

Perbandingan Algoritma Convex Hull: Jarvis March dan Graham Scan

Fenina Adline Twince Tobing¹, Prayogo²

¹ Program Studi Informatika, Fakultas Teknik dan Informatika Universitas Multimedia Nusantara,
Tangerang, Indonesia
fenina.tobing@umn.ac.id

² Program Studi Teknik Informatika, AMIK Mapan, Tangerang, Indonesia
prayogoster@gmail.com

Diterima 02 November 2020

Disetujui 27 November 2020

Abstract—Comparison of algorithms is needed to determine the level of efficiency of an algorithm. The existence of a speed comparison in an algorithm that will make a unique shape is sometimes a problem that must be solved by comparing one algorithm with another. This problem can be solved by using Jarvis March and Graham Scan where this algorithm will create a unique shape of a point followed by testing the speed comparison of the two algorithms and the result can be stated that the Graham Scan algorithm in general works faster than the algorithm Jarvis March.

Index Terms—algorithms, Convex Hull, Graham Scan, Jarvis March

I. PENDAHULUAN

Convex Hull adalah sebuah struktur yang terdapat di segala tempat dalam komputasi geometri yang sangat berguna, juga sangat membantu dalam mengkonstruksi struktur lainnya seperti diagram *Voronoi* serta di dalam pengaplikasian analisis gambar yang tidak diperhatikan [1].

Pada konteks ruang-2 dimensi, *Convex Hull* dapat diibaratkan sebagai karet gelang dan titik-titik diibaratkan sebagai paku yang tertancap dalam permukaan rata. Saat karet gelang diregangkan sampai dapat mencakup seluruh paku lalu dilepas, karet gelang akan merapat sehingga hanya menyentuh titik-titik yang paling jauh dari tengah.

Convex Hull memiliki beberapa fungsi yang di antaranya adalah sebagai pencegah terjadinya tabrakan dan sebagai penganalisa sebuah objek dengan struktur yang bergantung pada perhitungan di dalam *Convex Hull* itu sendiri [1]. *Convex Hull* juga dapat dipakai untuk dekomposisi bentuk, pengindeksan objek, kecerdasan buatan, pengolahan citra, dan simulasi medis [10].

Algoritma merupakan hal yang penting dalam menyelesaikan sebuah permasalahan. Dibutuhkan sebuah metode yang tepat karena dapat mempengaruhi hasil yang diinginkan. Sebaik apapun

algoritma, jika menghasilkan output yang salah, maka algoritma tersebut bukanlah algoritma yang baik [11].

Tujuan penelitian ini adalah mengadakan perbandingan kecepatan terhadap algoritma *Convex Hull* yaitu *Jarvis March* dan *Graham Scan* di mana nantinya akan diketahui kelebihan dan kekurangan dari masing-masing algoritma.

II. LANDASAN TEORI

Kedua fungsi *Convex Hull* memakai fungsi orientasi, dimana fungsi tersebut mengecek apabila tiga titik berurutan akan membuat garis yang akan membelok searah / berlawanan jarum jam, atau bahkan lurus. Orientasi ini didapatkan menggunakan *vector cross product*. Rumus dari *vector cross product* adalah $v1.y \times v2.x - v1.x \times v2.y$, dimana p , q , dan r adalah tiga titik berurutan, $v1$ adalah vektor selisih q dengan p , dan $v2$ adalah vektor selisih r dengan q .*

Bila orientasi bernilai 0, ketiga titik tersebut segaris atau lurus. Bila orientasi bernilai lebih dari 0, ketiga titik tersebut membelok searah jarum jam. Bila orientasi bernilai kurang dari 0, ketiga titik tersebut membelok berlawanan jarum jam. Dalam kode yang dipakai, orientasi lurus dinotasikan sebagai 0, searah jarum jam sebagai 1, dan berlawanan jarum jam sebagai 2.[7]

A. Algoritma Jarvis March

Jarvis March adalah sebuah algoritma yang dipakai untuk mencari titik-titik yang membentuk bentuk yang paling cembung dari titik-titik yang sudah disediakan. *Jarvis March* dimulai dari titik yang paling kiri, algoritma menyimpan titik *Convex Hull* yang memiliki rotasi berlawanan arah jarum jam [2].

Dari titik mulai, algoritma dapat memilih titik selanjutnya dengan memeriksa orientasi titik tersebut dari titik mulai. Titik dengan sudut terbesar akan

dipilih. *Convex Hull* akan selesai dibuat saat titik selanjutnya adalah titik mulai [2].

B. Algoritma Graham Scan

Graham Scan adalah sebuah metode untuk mencari convex hull dari titik yang disediakan dengan menggunakan kompleksitas waktu $O(n \log n)$. Algoritma akan mencari semua titik pada *Convex Hull* yang sudah ditempatkan dengan panjang batasan [3].

Algoritma menggunakan *stack* untuk mencari dan menghilangkan batasan yang berbentuk cekung secara efisien.[3] Algoritma *Graham Scan* akan mencari titik-titik pada *Convex Hull*. *Convex Hull* akan dimulai ketika titik pertama adalah titik yang paling dekat [4].

Dari titik mulai, titik $n-1$ yang tersisa akan diurutkan berdasarkan arah berlawanan jarum jam. Jika titik adalah 2 atau lebih maka akan membentuk titik yang sama, kemudian algoritma akan menghapus semua titik yang berada ditempat yang sama terkecuali titik mulai yang berada paling jauh [4].

III. HASIL DAN PEMBAHASAN

Berikut ini adalah potongan program fungsi jarak (khusus untuk *Graham Scan*) dan fungsi orientasi dari kedua fungsi *Convex Hull*:

```
function orientation(p, q, r) {
  var val = (q.y - p.y) * (r.x - q.x) -
    (q.x - p.x) * (r.y - q.y);

  //Collinear
  if (val == 0) return 0;

  //Clockwise / Counterclockwise
  return (val > 0) ? 1 : 2;
}
```

Gambar 1. Fungsi orientasi dan fungsi jarak (khusus untuk *Graham Scan*)[7]

Berikut ini adalah potongan program algoritma *Jarvis March* yang telah diimplementasikan dalam bahasa pemrograman *Javascript* [5] :

```
function convexHullJarvis(points) {
  var n = points.length;

  if (n < 3) return;

  var hull = [];
```

```
var l = 0;
for (var i = 1; i < n; i++)
  if (points[i].x < points[l].x)
    l = i;

var p = l, q;
do{
  hull.push(points[p]);

  q = (p + 1) % n;
  for (var i = 0; i < n; i++){
    if (orientation(points[p],
                    points[i],
                    points[q]) == 2)
      q = i;
  }
  p = q;
} while (p != l);

return hull;
}
```

Gambar 2. Potongan program dari algoritma *Jarvis March*[5]

Berikut ini adalah potongan program algoritma *Graham Scan* yang telah diimplementasikan dalam bahasa pemrograman *Javascript* [6] :

```
function convexHullGraham(points) {
  var n = points.length;

  if (n < 3) return;
  for (var i = 1; i < n; i++){
    var pivot = points[0], ymin = pivot.y;
    for (var i = 1; i < points.length; i++) {
      var y = points[i].y;
      if((y < ymin) ||
        (y == ymin && points[i].x < pivot.x)){
        pivot = points[i], ymin = pivot.y;
      }
    }
  }
  points.sort(function(a, b){
    var o = orientation(pivot, a, b);

    var dstB = distSq(pivot, b);
    var dstA = distSq(pivot, a);

    if(o == 0) return(dstB >= dstA) ? -1 : 1;
    return (o == 2) ? -1 : 1;
  });
}
```

```

points.length = m;

if (m < 3) return;

// Initialize Result
var hull = [
    points[0],
    points[1],
    points[2]
];
    
```

Gambar 3. Potongan program dari algoritma *Graham Scan*[6]

Dataset dibentuk secara acak dengan target jumlah yang ditentukan. Titik yang baru dibuat akan dibuang jika memiliki posisi yang sama dengan titik yang sudah dibuat.

Target jumlah titik adalah 10 titik, 100 titik, 1.000 titik, dan 10.000 titik. Algoritma ini dijalankan sebanyak 20 kali per target.

Berikut dibawah ini adalah hasil penelitian dari perbandingan kedua algoritma.

Tabel 1. Nilai minimum, median, *mean*, dan *maximum* pada *Jarvis March*

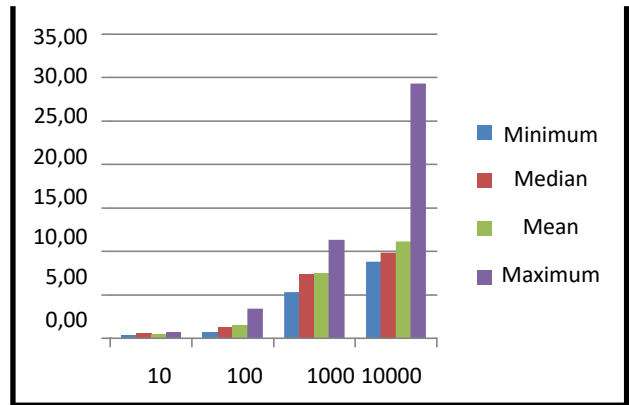
Jarvis March				
	10	100	1000	10000
Minimum	0,35	0,63	5,33	8,82
Median	0,50	1,27	7,34	9,84
Mean	0,49	1,44	7,43	11,16
Maximum	0,71	3,42	11,31	29,32

*Notes: Perbandingan kecepatan dihitung dalam satuan *millisecond*.

Tabel 2. Nilai minimum, median, *mean*, dan *maximum* pada *Graham Scan*

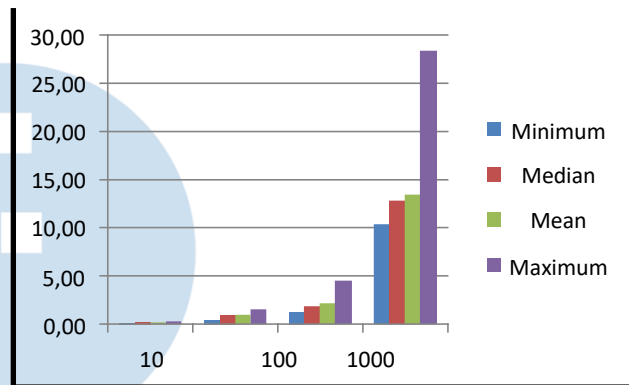
Graham Scan				
	10	100	1000	10000
Minimum	0,12	0,48	1,26	10,36
Median	0,19	1,00	1,88	12,80
Mean	0,20	0,96	2,17	13,42
Maximum	0,29	1,57	4,54	28,38

*Notes: Perbandingan kecepatan dihitung dalam satuan *millisecond*.



*Nilai di bawah adalah target jumlah titik, nilai di samping kiri adalah *runtime* algoritma dijalankan.

Gambar 4. Diagram batang pada *Jarvis March*



* Nilai di bawah adalah target jumlah titik, nilai di samping kiri adalah *runtime* algoritma dijalankan.

Gambar 5. Diagram batang pada *Graham Scan*

Diperkirakan melalui pencocokan kurva *Jarvis March* memiliki kecepatan yang sama dengan *Graham Scan* saat menggunakan sekitar 7400 titik.

Pada penelitian ini, digunakan sebanyak 20 kali percobaan di dalam target jumlah titik yang berbeda di antara kedua algoritma diatas untuk melihat hasil yang maksimal. Sehingga di dalam didapat median, minimum, *maximum*, dan *mean* di setiap target jumlah titik yang berbeda.

Pada penelitian ini, kedua algoritma dijalankan sebanyak 20 dalam masing-masing target jumlah titik sehingga dapat melihat hasil nilai minimum, median, rata-rata (*mean*), dan maksimum yang akurat.

IV. SIMPULAN

Berdasarkan dari pembahasan yang telah dilakukan, dapat ditarik kesimpulan bahwa algoritma Convex Hull merupakan algoritma yang populer dengan kelebihanannya dalam membandingkan kecepatan. Antara kedua algoritma convex hull, dapat dinyatakan bahwa algoritma Graham Scan secara umum bekerja dengan lebih cepat dibandingkan

dengan algoritma *Jarvis March*. Saat jumlah titik yang dioperasikan lebih dari 7400 titik, algoritma *Jarvis March* bekerja lebih cepat dibandingkan dengan *Graham Scan*.

DAFTAR PUSTAKA

- [1] "Convex Hull | Brilliant Math & Science Wiki" [Online]. Available: <https://brilliant.org/wiki/convex-hull/> [Accessed 10 Oktober 2020]
- [2] "Jarvis March Algorithm" [Online]. Available: <https://www.tutorialspoint.com/Jarvis-March-Algorithm> [Accessed 10 Oktober 2020]
- [3] Pankaj Sharma, "Graham Scan Algorithm to find Convex Hull" [Online]. Available: <https://iq.opengenus.org/graham-scan-convex-hull/> [Accessed 10 Oktober 2020]
- [4] Samuel Sam, "Graham Scan Algorithm" [Online]. Available: <https://www.tutorialspoint.com/Graham-Scan-Algorithm> [Accessed 10 Oktober 2020]
- [5] "Convex Hull | Set 1 (Jarvis's Algorithm or Wrapping) – GeeksforGeeks" [Online]. Available: <https://www.geeksforgeeks.org/convex-hull-set-1-jarvis-algorithm-or-wrapping/> [Accessed 15 Oktober 2020]
- [6] "Convex Hull | Set 2 (Graham Scan) – GeeksforGeek" [Online]. Available: <https://www.geeksforgeeks.org/convex-hull-set-2-graham-scan/> [Accessed 15 Oktober 2020]
- [7] "Orientation of 3 ordered points – Geeks for Geeks" [Online]. Available: <https://www.geeksforgeeks.org/orientation-3-ordered-points/> [Accessed 25 Oktober 2020]
- [8] M. A. Jayaram, Hasan Fleyeh, "Convex Hulls in Image Processing: A Scoping Review" [Online]. Available: <http://article.sapub.org/10.5923.j.ajis.20160602.03.html> [Accessed 1 November 2020]
- [9] Tobing, F.A.T. and Nainggolan, R., 2020. ANALISIS PERBANDINGAN PENGGUNAAN METODE BINARY SEARCH DENGAN REGULAR SEARCH EXPRESSION. *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, 4(2), pp.168-172.
- [10] Setiabudi, D. H. and Jayasaputra, E. " Uji Kecepatan Algoritma Convex-Hull: Graham dan Melkman" *Jurnal Teknik Elektro Vol. 2, No. 1, Maret 2002: 27- 31*.
- [11] Tobing, F.A.T. and Tambunan, J.R., 2020. Analisis Perbandingan Efisiensi Algoritma Brute Force dan Divide and Conquer dalam Proses Pengurutan Angka. *Ultimatics: Jurnal Teknik Informatika*, 12(1), pp.52-58.