan menajdi satu kelopok kembali. Algoritma convex hull akan berperan dalam pembentukan pola ini, namun untuk mempercepat kinerja dalam pembentukan pola 3D dibutuhkan algoritma quickhull yang masih berhubungan dengan quicksort. Kedua algoritma tersebut pun menerapkan algoritma divide and conquer. Aplikasi pembentukan pola 3D ini diterapkan pada printer 3D.**

***Index Terms*—printer 3D, divide and conquer, convex hull, quickhull, quicksort**

I. PENDAHULUAN

Dalam era globalisasi ini, teknologi semakin berkembang pesat. Hampir semua orang di dunia ini membutuhkan teknologi untuk mempermudah semua pekerjaan mereka. Perkembangan teknologi sangat membantu manusia untuk mengerjakan berbagai persoalan dengan efektif dan efisien.

Hampir setiap hari benda – benda elektronik ditemukan disekitar manusia. Hal tersebut bukanlah hal yang tabuh untuk zaman sekarang, sebaliknya benda elektronik tersebut membantu manusia untuk mempersingkat waktu mereka dalam mengerjakan suatu tugas. Misalnya saja untuk mencetak sebuah berkas atau laporan dibutuhkan sebuah printer.

Pada zaman dulu mungkin orang masih merasa kesulitan untuk mencetak suatu berkas. Printer sudah mengalami perubahan-perubahan yang semakin canggih untuk setiap zamannya, berawal dari printer dot matrix, deskjet/inkjet printer, laserjet, thermal printer, hingga printer 3D.

Printer 3D merupakan suatu inovasi baru yang ditemukan pada zaman sekarang. Saat ini orang tidak perlu merasa kesulitan untuk membuat suatu benda yang mereka ingini karena printer 3D mampu mencetak objek atau miniatur tiga dimensi.

Untuk membentuk potongan – potongan pola berbentuk polygon pada sebuah printer 3D, dibutuhkan satu set titik – titik yang akan dibentuk. Pada awalnya dibutuhkan suatu pembagian dan penggabungan area yang akan menggunakan algoritma convex hull yang merepresentasikan algoritma divide and conquer.

II. LANDASAN TEORI

2.1 Divide and Conquer

2.1.1 Sejarah Divide and Conquer

Pada zaman dahulu, divide and conquer merupakan strategi militer yang dikenal dengannama *divide ut imperes*. Saat ini strategi tersebut menjadi strategi fundamental di dalam ilmu komputer dengan nama divide and conquer. Algoritma divide and conquer ini ditemukan oleh seorang ilmuwan Rusia bernama Anatolii Alexeevich Karatsuba pada tahun 1960. Pada mulanya beliau menemukan algoritma yang lebih mangkus untuk mengalikan dua buah bilangan bulat yang besar.

2.1.2 Definisi Divide and Conquer

Dividide and conquer merupakan algoritma yang sangat popular di dunia ilmu komputer. Divide and conquer merupakan algorita yang berprinsip memecah – mecah suatu permasalahan yang terlalu besar menjadi bagian – bagian kecil, sehingga lebih mudah untuk diselesaikan. Langkah – langkah umum algoritma untuk divide and conquer adalah, sebagai berikut :

- Divide
  membagi masalah menjadi beberapa upa-masalah yang memiliki kemiripan dengan masalah semula namun berukuran lebih kecil (idealnya berukuran hampir sama).
- Conquer
  memecahkan atau menyelesaikan masing – masing upa-masalah secara rekursif
- Combine
  Menggabungkan solusi masing – masing upa-masalah sehingga membentuk solusi masalah semula.

Berikut ini merupakan skema umum dari algoritma divide and conquer :

```
prosedur divideConquer (input n : integer) {
  Input : masukan berupa integer yang
  berukuran n
  Output : solusi dari masalah semula
}

Kamus
  r, k : integer

Algoritma
  If (n < n0 then {masalah sudah cukup kecil }
    Solve sub-masalah yang berukuran n ini
  else
    bagi menjadi r sub-masalah, masing -
    masing berukuran n/k
    for masing – masing r upa-masalah do
      divideConquer(n/k)
    endfor
    Combine solusi dari r sub-masalah menjadi
    solusi masalah semula
  endif
```

2.2 Quicksort

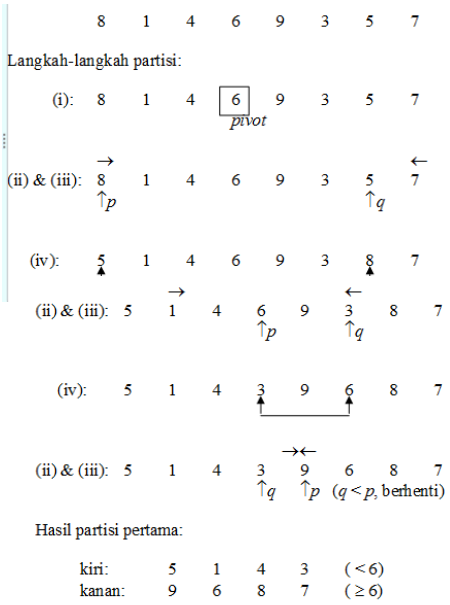
2.2.1 Definisi Quick Sort

Quicksort merupakan algoritma sorting yang dikembangkan oleh Tony Hoare pada tahun 1962. Algoritma ini juga dikenal sebagai *partition-exchange sort (sorting pembagian pembagi).* Algoritma quicksort termasuk dalam algoritma yang sulit membagi, mudah menggabung (*hard split / easy join*)

Teknik mempartisi tabel adalah, sebagai berikut :

- pilih $x \in \{ A1, A2, ..., An\}$ sebagai elemen *pivot*
- pindai tabel dari kiri sampai ditemukan elemen $Ap \geq x$
- pindai tabel dari kanan sampai ditemukan elemen $Aq \leq x$
- pertukarkan $Ap \Leftrightarrow Aq$
- ulangi point kedua, dari posisi $p + 1$, dan point ketiga, dari posisi $q - 1$ sampai kedua pemindaian bertemu di tengah tabel

Cara pemilihan pivot dapat dilakukan dengan tiga cara. Pertama pivot sebagai elemen pertama atau elemen terakhir atau elemen tengah tabel. Kedua, pivot dipilih secara acak dari salah satu elemen tersebut. Ketiga pivot sebagai elemen median tabel.



Gambar 2.2.1 Contoh paritisi tabel

2.2.2 Kompleksitas Algoritma

2.2.2.1 Kasus terbaik

Kasus terbaik dari algoritma ini terjadi saat pivot adalah elemen median sedemikian sehingga kedua upatabel berukuran relatif sama seiap kali pemartisian.

$T(n)$: a , $n > 1$

$2T(n/2) + cn$, $n > 1$

$O(n^2 \log n)$

2.2.2.1 Kasus terburuk

Kasus terburuk dari algoritam quicksort terjadi pada saat setiap partisi pivot selalu elemen maksimum atau elemen minimum tabel.

$T(n)$: a , $n = 1$

$2T(n-1) + cn$, $n > 1$

$O(n^2)$

2.2.2.3 Kasus rata-rata (average case)

Kasus ini terjadi jika pivot dipilih secara acak dari elemen tabel dan pengulang setiap elemen dipilih menjadi pivot adalah sama.

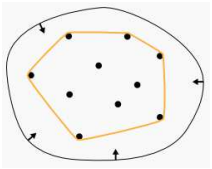
$Tavg(n) : O(n^2 \log n)$

2.3 Convex Hull

Convex hull adalah polygon yang disusun dari subset titik sedemikian sehingga tidak ada titik dari himpunan awal yang berada di luar polygon tersebut (semua titik berada di batas luar atau di dalam area yang dilingkupi oleh poligon tersebut).

Convex hull merupakan persoalan klasik dalam geometri komputansional. Untuk melakukan penyelesaian permasalahan algoritma ini digunakan algoritma divide and conquer. Konsep dari algoritma

ini adalah membagi area menjadi dua bagian, lalu menyelesaikan bagian atas dan bagian bawah dari area tersebut. Setelah itu dilakukan penggabungan dengan membentuk suatu pola dari titik – titik terluar.



Gambar 2.3.1 Convex Hull 2D

2.4 Quickhull

Quickhull merupakan algoritma yang melakukan perhitungan pada algoritma convex hull, namun untuk permasalahan kumpulan titik – titik yang tak berhingga, quickhull lebih mangkus dibandingkan dengan convex hull. Quickhull merupakan mirip pula dengan algoritma quicksort. Pada rata – rata kasus, quickhull lebih mangkus dibandingkan dengan algoritma convex hull,. Komplesitas rata – rata dari quickhull sama dengan quicksort, yaitu $O(n^2 \log n)$ dan pada kasus terburuk adalah $O(n^2)$.

2.5 3D-Printer

2.5.1 Sejarah 3D – Printer

Pada awalny 3D-printing lebih dikenal dengan *Rapid Prototyping*, yaitu sebuah proses dimana seorang engineer akan mendesain file CAD (Computer Aided Design) lalu mengirimnya ke mesin untuk membuat benda tersebut menjadi nyata. Namun, sangat disayangkan material yang digunakan tidak cukup kuat untuk disebut sebagai sebuah produk, lebih tepat disebut sebagai sebuah model untuk menggambarkan bentuk nyatanya.

3D-printing mulai berhasil ketika sebuah perusahaan desain 3D menemukan material baru yang disebut nanocomposite. Nanocomposite merupakan gabungan dari berbagai material plastik dan besi. Pada tahun 1986, Charles W.Hull memiliki hak paten atas 3D-printing ini.

2.5.2 Cara Kerja 3D-Printing

Berikut ini adalah teknik – teknik yang digunakan 3D printer :

- Stereolithography (SLA) adalah teknik pertama untuk 3D-printing. Cara kerjanya adalah menambahkan layer terus menerus pada bahan photopolymer menuju keatas. Material yang digunakan pada awalnya adalah liquid (cairan) dan akan mengeras ketika liquid tersebut terkena sinar ultraviolet.
- Digital Light Processing (DLP) adalah teknik yang hampir sama dengan SLA. Namun pada saat penyinaran digital, objek yang pada awalnya berbentuk liquid akan mengeras setengah dan objek akan tenggelam kebawah dan menaikan liquid sebagiannya lagi. Proses akan berlanjut hingga objek 3D berhasil dibentuk.
- Selective Laser Sintering (SLS) menggunakan tenaga yang sangat tinggi untuk menggabungkan berbagai material, seperti plastik, gelas, keramik, dan metal menjadi sebuah objek 3D
- Electron Beam Melting (EBM) merupakan proses 3D-printing untuk bahan metal.
- Multi Jet Modelling (MJM) mempunyai cara kerja yang serupa dengan inkjet printer. Ia menyebarkan sebuah layer dari resin powder dan menyemprotkan sebuah lem yang mempunyai berbagai warna dan akan mengeras pada suatu layer.
- Fused Deposition Modelling (FDM) menggunakan bahan nozzle yang dipanaskan dan akan melelehkan bahan seperti plastik pada hasil akhirnya.
- Semua aktivitas 3D printing kebanyakan akan menggunakan STL File. STL merupakan format 3D modeling yang membuat 3D printer melakukan tugasnya dengan efektif untuk memotong objek dari layer pada saat print. Kebanya file STL dibuat oleh CAD.

Gambar 2.5.2.1 3D Printer

III. ISI

Pada dasar teori dijelaskan carak kerja 3D printing membutuhkan suatu STL file. STF file akan diubah menjadi G-Code.

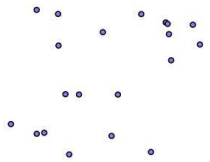
Untuk membentuk pola dibutuhkan pula CSG (Constructive Solid Geometry) pada 3D printing yang berguna untuk merepresentasikan kumpulan titik – titik yang akan diolah pada sebuah program. Untuk menyelesaikan permasalahan

tersebut dibutuhkan algoritma convex hull, namaun untuk membuatnya menjadi lebih manggkus algoritma quickhull dipilih menjadi salah satu solusi untuk permasalahan ini.

Seperti penjelasan pada bab sebelumnya quick merupakan convex hull dengan kumpulan titik – titik yang tak berhingga. Pemecahan masalah convex hull ini diselesaikan dengan menggunakan algoritma divide and conquer. Algoritma ini akan menggunakan rekursif untuk mencapai solusi. Pada prinsipnya, algoritma covex hull dan quickhull akan mencari pola dengan mengambil titik terluar.

Langkah – langkah untuk melakukan penyelesaian pada algoritma convex hull dengan menggunakan algoritma divide and conquer adalah sebagai berikut :

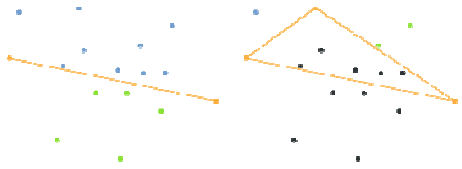
- Melakukan pengurutan pada titik – titik dari himpunan S yang diberikan berdasarkan koordinat absis – X



Gambar 3.1 Himpunan set titik – titik

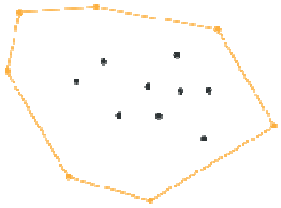
- Jika $|S| \leq 3$, maka lakukan pencarian convex hull secara brute force

- Jika tidak, partisi himpunan titik – titik pada S menjadi dua buah himpunan A dan himpunan B terdiri dari setengah jumlah dari |S| dan titik dengan koordinat absis – X yang terendah dan B terdiri dari setengah dari jumlah |S| dan titik dengan koordinat absis – X terbesar.



Gambar 3.2 Pembagian titik menjadi dua area

- Secara rekursif lakukan perhitungan terhadap H_A = conv(A) dan H_B = conv(B).
- Lakukan penggabungan (merge) terhadap kedua hull tersebut menjadi convex hul, H, dengan menghitung dan mencari upper dan lower tangents untuk H_A dan H_B degan mengabaikan semua titik yang berada diantara dua buah tangent ini.



Gambar 3.3 Penggabungan titik – titik menjadi satu bagian dan membentuk suatu pola

Kompleksitas pada saat pencarian cnvex hull secara brute – force adalah O(1), dimana brute force menandakan basis dari algoritma convex hull. Pada saat pengurutan titik dari himpunan S, kompleksitas yang terjadi adalah O(n logn n).

Berikut adalah algortima dari convex hull :

```
procedure convexHull (input P [1..n] : array of point, output L : List of Point)
{  Menyelesaikan masalah convex hull dengan
   algoritma divide and conquer
   Input : masukan array of point yang
   berukuran n
   Output : solusi dari masalah
}

Deklarasi
   r : integer
   la : list of point

Algoritma
   L = {}
   if  ( n ≤ 30) then
     {ukuran masalah sudah cukup kecil }
      SOLVE upa-masalah dengan metode brute-
      force
  else
      Bagi menjadi r upa-masalah, masing-masing
      berukuran n/k
      HA  = P[1..n/2]
      HB  = P[n/2+1..n]
      convexHull (HA)
      convexHull (HB)
     {gabungkan solusi dari r upa-masalah
      menjadi solusi masalah semula }
```

```
    H = prosedur gabungan Ha dan Hb dengan
    mencari lower tangent dan upper tangent

    La = listPoint(H)
    L = L U la
endif
```

Pada algoritma di atas, dapat dilihat bahwa terdapat prosedur untuk mencari lower tangent dan upper tangent. Berikut akan dijelaskan gambaran umum dalam pencarian lower tangent dari H_A dan H_B
LowerTangent (H_A, H_B) :

1. Misalkan a merupakan titik tekanan dari H_A
2. Misalkan b merupakan titik tekanan dari H_B
3. While (ab bukan merupakan lower tangent dari H_A dan H_B) do
   - While (ab bukan merupakan lower tangent dari H_A) do a -> a.predesesor
   - While (ab bukan merupakan lower tangent dari H_B) do b -> b.suksesor
4. Return ab

Berikut algoritma dari fungsi LowerTangent :

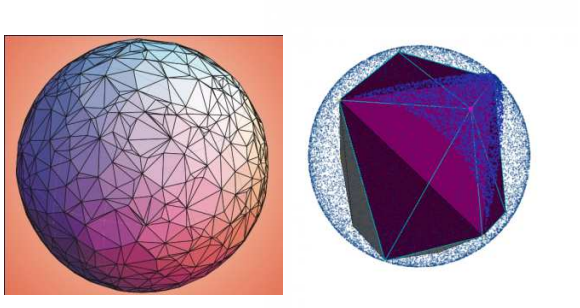
```
function LowerTangent (input HA, HB : list of
point ): list of point
{ mencari lower tangent dari Ha dan Hb }

Algoritma
 a = rightmost point Ha
 b = lefttmost point Hb

while (ab bukan lower tangent dari Ha dan Hb)do
  while (ab bukan lower tangent dari Ha) do
     a.pred()
  while (ab bukan lower tangent dari Ha) do
     b.succ()
```

Kompleksitas waktu dari algoritma yang digunakan di atas dapat dinyatakan dalam relasi rekurens. Diberikan masukan dengan ukuran n, pertimbangan waktu yang dibutuhkan untuk menjalankan semua bagian dari prosedur. Hal ini akan mencakup waktu untuk mempartisi himpunan titik, menghitung kedua tangent, dan mengembalikan hasil akhir. Dari langkah pertama dan ketiga apat dijalankan dalam kompleksitas waktu O(n). Sehingga didapat kompleksitas waktu sebesar O(n log n)

$$T(n) : \begin{cases} 1 & , |S| < 3 \\ n + 2T(n/2) & , \text{untuk n lainnya} \end{cases}$$

Dilihat dari kompleksitas di atas, algoritma convex hull termasuk algoritma yang cukup mangkus digunakan dan algoritma ini dapat digunakan sampai dimensi lebih dari tiga.

Hasil dari Convex hull yang berdimensi tiga :



Gambar 3.4 Convex hull berdimensi tiga

IV. KESIMPULAN

Dapat disimpulkan bahwa algoritma convex hull dapat meggunakan algoritma divide and conquer untuk mencari titik terluar. Terjadi pembagian pada saat satu set titik akan dibagi menjadi dua area yaitu, area atas dan area bawah. Pada setiap area akan dicari titik terluar dari kumpulan titik – titik tersebut. Setelah dua area tersebut menemukan titik terluar, maka kedua area tersebut akan digabungkan menjadi satu dan membentuk suatu pola seperti polygon. Pembentukan pola ini akan membantu dalam proses kerja dari 3D printing. Dengan adanya algoritma ini, 3D printing dapat lebih mudah untuk menemukan pola, selain itu 3D printing menggunakan algoritma quickhull. Algoritma quickhull mangkus pada beberapa kasus sehingga algoritma quickhull pun dipilih untuk menyelesaikan persoalan ini. Quickhull merupakan convex hull dengan kumpulan titik – titik yang tak berhingga.

V. Ucapan Terimakasih

Penulis ingin mengucapkan terima kasih kepada Tuhan Yang Maha Esa karena telah memampukan penulis untuk menyelesaikan makalah ini dengan baik dan tepat waktu. Selain itu, penulis mengucapkan terima kasih juga kepada Bapak Rinaldi dan Ibu Ulfa yang telah mengajarkan dasar-dasar teori yang penulis butuhkan untuk menyelesaikan makalah ini. Tidak hanya bahan untuk makalah ini, penulis juga ingin mengucapkan terima kasih kepada kedua dosen pengajar yang telah membimbing dan menemani selama satu semester ini mempelajari semua pelajaran Strategi Algoritma yang menarik dan telah menambah wawasan dan pengetahuan baru yang bermanfaat.

REFERENSI

http://stackoverflow.com/questions/18416861/how-to-find-convex-hull-in-a-3-dimensional-space Sabtu, 2 Mei 20.00

http://thomasdiewald.com/blog/?p=1888 Sabtu, 2 Mei 21.00

http://stackoverflow.com/questions/18416861/how-to-find-convex-hull-in-a-3-dimensional-space Sabtu, 2 Mei 21.00

http://tansen-ekoteknik.blogspot.com/2010/05/algoritma-divide-and-conquer.html Sabtu, 2 Mei 20.30

http://id.wikipedia.org/wiki/Quicksort Sabtu, 2 Mei 21.09

http://pamularmx.blogspot.com/2013/10/tugas-softskill-new-media.html Sabtu, 2 Mei 22.30

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, 2 Mei 2015

Jessica Andjani / 13513086