

Algoritma Brute Force pada Fluid Particle Engine

Alfian Ramadhan 13509078
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
alfian.ramadhan@students.itb.ac.id

Abstrak—*Fluid particle engine* merupakan *engine* yang memvisualisasikan pergerakan partikel fluida. Saat ini, masih jarang adanya pembahasan mengenai pembuatan visualisasi fluida yang digunakan sebagai *engine* dalam aplikasi maupun *game design* terutama mengenai bagaimana cara yang paling efektif untuk memproses setiap partikel agar dapat membentuk bentuk fluida yang efektif dari segi waktu. Untuk itu akan dibahas mengenai penggunaan algoritma Brute Force terhadap partikel fluida agar dapat dilakukan pemrosesan dengan cepat dan efektif sesuai dengan pergerakan fluida dan kemudian dapat ditampilkan secara visual dengan baik.

Kata Kunci—*fluid particle, engine, Brute Force, game design, aplikasi.*

I. PENDAHULUAN

Fluida merupakan sumber daya alami yang sering dijumpai sehari-hari. Contoh bentuk fluida yang umum, misalnya, angin, gelombang laut, atau bahkan segelas air. Fenomena tersebut merupakan kejadian yang sederhana yang biasa terlihat, namun dapat menjadi sangat kompleks dan rumit apabila disimulasikan secara nyata.

Dalam ukuran paling kecilnya, fluida hanya berbentuk partikel-partikel kecil. Partikel tersebut saling berikatan membentuk molekul besar kemudian membentuk satu kesatuan apabila berjumlah banyak.

Apabila diteliti lebih lanjut, kumpulan fluida hanyalah seperti kumpulan bola-bola kecil yang saling berikatan satu sama lain dan terlihat sebagai satu genangan fluida. Kumpulan fluida tersebut saling berikatan dalam jarak tertentu yang dapat dijangkauanya.



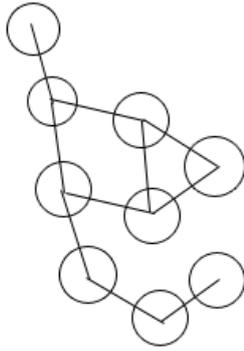
Gambar 1 Ilustrasi penumpahan fluida ke dalam gelas [3]



Gambar 2 Tetesan Fluida



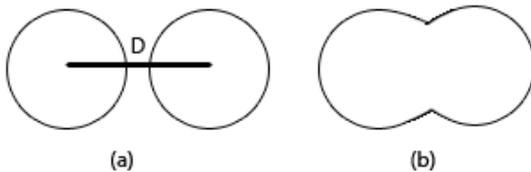
Gambar 3 Partikel Fluida bentuk kecil



Gambar 4 Partikel Fluida dalam bentuk garis-garis

Perlu ditekankan bahwa dalam makalah ini, diasumsikan bahwa partikel-partikel fluida hanyalah berbentuk sebagai bulatan partikel kecil yang apabila terhubung dengan partikel lainnya, akan membentuk satu ikatan yang digambarkan berupa garis antara kedua partikel tersebut.

II. MODEL FLUIDA SEBAGAI PARTIKEL



Gambar 5 Hubungan partikel

Partikel fluida dimodelkan sebagai bulatan partikel kecil. Antara partikel satu dengan partikel lainnya dihitung jarak yang ditandai pada gambar (a) dengan huruf D. D adalah jarak antarpusat kedua lingkaran dengan rumus:

$$D^2 = (y_1^2 - y_2^2) - (x_1^2 - x_2^2)$$

Rumus tersebut merupakan rumus umum yang digunakan untuk menghitung jarak pusat kedua lingkaran dengan memanfaatkan titik pusat lingkaran kemudian diambil garis yang menghubungkan keduanya.

Pada gambar (b) diperlihatkan bahwa ketika kedua partikel bersatu, akan membentuk ikatan sehingga digambarkan dengan bulatan yang saling mengiris. Pada program yang akan dibuat, hubungan kedua partikel yang berikatan akan ditandai dengan sebuah garis.

II. ALGORITMA BRUTE FORCE

Brute force adalah sebuah pendekatan yang lempang (straightforward) untuk memecahkan suatu masalah (problem statement) dan definisi konsep yang melibatkan Algoritma brute force memecahkan masalah dengan sangat sederhana, langsung, dan dengan cara yang jelas (obvious way) meskipun bukan merupakan solusi yang paling mangkus. Karakteristik Algoritma Brute Force.

1. Algoritma Brute Force umumnya tidak “cerdas” dan tidak mangkus karena ia membutuhkan jumlah langkah yang besar dalam penyelesaiannya. Kadang pula algoritma Brute Force disebut juga algoritma naïf (naïve algorithm).
2. Algoritma Brute Force lebih cocok untuk masalah yang berukuran kecil.
3. Meskipun bukan metode yang mangkus, hampir semua masalah dapat diselesaikan dengan algoritma Brute Force.

Kekuatan dan Kelemahan Metode Brute Force

Kekuatan:

1. Metode brute force dapat digunakan untuk memecahkan hampir sebagian besar masalah (wide applicability).
2. Metode brute force sederhana dan mudah dimengerti.
3. Metode brute force menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan string, perkalian matriks.
4. Metode brute force menghasilkan algoritma baku (standard) untuk tugas-tugas komputasi seperti penjumlahan/perkalian n buah bilangan, menentukan elemen minimum atau maksimum di dalam tabel (list).

Kelemahan:

1. Metode brute force jarang menghasilkan algoritma yang mangkus.
2. Beberapa algoritma brute force lambat sehingga tidak dapat diterima.
3. Tidak sekonstruktif/sekreatif teknik pemecahan masalah lainnya.

III. PENERAPAN ALGORITMA

Algoritma Brute Force pada partikel fluida ini melakukan iterasi satu-persatu mulai dari partikel pertama sampai partikel terakhir yang ada. Algoritma ini mengukur jarak dari partikel pertama sampai partikel ke-N dengan partikel lainnya. Misal terdapat N buah partikel. Algoritma tersebut akan menghitung jarak antara partikel 1 dengan 2 kemudian dengan 3, dan seterusnya, apabila jaraknya mencukupi untuk

melakukan hubungan maka digambar sebuah garis penghubung. Iterasi tersebut dilanjutkan dengan partikel selanjutnya sampai N, namun tanpa adanya redundansi pengecekan antarpartikel sehingga kompleksitasnya tidak begitu mahal.

Pseudocode Brute Force Fluida

```

procedure BruteForceFLuida ()
{ Prosedur utama untuk melakukan
pengecekan secara Brute Force terhadap
semua partikel apakah suatu partikel
berikatan dengan partikel lainnya
}

Kamus
i, j : integer
living : List of Partikel

Algoritma
for (i=0; i<living.size(); i++){
    for (j=i+1; j<living.size(); j++){
        if (pair(living.get(i),
living.get(j), 30)){

g2d.draw(new
Line2D.Double(living.get(i).getCenter().ge
tX(), living.get(i).getCenter().getY(),
living.get(j).getCenter().getX(),
living.get(j).getCenter().getY()));

        }
    }
}

```

Prosedur di atas merupakan prosedur utama dalam menjalankan operasi Brute Force dalam pengecekan setiap partikel dengan kondisi jika dapat dihubungkan maka digambar sebuah garis penghubung.

Pseudocode Jarak Partikel

```

function Pair (a,b : Partikel) → boolean
{ Merupakan fungsi untuk mengecek apakah
kedua partikel dapat berikatan atau tidak
}

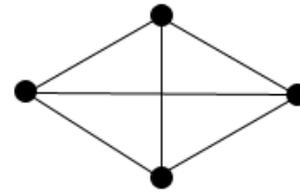
Kamus
pair : boolean
D : double

Algoritma
pair ← false;
D ← Math.sqrt(Math.pow((b.getY()-
a.getY()),2)+Math.pow((b.getX()-
a.getX()),2));
if (D<=delta) pair ← true;
→ pair;

```

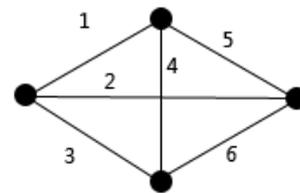
Fungsi di atas merupakan fungsi yang mengembalikan nilai true jika kedua partikel dapat dihubungkan dan false jika kedua partikel ternyata memiliki jarak yang cukup jauh untuk dihubungkan.

Algoritma Brute Force ini melakukan iterasi sebesar jumlah sisi. Hal ini dapat digambarkan dengan gambar berikut:



Gambar 6 Hubungan partikel dalam graf

Setiap simpul dicek dengan simpul lainnya dan dihubungkan oleh sebuah garis. Begitu pula algoritma Brute Force untuk partikel fluida ini. Setiap partikel dicek apabila memenuhi jarak yang cukup untuk melakukan ikatan maka partikel tersebut akan dihubungkan. Namun pengecekan hanya dilakukan antara satu partikel dengan partikel lainnya tanpa adanya redundansi pengecekan. Satu simpul dihubungkan dengan simpul lainnya dengan pengecekan satu arah saja.



Gambar 7 Penelusuran Brute Force

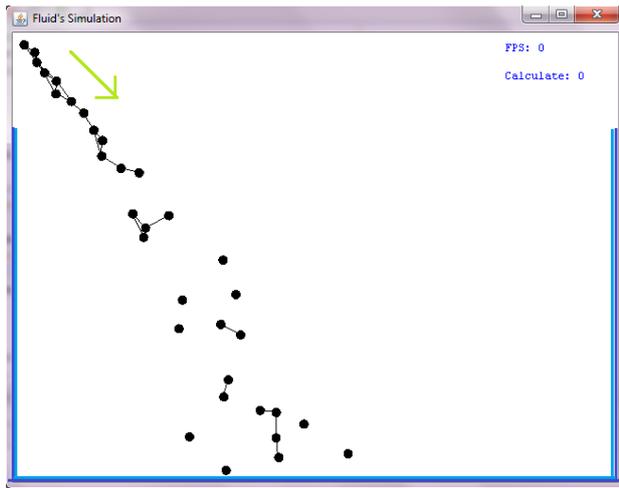
Algoritma Brute Force tersebut dapat dihitung dengan rumus :

$$\sum_{i=1}^{n-1} i = 1 + 2 + 3 + \dots + (n-1) = \frac{(1+(n-1)) \times (n-1)}{2}$$

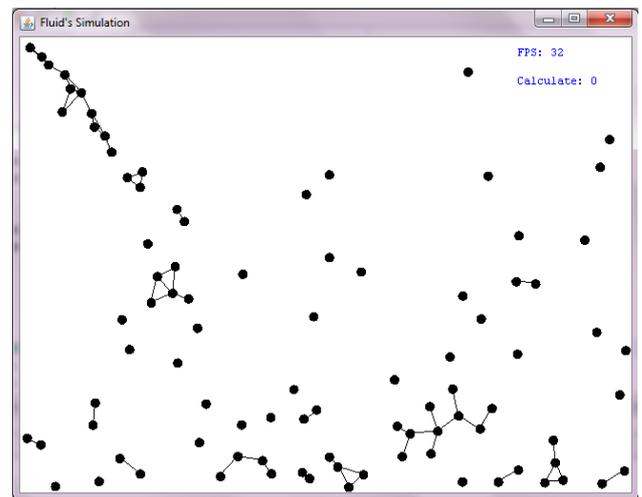
Rumus di atas menghasilkan kompleksitas sebesar $O(n^2)$

IV. IMPLEMENTASI

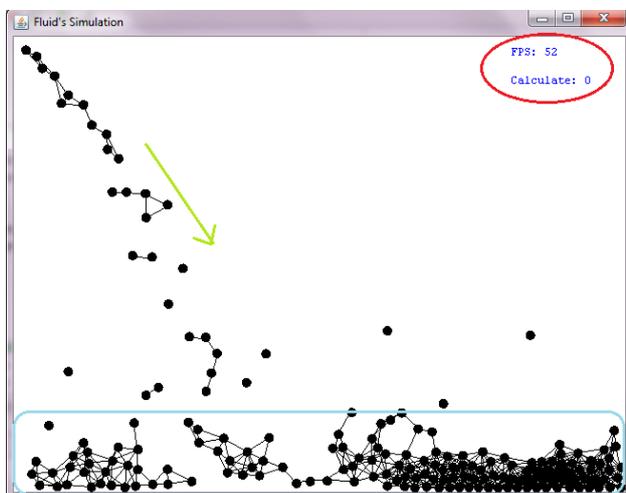
Pada tahap implementasi, penulis membuat suatu program yang diubah dari *physic engine* java yang *open source* menjadi program yang dijadikan percobaan untuk menyelesaikan permasalahan *engine* partikel fluida ini. Dengan IDE Netbeans, program dibuat sedemikian rupa agar menyerupai kejadian saat seseorang memasukkan air ke dalam gelas. Diasumsikan partikel-partikel fluida tersebut berbentuk bulatan hitam yang apabila terhubung dengan partikel lainnya akan diberi tanda suatu garis. Dapat dilihat pada gambar di bawah ini.



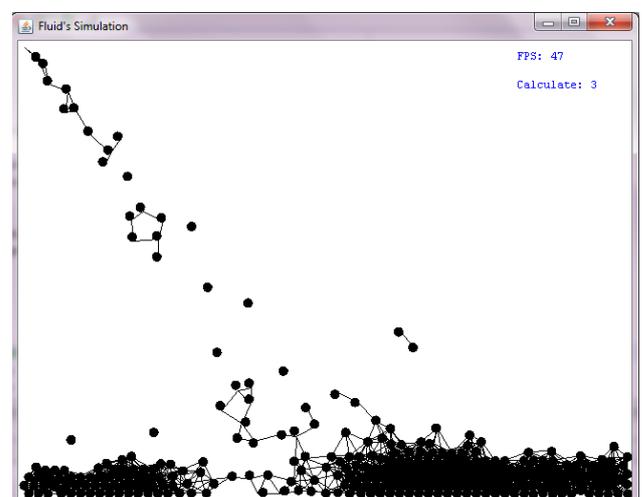
Gambar 8 Tampilan program pengujian



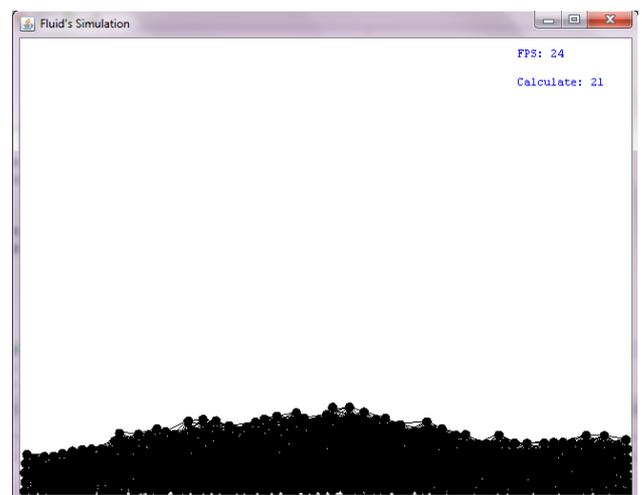
Gambar 10 Pengujian dengan $N = 100$



Gambar 9 Tampilan pengujian program bagian 2



Gambar 11 Pengujian dengan $N = 300$

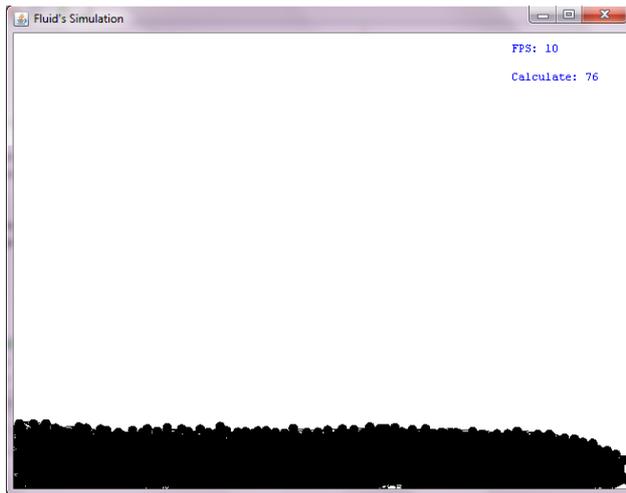


Gambar 12 Pengujian dengan $N = 600$

Gambar di atas mengilustrasikan tumpahan air dalam sebuah kotak biru. Air dalam bentuk partikel fluida ditumpahkan secara miring dan nantinya akan berbenturan dengan dinding kotak. Layaknya partikel fluida yang ditumpahkan ke dalam gelas lalu bersatu padu membentuk genangan dan berikatan. Terdapat juga informasi di kanan atas mengenai jumlah frame per detik dan kalkulasi dalam pengecekan menggunakan Brute Force dalam satuan milidetik.

IV. HASIL PENGUJIAN DAN ANALISIS

Dilakukan pengujian terhadap program dengan melakukan variasi terhadap jumlah partikel yang digunakan.



Gambar 13 Pengujian dengan $N = 1000$

Pada beberapa gambar di atas memperlihatkan perlakuan partikel fluida dengan jumlah frame per detik dan waktu pengecekan Brute Force yang bervariasi. Waktu kalkulasi yang bernilai nol menandakan bahwa pengecekan berkisar kurang dari satu milidetik. Hal tersebut dikarenakan pembulatan dari bahasa pemrograman Java. Untuk $N = 1000$, gambar partikel lebih terlihat rapat daripada $N = 600$ dikarenakan efek partikel yang saling berhubungan akan saling rapat sesuai dengan sifat fluida.

| Jumlah Partikel | Frame per second | Waktu Pengecekan (milisecond) |
|-----------------|------------------|-------------------------------|
| 100 | 32 | 0 |
| 300 | 47 | 3 |
| 600 | 24 | 21 |
| 1000 | 10 | 76 |

Tabel 1 Waktu hasil pengujian

Berdasarkan tabel informasi di atas dijelaskan bahwa semakin banyak partikelnya, semakin besar waktu yang diperlukan untuk menghitung pengecekan Brute Force dalam pengecekan apakah partikel berikatan. Apabila diambil suatu pembulatan, dalam seribu partikel fluida, algoritma Brute Force dapat mengecek semua partikel hanya dalam hitungan tidak lebih dari 100 milisecond atau tidak lebih dari sekitar satu per sepuluh detik. Dengan jangka waktu yang cukup singkat, algoritma ini dapat digunakan dalam pengaplikasian *engine* partikel fluida. Seribu partikel ini dapat dibentuk sebagai bentuk fluida yang dapat digambar ulang sebagai zat cair sehingga tampilannya tidak seperti gambar yang disajikan. Dengan algoritma Brute Force, partikel fluida dapat dipakai sebagai *engine* dalam aplikasi maupun game dengan visualisasi yang baik.

V. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil dari pembahasan di atas antara lain:

1. Algoritma Brute Force cocok untuk menyelesaikan permasalahan pergerakan partikel fluida dalam pengaplikasiannya sebagai *engine*.
2. Algoritma Brute Force ini merupakan algoritma yang mampu menyelesaikan masalah dengan cara sederhana, langsung, dan jelas meskipun bukanlah algoritma yang mangkus.
3. Untuk jumlah partikel yang semakin besar, pemrosesan partikel akan semakin lambat sehingga dibutuhkan algoritma yang lebih mangkus lagi.

Saran untuk pengembangan selanjutnya:

1. Diperlukan penelitian lebih lanjut dengan menggunakan algoritma lainnya sehingga pemrosesan pergerakan fluida ini dapat lebih efektif lagi.
2. Lakukan pengujian dengan komputer yang memiliki spesifikasi tinggi dan kaku yang lebih cocok untuk kalkulasi tinggi agar pengujian dapat dilakukan dengan jumlah partikel yang sangat besar sesuai dengan visualisasi fluida yang alami.

REFERENSI

- [1] Munir, Rinaldi, 2009, *Diktat Kuliah IF3051 Strategi Algoritma*, Program Studi Teknik Informatika ITB, hal 8-19.
- [2] www.informatika.org/~rinaldi (diakses pada tanggal 11 Desember 2011)
- [3] Muller, Mathias, etc, *Particle-Based Fluid Simulation for Interactive Applications*. Switzerland, Department of Computer Science, Federal Institute of Technology Zurich

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, 8 Desember 2011

Alfian Ramadhan
NIM 13509078