BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN APLIKASI

3.1 Metodologi Penelitian

Pada penelitian ini, metodologi yang akan digunakan adalah sebagai berikut.

3.1.1 Studi Literatur

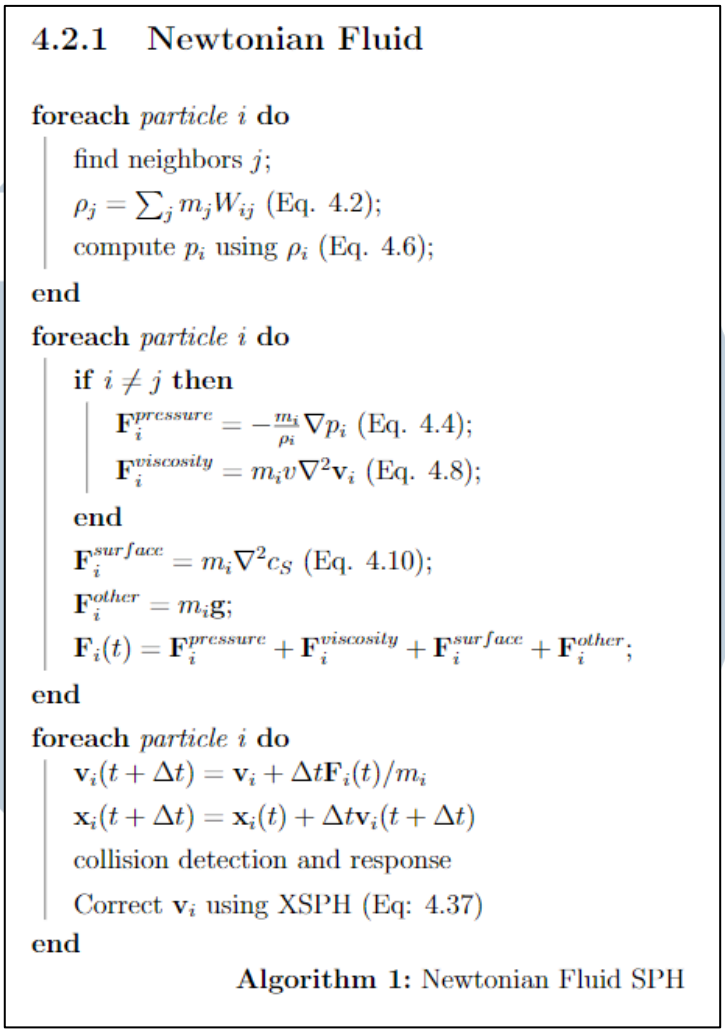
Pada studi literatur yang akan dilakukan, dipelajari teori mengenai *Computational Fluid Dynamics,* dan *Smoothed Particle Hydrodynamics* (SPH).

3.1.2 Identifikasi Masalah

Identifikasi masalah dilakukan dengan cara mencari kekurangan dari penelitian lain. Selain itu, membandingkan hasil jumlah partikel dan kecepatan komputasinya pada penelitian lain.

3.1.3 Implementasi Algoritma SPH

Alur implementasi algoritma SPH dilakukan sesuai dengan alur yang dijelaskan oleh Ertekin (2015) seperti pada Gambar 4.1. Pada Gambar 4.1, alur algoritma terbagi menjadi tiga bagian *foreach loop,* setiap bagian memiliki fungsi masing-masing. Dikutip dari Schuermann (2017), fungsi-fungsi pada setiap bagian tersebut secara berurutan dinamakan fungsi perhitungan kepadatan massa dan tekanan, fungsi perhitungan gaya, serta fungsi integrasi.



4.2.1 Newtonian Fluid

foreach *particle* i **do**
  find neighbors j;
  $\rho_j = \sum_j m_j W_{ij}$ (Eq. 4.2);
  compute p_i using ρ_i (Eq. 4.6);
end
foreach *particle* i **do**
  if $i \neq j$ **then**
    $\mathbf{F}_i^{pressure} = -\frac{m_i}{\rho_i}\nabla p_i$ (Eq. 4.4);
    $\mathbf{F}_i^{viscosity} = m_i v \nabla^2 \mathbf{v}_i$ (Eq. 4.8);
  end
  $\mathbf{F}_i^{surface} = m_i \nabla^2 c_S$ (Eq. 4.10);
  $\mathbf{F}_i^{other} = m_i \mathbf{g}$;
  $\mathbf{F}_i(t) = \mathbf{F}_i^{pressure} + \mathbf{F}_i^{viscosity} + \mathbf{F}_i^{surface} + \mathbf{F}_i^{other}$;
end
foreach *particle* i **do**
  $\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i + \Delta t \mathbf{F}_i(t)/m_i$
  $\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t \mathbf{v}_i(t + \Delta t)$
  collision detection and response
  Correct $\mathbf{v}_i$ using XSPH (Eq: 4.37)
end

Algorithm 1: Newtonian Fluid SPH

Gambar 4.1 Alur Algoritma SPH (Ertekin, 2015)

Fungsi perhitungan kepadatan massa dan tekanan diimplementasikan menggunakan Rumus 2.2 untuk menghitung kepadatan massa dari setiap partikelnya, lalu perhitungan tekanan dilakukan mengikuti rumus sederhana dari Schuermann (2017) yaitu $R * (\rho_j - \rho)$ dimana R adalah konstanta gas, ρ_j adalah kepadatan massa dari setiap partikel j, dan ρ adalah nilai massa jenis yang bersifat konstan.

Fungsi perhitungan gaya dapat berisi gaya apapun yang dapat diterima misalnya gaya *surface tension*, gaya eksternal, dan lain lain. Namun, pada penelitian ini hanya akan mengimplementasikan gaya tekanan, gaya gravitasi dan

gaya *viscosity*, sehingga variabel $F_i^{surface}$ pada Gambar 4.1 tidak digunakan. Gaya tekanan dihitung menggunakan Rumus 2.3, gaya *viscosity* dihitung dengan menggunakan Rumus 2.4, dan gaya gravitasi merupakan nilai konstan.

Fungsi integrasi merupakan fungsi yang menggunakan nilai hitungan gaya gabungan untuk menghitung *velocity* dari setiap partikel. Kemudian, nilai *velocity* dari setiap partikel tersebut akan digunakan untuk menghitung posisi partikel yang bersangkutan.

3.1.4 Pencarian Metode Optimisasi

Algoritma dasar SPH yang telah diimplementasikan akan dianalisa untuk mengetahui bagian yang memiliki kompleksitas dan atau biaya komputasi yang tinggi. Kemudian, setelah ditemukan, akan dilakukan pencarian metode untuk mengoptimisasikan bagian tersebut.

3.1.5 Implementasi Metode Optimisasi

Setelah menentukan metode optimisasi apa saja yang dapat digunakan, dilakukan implementasi metode optimisasi terhadap algoritma dasar SPH. Hasil implementasi dan algoritma SPH tanpa implementasi akan digunakan pada Program simulasi keseluruhannya.

3.1.6 Perancangan dan Pembuatan Program

Perancangan program akan dilakukan untuk menyediakan gambaran untuk penggabungan kombinasi-kombinasi algoritma SPH beserta metode optimisasinya. Program dibuat dengan tujuan untuk memperlihatkan hasil implementasi dari setiap kombinasi metode optimisasi sehingga dapat menunjukan metode optimisasi yang daya komputasi paling ringan dibandingkan dengan yang lainnya. Kemudian, dari hasil rancangan akan dibuat program. Dari hasil pembuatan program, setiap

kombinasi metode optimisasi akan dijalankan dengan menerima input berupa jumlah partikel. Setelah itu, program akan menampilkan kecepatan *rendering* dalam bentuk FPS. Nilai FPS akan dicatat dan dibandingkan untuk setiap metode optimisasinya.

3.1.7 Uji Coba dan Evaluasi

Program akan diuji dan dievaluasikan pada setiap kombinasi metodenya untuk mendapat hasil yang paling optimal berdasarkan jumlah partikel dengan kecepatan *rendering* minimal yaitu 10 FPS.

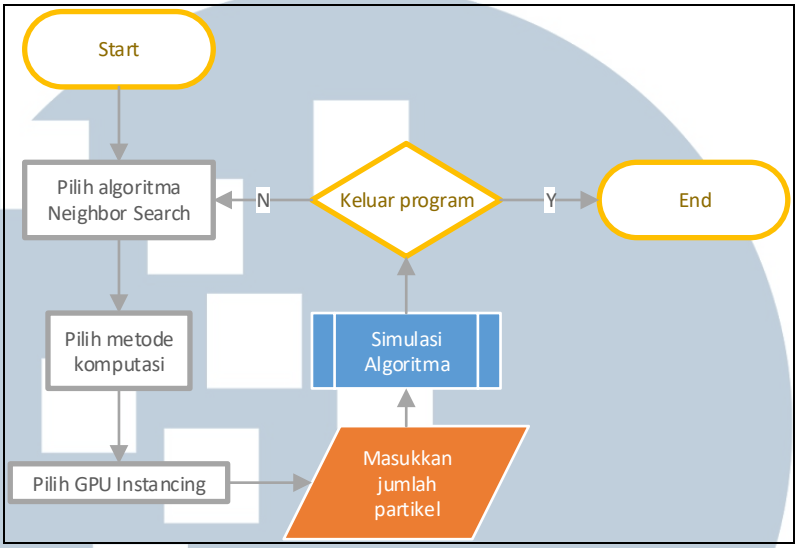
3.2 Perancangan Aplikasi

Perancangan aplikasi dari penelitian ini dibagi menjadi 3 bagian, yaitu *pseudocode*, *flowchart* dan perancangan antarmuka.

3.2.1 Flowchart

Alur dari aplikasi yang telah dibuat digambarkan dalam dua *flowchart* yaitu *flowchart* Program dan *flowchart* Simulasi, dimana *flowchart* Program menjelaskan alur berjalannya program secara keseluruhan. *Flowchart* Simulasi, yang menjelaskan alur berjalannya algoritma SPH pada saat simulasi berjalan.

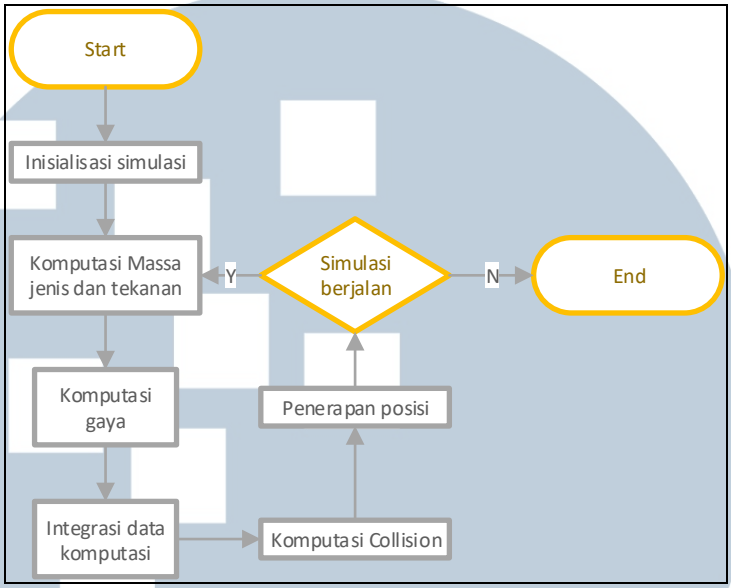
A. Flowchart Program



Gambar 3.2 Flowchart Program

Flowchart Program yang digambarkan pada Gambar 3.2 menjelaskan bahwa pengguna akan diberikan pilihan sebelum melakukan simulasi yaitu pemilihan algoritma Nearest Neighbor Search, metode komputasi, GPU Instancing, dan jumlah partikel yang ingin di simulasikan. Ketika pengguna sudah memberikan pilihan, maka pengguna dapat memulai simulasi yang digambarkan pada Gambar 3.3.

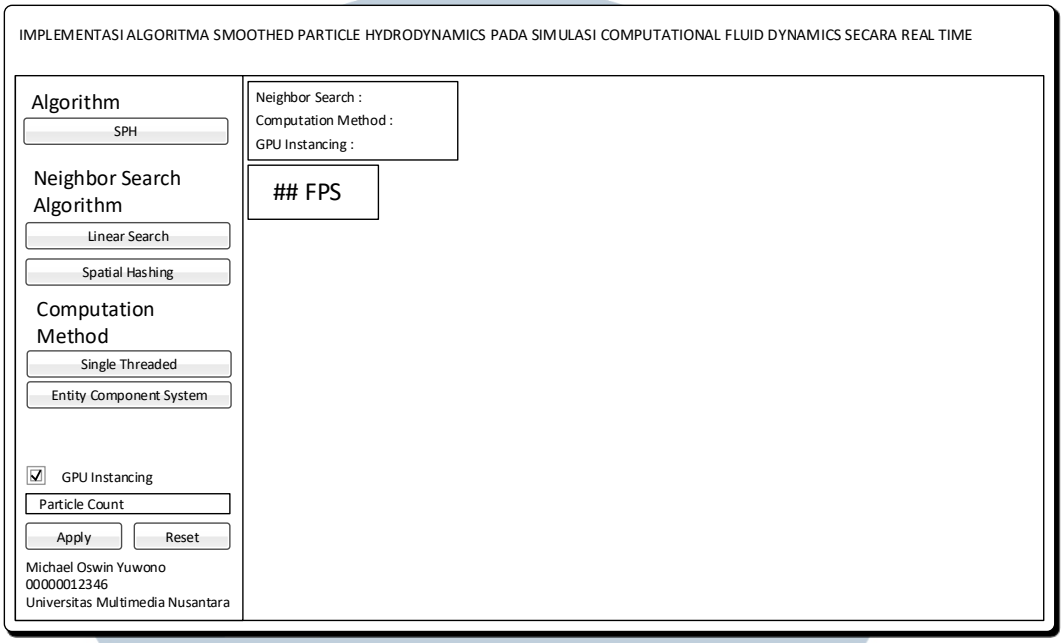
B. Flowchart Simulasi Algoritma



Gambar 3.3 Flowchart Simulasi

Flowchart Simulasi seperti pada Gambar 3.3, merupakan *flowchart* yang menjelaskan alur algoritma SPH berjalan. Pertama, simulasi akan melakukan inisialisasi parameter sesuai dengan pilihan pengguna. Setelah itu, akan terjadi loop yang berjalan selama simulasi masih berjalan, dalam arti pengguna belum menekan tombol Reset yang bisa dilihat pada rancangan antarmuka di Gambar 3.4. Dalam loop tersebut, pertama simulasi akan menghitung kepadatan massa dan tekanan setiap partikel. Kemudian, simulasi akan menghitung komputasi gaya yang setelah itu simulasi akan melakukan integrasi antara velocity dan posisi setiap partikel dari nilai yang didapat pada komputasi kepadatan massa dan tekanan, serta gaya. Lalu, akan terjadi simulasi *collision* untuk memungkinkan partikel berada pada wadah yang disediakan. Setelah itu, posisi partikel akan diterapkan sesuai dengan hasil semua komputasi sebelumnya. Komputasi yang telah dijelaskan akan berjalan terus menerus diulang dari komputasi kepadatan massa pada setiap *frame*-nya selama simulasi berjalan.

3.2.2 Rancangan Antarmuka



Gambar 3.4 Rancangan Antarmuka Aplikasi

Gambar 3.4 merupakan rancangan antarmuka aplikasi, terdiri dari tiga panel utama yaitu panel pilihan, panel informasi, dan panel *Frames per Second* (FPS). Panel pilihan terdiri dari pilihan-pilihan optimisasi seperti algoritma Nearest Neighbor Search, metode komputasi, GPU Instancing, dan input jumlah partikel. Lalu panel informasi terdiri dari informasi algoritma atau metode yang sudah terpilih. Kemudian, panel FPS menampilkan informasi kecepatan *rendering* yang nilainya berubah setiap saat. Nilai FPS dapat bervariasi pada setiap perangkatnya.