

Model Penghindaran Tabrakan Multi Obyek Menggunakan Repulsive Field

Akuwan Saleh¹, Moch. Hariadi², Supeno Mardi S. N²

^{1,2}Pasca Sarjana, Jurusan Teknik Elektro, Fakultas Teknik Industri
Institut Teknologi Sepuluh Nopember, Surabaya

e-mail: akuwan@eepis-its.edu, mochar@ee.its.ac.id, mardi@ee.its.ac.id

Abstrak

Mendeteksi tabrakan antar obyek dan menampilkan gerakan penghindaran tabrakan yang alami menjadi topik yang diinginkan dalam berbagai aplikasi navigasi agen otonom. Pada umumnya, gerakan penghindaran tabrakan multi obyek statis dan dinamis secara alami banyak menggunakan teknik dengan pengambilan keputusan untuk bergerak menghindari kearah lain. Teknik penghindaran ini mengalami kesulitan ketika jumlah obyek diperbanyak.

Pada penelitian ini menyajikan teknik yang mampu menampilkan gerakan penghindaran tabrakan multi obyek statis dan dinamis secara alami. Metode yang digunakan Sphere-Plane Detection (SPD) dan Sphere-Sphere Detection (SSD) untuk mendeteksi tabrakan dengan referensi jarak dan metode potential field jenis repulsive untuk penghindaran tabrakan. Pengujian dilakukan dengan jumlah obyek 100 yang terdiri dari 34 agen A, 33 agen B dan 33 agen C, multi penghalang dengan dan tanpa repulsive, penghalang statis dan dinamis, serta diuji dalam animasi boid.

Dari pengujian yang telah dilakukan diperoleh nilai vektor $|V|$ dari posisi awal sampai ke tujuan untuk agen A_2 sebesar 51.4522, agen B_1 bernilai 45.0853 dan agen C_4 sebesar 27.7237, setiap agen memiliki lebih dari satu jalur untuk menuju tujuan, didapatkan tiga model gerakan penghindaran dan tiga parameter yang mempengaruhi gerakan agen yaitu perubahan nilai repulsive, jarak antar penghalang dan penghalang dinamis.

Kata kunci: deteksi tabrakan, SPD, SSD, penghindaran tabrakan, repulsive field.

1. Pendahuluan

Deteksi tabrakan merupakan bagian penting dalam game 3D. Ini memastikan bahwa permainan dengan dasar fisika relatif lebih realistis, sehingga obyek tidak terpotong ketika melalui benda lainnya atau melayang

ketika harus jatuh. Seberapa baik permainan dapat mendeteksi tabrakan merupakan bagian integral dari *believability* dalam menikmati sebuah permainan. Sebuah sistem pendeteksian tabrakan di implementasikan dengan buruk dapat menjadi kelemahan untuk sebuah produk, sedangkan implementasi yang sangat baik dapat menghasilkan hasil yang luar biasa.

Dua bagian utama dalam tabrakan adalah mendeteksi apakah telah terjadi tabrakan atau tidak. Dengan demikian, menanggapi tabrakan dan menemukan jika tabrakan telah terjadi adalah dasar dari masalah ini.

Pada penelitian ini obyek yang berada dalam dunia virtual yang dibuat dengan VRML (*Virtual Reality Modeling Language*) adalah standar internasional untuk menggambarkan bentuk 3-D dengan pemandangan di *World Wide Web* [4]. Dilengkapi dengan sensor, salah satunya adalah bentuk *sphere* dengan radius tertentu, sehingga dapat menginformasikan apakah obyek lain berada dalam jarak jangkauan sensor atau tidak, jika jaraknya sama dengan jangkauan sensor maka obyek akan menghindari tabrakan dengan bergerak ke arah lain.

Dengan menyatukan obyek-obyek statis maupun dinamis dalam satu *scene* pada dunia *virtual* dan memberikan metoda *potential field* dengan jenis *field repulsive* dapat menghasilkan penghindaran terjadinya tabrakan antar obyek tersebut. Penerapan metode representasi berbagai *shape* dalam pemodelan 3D untuk animasi komputer, perlu adanya algoritma deteksi tabrakan yang dapat diterapkan pada koleksi benda yang terdiri dari beragam jenis *shape* [1]. Metode *Sphere-Plane Detection* (SPD), *Sphere-Sphere Detection* (SSD) dan deteksi penghindaran tabrakan antar obyek yang mengacu pada metode *Potential Field* khususnya jenis *Repulsive* sebagai algoritma yang ada pada penelitian ini yang akan disimulasikan. Penggunaan metode ini mampu menampilkan gerakan sebagai prepartikel dengan meminimalkan energi, dan mengadopsi rangkaian prespektif pada sistem. Hasil dari penelitian

ini dapat diimplementasikan dalam animasi komputer tentang *boid* atau perilaku sekelompok agen.

2. Kajian Pustaka

2.1. Metode *Sphere-Plane Detection*

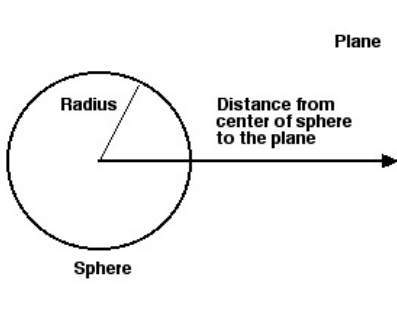
Salah satu cara untuk mendeteksi tabrakan dalam dunia virtual 3D adalah dengan menggunakan metode *Sphere-Plane Detection (SPD)* [5]. Pada gambar 1. merupakan metode SPD untuk menentukan apakah sebuah obyek memiliki berpotongan dengan bola. Jika jarak dari pusat bola ke sebuah obyek (*d*) kurang dari atau sama dengan jari-jari bola (*r*), maka tabrakan telah terjadi. Jarak antara titik, yang merupakan titik pusat bola dan *plane* dihitung dengan mengambil dot product posisi normal dan lingkup plane itu.

$$\text{jarak} = \text{sphere.position} \cdot \text{plane.normal} \quad (1)$$

atau $\mathbf{d} = (C - P_0) \cdot \mathbf{n}$; $\mathbf{D} = -P_0 \cdot \mathbf{n}$

maka: $\mathbf{d} = \mathbf{n} \cdot \mathbf{C} + D$ (2)

dimana: *C* = titik pusat obyek *sphere*, *p₀* = setiap titik pada *plane*
n = unit normal pada *plane*



Gambar 1. Metode *sphere-plane detection*

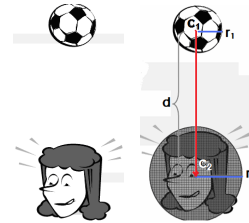
Persamaan *plane* dengan dengan vektor normal $N=(A,B,C)$ melalui titik (x, y, z) . adalah $A(x-x_0) + B(y - y_0) + C(z - z_0) = 0$ atau $Ax + By + Cz + D = 0$, maka nilai $D = -Ax_0 -By_0 -Cz_0$. Dengan demikian besar dari unit normal vektor dengan notasi *n* adalah:

$$\mathbf{n} = \frac{N}{\|N\|} = \frac{(A, B, C)}{\sqrt{A^2 + B^2 + C^2}} \quad (3)$$

2.2. Metode *Sphere-Sphere Detection*

Cara lain untuk mendeteksi terjadinya tabrakan antar obyek dengan *sphere-sphere detection* yaitu jika jarak dari pusat bola ke sebuah pusat obyek bola yang lain adalah kurang dari jumlah jari-jari kedua bola

tersebut [2]. Sebagai ilustrasi diperlihatkan pada gambar 2.



Gambar 2. Metode *sphere-sphere detection*

Persamaan untuk menentukan jarak antara obyek di atas adalah:

$$\mathbf{D} = (C_2 - C_1)$$

$$d = \sqrt{(D.x)^2 + (D.y)^2 + (D.z)^2}$$

maka tabrakan telah terjadi jika:

$$d < (r_1 + r_2) \quad (4)$$

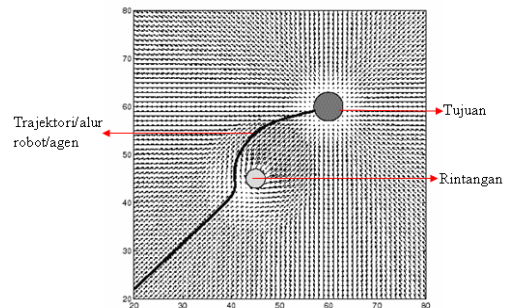
2.3. *Potential Field*

Penggunaan metode ini mampu menampilkan gerakan sebagai partikel dengan meminimalkan energi, dan mengadopsi rangkaian prespektif pada sistem. Memandu semua gerakan individu secara serempak. Menyatukan perencanaan alur global dan penghindaran benturan lokal ke dalam kerangka optimasi tunggal [8].

Ada dua model dari potensial field yaitu *attractive* dan *repulsive* [3]. Implementasi atraksi dan penghindaran dari kedua model potensial field ini adalah:

1. **Tujuan:** menggunakan model attractive field.
2. **Rintangan:** menggunakan model repulsive fields.
3. **Agan:** menggunakan model repulsive fields.
4. **Hasil ideal:** bergerak menuju sasaran sambil menghindari tabrakan dengan rintangan.
5. **Rintangan dinamis:** update potential field dengan bergerak cepat untuk menghindari rintangan.

Sebagai contoh diperlihatkan hasil trajektori robot.



Gambar 3. Hasil trajektori robot^[3]

Aplikasinya untuk perhitungan alur optimal dengan ketentuan penambahan dua medan divisualisasikan sebagai medan vektor:

$$(x, y) \rightarrow (\Delta x, \Delta y) \tag{5}$$

2.4. Algoritma Boids

Boids adalah sebuah algoritma yang merepresentasikan gerak dari sebuah kawanan. Perilaku yang dihasilkan sangat mirip dengan kumpulan ikan atau kawanan burung. Gerak *boids* dihasilkan dari tiga aturan sederhana yaitu *cohesion*, *alignment*, *separation*[7].

Separation

Pembatasan jika sebuah agen terlalu dekat dengan agen lainnya, dengan cara melakukan penyesuaian arah dan kecepatan untuk menghindari benturan (*collision*).

$$\sum_{n \in N} \text{Normalize}(\text{agent}_{pos} - n_{pos}) \tag{6-1}$$

Alignment

Mengambil rata-rata dari semua percepatan agen yang lain dan melakukan penyesuaian percepatan untuk pindah kearah kelompok

$$\sum_{n \in N} \text{Normalize}(n_{vel}) \tag{6-2}$$

2)

Cohesion

Menghitung pusat keseluruhan kelompok dan mengarahkan agen ke arah titik pusatnya.

$$\frac{\sum_{n \in N} n_{pos}}{|N|} \tag{6-3}$$

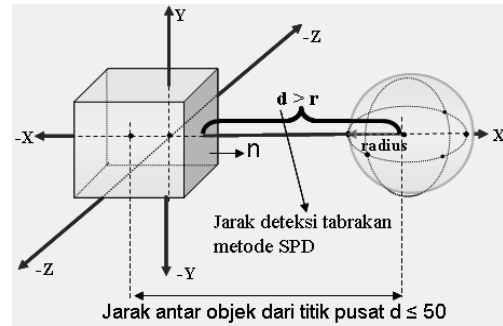
dimana:

- n_{pos} = Posisi Jumlah Agen
- agent_{pos} = Posisi Agen
- n_{vel} = Kecepatan Seluruh Jumlah Agen

3. Disain Sistem

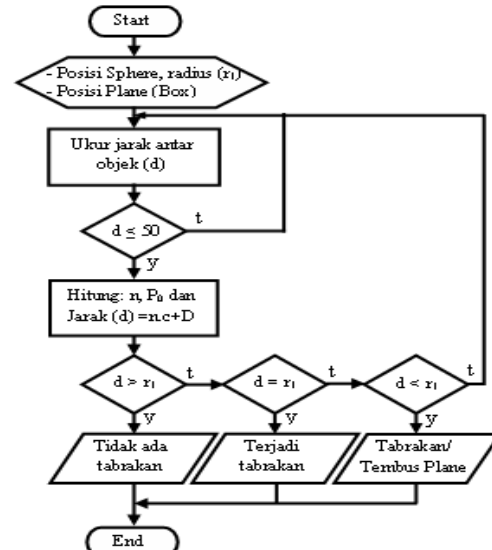
3.1. Desain Deteksi Tabrakan

Desain obyek-obyek yang telah dibuat diantaranya yaitu obyek *geometry box* dan *geometry sphere* yang akan diterapkan sebagai metode SPD.



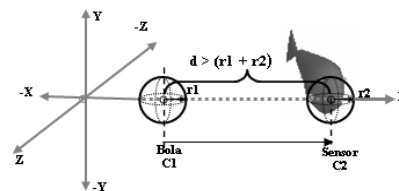
Gambar 4. Geometri *box* dan *sphere*

Algoritma dari metode SPD ditunjukkan pada gambar 5.



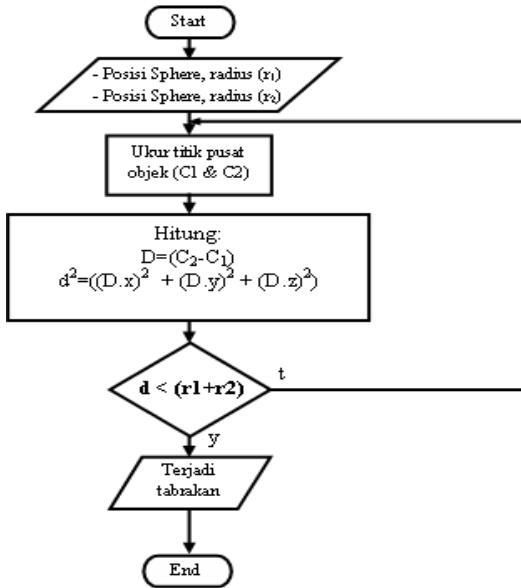
Gambar 5. Flowchart metode SPD

Obyek *geometry sphere* sebagai sensor pada mulut ikan dan *geometry sphere* sebagai makanan/bola merupakan penerapan metode SSD.



Gambar 6. Geometri *sphere* dan sensor *sphere*

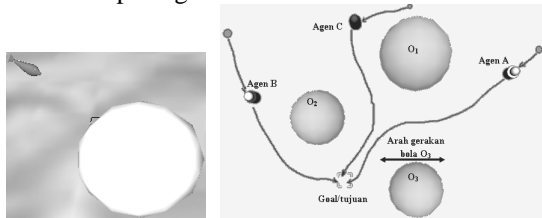
Algoritma dari metode SSD ditunjukkan pada gambar berikut:



Gambar 7. Flowchart metode SSD

3.2. Desain Penghindaran Tabrakan Antar Obyek

Proses penghindaran tabrakan antar obyek didisain menggunakan metode *repulsion*/tolakan. Penggunaan metode *potensial field* jenis *repulsive* pada *obstacle* berbentuk bola dengan *single* dan *multi* agen diperlihatkan pada gambar 8.

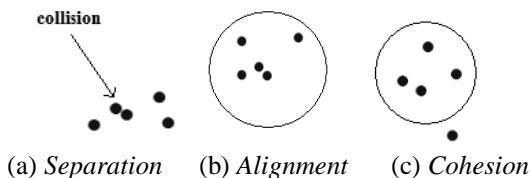


Gambar 8. Penggunaan *repulsive field*

3.3. Desain Penghindaran Tabrakan Antar Obyek pada Aplikasi *Boids*

3.3.1. Algoritma *Boid*

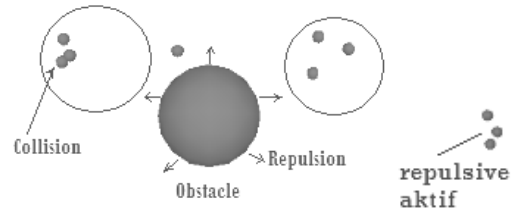
Penghindaran tabrakan antar agen dalam kelompok, penyesuaian kecepatan untuk pindah kearah kelompok dan mengarahkan agen ke titik pusat kelompok diilustrasikan seperti pada gambar 9.



Gambar 9. Ilustrasi *boid*

3.3.2. *Repulsive Field*

Penggunaan *repulsive field* pada aplikasi *boid*, diterapkan pada tiap agen dan *obstacle*. Pemberian *repulsive* pada agen bertujuan untuk melihat respon tolakan ketika terjadi *collision* baik antar agen maupun dengan penghalang.



(a) *Collision* antar agen (b) Tolakan antar agen

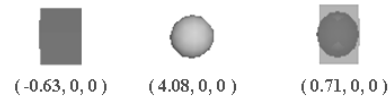
Gambar 10. *Boid* dan *obstacle*

4. Analisa

Pada bagian ini akan dilakukan pengujian berupa simulasi dan analisa dari metode *SPD*, *SSD*, *Repulsive Field* dan animasi *boid*.

4.1. Pengujian Metode *SPD*

Simulasi tabrakan kotak Vs Bola sebagai pengujian metode *Sphere-Plane Detection*.



Posisi kotak untuk sumbu x, y, dan z adalah (-0.63, 0, 0) dan posisi bola pada sumbu x, y, dan z yaitu (4.08, 0, 0). Radius bola = *geometry Sphere { radius .8 }*, Ukuran kotak = *geometry Box { size 1.5 1.5 1.5 }*, titik $p_0 = (-0.63 + (1.5/2)=0.12, N=(0.12, 0, 0)$.

Persamaan *plane* : $Ax + By + Cz + D = 0$

$$0.12+D= 0, \text{ maka } D = -0.12.$$

Unit vektor n adalah (persamaan 3):

$$n = \frac{N}{\|N\|} = \frac{(A, B, C)}{\sqrt{A^2 + B^2 + C^2}} = \frac{(0.12)}{\sqrt{(0.12)^2}} = 1$$

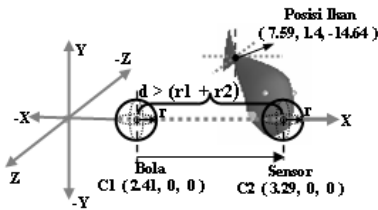
Jarak (persamaan 2) $d = 4.08 - 0.12 = 3.96$ atau $d > r$.

Dengan cara yang sama jarak kotak dengan bola ketika pada satu posisi (0.71, 0, 0) adalah $d = 0.71 - 1.46 = -0.75$ atau $d < -r$.

4.2. Pengujian Metode *SSD*

Simulasi tabrakan ikan Vs makanan/bola sebagai pengujian metode *SSD*. Bola dengan $r1 = \textit{geometry Sphere { radius .1}}$ berada pada sumbu X, Y, dan Z adalah (2.41, 0, 0), posisi ikan sumbu X, Y, dan Z

yaitu (7.59, 1.4, -14.64), dan posisi sensor dengan radius $r_2 = bSphereRadius .5$ berada pada koordinat (3.29, 0, 0). Hasil simulasinya sebagai berikut:



Perhitungan jarak (d) dari kedua obyek:

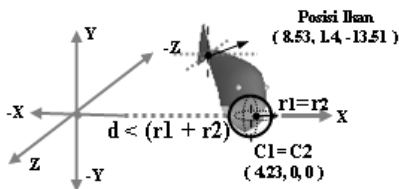
$$d = \sqrt{(D.x)^2 + (D.y)^2 + (D.z)^2} ; D = (C2 - C1)$$

$$D = (3.29 - 2.41) = 0.88 ; r_1 + r_2 = 0.1 + 0.5 = 0.6$$

Sehingga nilai d adalah:

$$d = \sqrt{(0.88 \times 3.29)^2} = 2.8952 \gg d > (r_1 + r_2).$$

Ketika posisi makanan satu posisi dengan sensor:



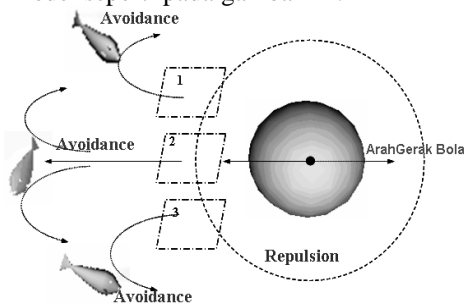
$$\text{Jarak } d = \sqrt{(0 \times 4.23)^2} = 0 \gg d < (r_1 + r_2).$$

4.3. Pengujian Metode Repulsive Field

Pengujian akan dilakukan dengan agen tunggal maupun multi agen.

4.3.1. Penghindaran tabrakan agen dan penghalang tunggal

Simulasi hambatan dinamis bentuk bola dengan nilai repulsion 500 bergerak kearah sumbu +x dan -x sedangkan sumbu y dan z konstan. Obyek hambatan mengakibatkan ikan bergerak menghindari tabrakan dengan model seperti pada gambar 11.



Gambar 11. Model gerakan avoidance

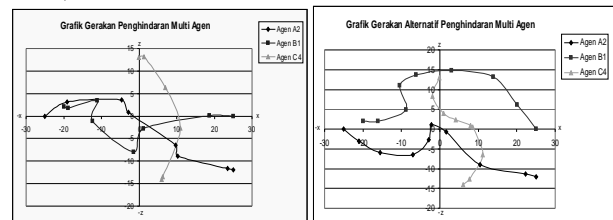
Tabel 1. Posisi terjadinya gerakan avoidance

No	Posisi Hambatan			Posisi Ikan/Goal		
	X	Y	Z	X	Y	Z
1	-3.14	0	7	-5.28	0	4.07
2	-2.15	0	7	-5.28	0	7
3	-2.84	0	7	-5.28	0	9.98

Pada posisi 1 ikan bergerak menghindari berputar menjauh menuju ke arah sumbu -z, hal ini disebabkan oleh $Z_{goal} \& \text{ ikan} < Z_{hambatan}$, sehingga *repulsion*/tolakan dari hambatan mendorong ke arah -z. Pada posisi 2 ikan bergerak menghindari menuju kearah sumbu -x karena nilai $Z_{goal} \& \text{ ikan} = Z_{hambatan}$, sehingga *repulsion* dari hambatan mendorong ke arah sumbu -x kemudian ke arah -z atau z. Pada posisi ke 3 ikan bergerak menghindari berputar ke arah sumbu +z, karena nilai $Z_{goal} \& \text{ ikan} > Z_{hambatan}$, sehingga *repulsion* dari hambatan mendorong ke arah sumbu z positif.

4.3.2. Penghindaran tabrakan multi obyek

Jumlah agen 100 (agen A=34, Agen B=33 dan agen C=33).



Gambar 12. Model gerakan avoidance multi agen

Vector dari P1(25, 0, -12) ke P2=(23.45,0, -11.66):

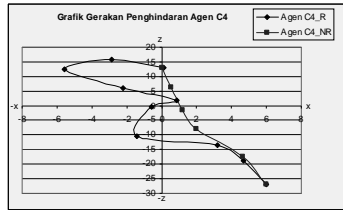
$$|V| = \sqrt{(V_x)^2 + (V_y)^2 + (V_z)^2}$$

$$|V| = \sqrt{2.4025 + 0.1156} = 1.58685$$

Vector dari posisi awal Pa (25, 0, -12) ke posisi akhir Pb (-25.05, 0, -0.07): $|V| = \sqrt{2505.0025 + 142.3249} = 51.4522$

4.3.3. Perbandingan penggunaan repulsive dan non repulsive

Jumlah Agen = 100, jumlah obstacle (disable & enable) repulsive field = 4 (3 Obyek Bola & 1 Obyek kotak).

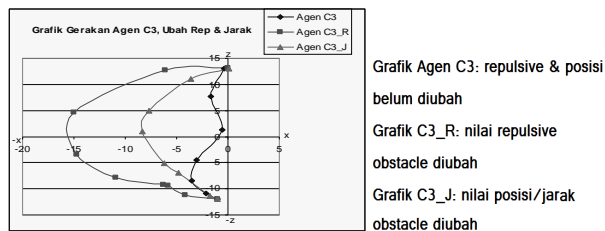


Gambar 13. Model gerakan *avoidance* menggunakan *repulsive & non repulsive*

Pada simulasi *Non Repulsive* agen (C4_NR) bergerak menuju tujuan mengabaikan *obstacle* tanpa *repulsive* dan cenderung bergerak linier/lurus sebagai jalur terdekat untuk mencapai tujuan. Sedangkan pada simulasi *repulsive* agen (C4_R) bergerak menuju tujuan menghindari *obstacle* dengan *repulsive* dan cenderung bergerak non linier.

4.3.4. Pengaruh Perubahan Nilai *Repulsive Field* dan Jarak Antar *Obstacle* Terhadap Gerakan Agen

Perubahan nilai *repulsive field* dan posisi dari *obstacle* mengakibatkan gerakan agen berubah seperti diperlihatkan pada gambar 14.



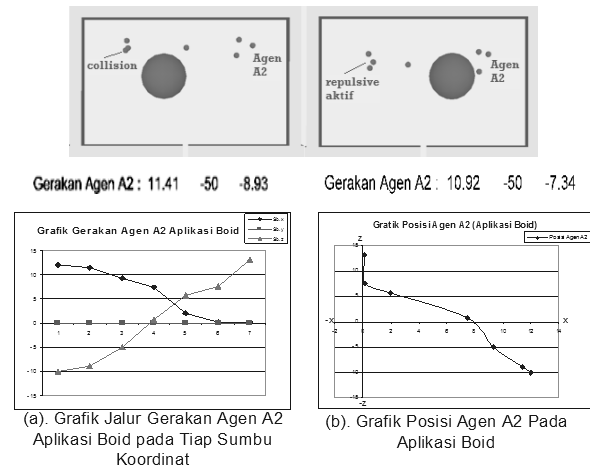
Gambar 14. Model gerakan *avoidance* hasil perubahan nilai *repulsive field* dan posisi *obstacle*

Grafik gerakan agen (C3_R) dari hasil perubahan nilai *repulsive* yang semakin besar mengakibatkan agen tidak dapat melewati celah diantara 2 *obstacle* karena *repulsive field* dari kedua *obstacle* tersebut berhimpit, sehingga jalur yang dilewati oleh agen untuk menuju tujuan semakin jauh. Grafik gerakan agen (C3_J) lebih pendek dari pada jalur (C3_R), ini disebabkan oleh semakin dekatnya jarak dari kedua *obstacle* sehingga *repulsive field* dari ke dua *obstacle* berhimpit.

4.4. Pengujian Penghindaran Tabrakan Antar Obyek pada Aplikasi *Boids*

Diketahui posisi setiap agen ($agen_{Pos}$) dan kecepatannya, nilai n_{Vel} adalah 33.5 dan $n_{Pos}=2, 0, -76$. Nilai *repulsive field* yang diberikan pada agen=1 dan *obstacle*=175, gambar 15 merupakan grafik gerakan

menghindari benturan antar agen maupun dengan penghalang.



Gambar 15. Gerakan *avoidance* pada aplikasi *boid*

Pada gambar 15, ketika terjadi *collision* antar agen *repulsive field* langsung berfungsi menolak sehingga kedua agen tersebut terpisah kembali dan tidak terjadi tumpukan dari 2 agen pada 1 posisi.

5. Kesimpulan

Dari hasil simulasi dan analisa data di atas dapat ditarik kesimpulan bahwa :

1. Besarnya nilai vektor $|V|$ dari posisi awal sampai ke tujuan untuk agen A2=51.4522, agen B1=45.0853 dan agen C4=27.7237
2. Tiga parameter yang mempengaruhi gerakan agen yaitu nilai *repulsive*, jarak dan gerakan penghalang.
3. Setiap agen memiliki lebih dari satu model jalur untuk menuju tujuan.
4. Tanpa *repulsive field* pada penghalang grafik gerakan agen cenderung mendekati linier sebagai jalur terdekat mencapai tujuan.
5. Pemberian *repulsive field* pada agen maupun *obstacle*, pengaturan posisi dan kecepatan agen menghasilkan tiga aturan perilaku *boid collision* (*separation*), *alignment* dan *cohesion* (*centering*).

Daftar Pustaka

- [1] Bergen Gino van den, (1999): "Collision Detection in Interactive 3D Computer Animation", Eindhoven University of Technology, University Press Facilities - Eindhoven.
- [2] Gomes Miguel, (1999): "Simple Intersection Test For Games", at http://www.gamasutra.com/view/feature/3383/simple_intersection_tests_for_games.php

- [3] Kuipers Benjamin, (2009): Lecture 7: “Potential Fields and Model Predictive Control”, CS 344R: Robotics.
- [4] Nadeau David R, Moreland John L.& Heck Michael M, (1998): “Introduction to VRML 97 – Lecturers”, University of California at San Diego.
- [5] PIGE, (2001): “Collision Detection”, at <http://www.edenwaith.com/products/pige/tutorials/collision.php>.
- [6] Parallel Graphics, (2011): “Object-To-Object Collision Detection Interface”, at <http://www.parallelgraphics.com/developer/products/cortona/extensions/collision/>
- [7] Reynolds Craig W, (1987): “Steering Behaviors For Autonomous Characters”, Sony Computer Entertainment America, Boulevard, California
- [8] Treuille Adrien, Cooper Seth, Popovic Zoran, (2007): “Continuum Crowds”, University of Washington, Electronic Arts.
- [9] Tu Xiaoyuan & Terzopoulos Demetri, (1994): “Artificial Fishes: Phisics, Locomotion, Perception, Behavior”, *Proc. ACM SIGGRAPH Conference*.