

ANIMASI 3D PERGERAKAN BURUNG TERBANG MENGUNAKAN MODEL *BOIDS*

Ara Sikenali¹, Atthariq²

Teknologi Informasi dan Komputer, Teknik Multimedia dan Jaringan, Politeknik Negeri Lhokseumawe,
Jalan banda Aceh-Medan KM 275,5.Buketrata – Lhokseumawe, 24301, P.O.Box 90 Telepon (0645)4278,
fax. 42785, Indonesia

E-mail: arasikenali38@gmail.com, atthariq.huzaifah@gmail.com,

Abstrak

Animasi 3D mulai berkembang dengan sangat pesat, simulasi perilaku burung merupakan tema riset dalam bidang komputer grafik yang memiliki banyak manfaat, antara lain untuk membuat simulasi pergerakan, pendeteksi tabrakan dan sebagai unsur pendukung realisme dalam animasi. Simulasi pergerakan kerumunan ini menggunakan metode algoritma boid dengan memperhitungkan jarak, kecepatan, dan pergerakan yang baik., saling menjaga jarak posisi dan ditambah perilaku menghindari hambatan secara berkelompok menggunakan algoritma *boids*.

Kata kunci : *Animation , modeling 3D, flocking bird, boid dan simulation.*

1. Pendahuluan

Perkembangan animasi 3D mulai berkembang dengan sangat pesat, baik itu di bidang perfilman, game, iklan tv, dan banyak bidang-bidang lainnya yang telah memanfaatkan animasi 3D.

Dalam beberapa tahun terakhir untuk simulasi kerumunan semakin meningkat untuk digunakan dalam berbagai permainan computer, film, pelatihan virtual, dan aplikasi pendidikan.

Gerakan dari sebuah kerumunan dalam alam seperti kehidupan burung, migrasi hewan darat, ataupun hewan laut (*school of fish*) memiliki gerakan-gerakan tertentu. Namun tipe gerakan-gerakan itu jarang ditemui dalam animasi komputer. Dalam pembuatan animasi visual grafik yang melibatkan kerumunan, akan sangat dimudahkan apabila kreator menggunakan algoritma dari distribusi *behavioral* model yang meliputi *boids*, *flock*, maupun *school of fish*.

Algoritma *boids* merupakan algoritma yang mendukung *flocking behavior* untuk mensimulasikan suatu kelompok agen, dimana dapat dilihat dari perilaku sekumpulan makhluk hidup yang berjalan membentuk pergerakan tertentu contohnya kerumun burung. Pergerakan agen berbasis *boids* dapat digambarkan dari pergerakan sekelompok agen yang terpisah yang kemudian dapat berkumpul untuk membentuk formasi baru. Formasi yang nantinya akan dijadikan objek oleh penulis adalah formasi kerumunan burung. Kerumunan burung memiliki karakteristik yang menarik dimana biasanya mereka berkerumunan dan terbang kearah yang sama. Kerumunan itu biasanya terpaksa terpisah oleh karena adanya *obstacle* (halangan) seperti pepohonan ataupun adanya predator di sekitar mereka. Algoritma *boids* yang digunakan harus dioptimasi

dengan *obstacle avoidance* sehingga pergerakan dan tingkah laku simulasi kerumunan burung tersebut terlihat lebih alami.

2. Metodologi Penelitian

2.1. Model Boid

Boids didefinisikan sebagai suatu jenis algoritma yang mempresentasikan gerak sebuah kawanan burung. *Boids* bereaksi hanya untuk *flockmates* dalam lingkungannya sendiri. Lingkungan ditandai dengan jarak (diukur dari pusat *boids*) dan sudut (diukur dari arah asal *boids*). Karakteristik dasar algoritma *boids* adalah gerak *boids* dihasilkan dari tiga aturan yaitu *cohesion alignment* dan *separation* ^[1].

a. Cohesion

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

$$\frac{\sum nPos}{|N|} \tag{1}$$

Dimana
 nPos = Posisi Jumlah Agent
 AgentPos = Posisi Agen
 nVel = Kecepatan Seluruh Jumlah Agen

b. Alignment

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

$$\sum_{n \in N} \text{Normalize}(nVel) \quad (2)$$

c. Pemisahan (Separation)

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

$$\sum_{n \in N} \text{Normalize}(agentPos - nPos) \quad (3)$$

2.2. Obstacle Avoidance

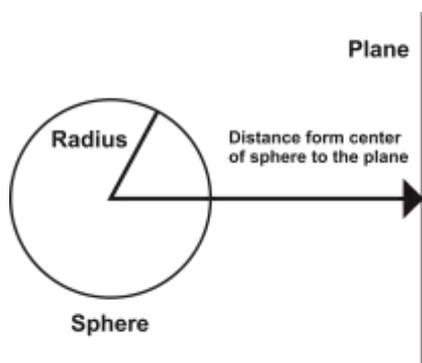
Perilaku menghindari hambatan (*obstacle avoidance*) memberikan kemampuan karakter untuk manuver di *environment* dengan menghindari hambatan sekitarnya. Implementasi dari perilaku *obstacle avoidance* berhubungan dengan penghindaran rintangan dimana tidak harus terjadi tabrakan. Silinder A dan B terletak disepanjang sumbu didepan karakter bola. Perilaku menghindari hambatan mempertimbangkan setiap kendala yang pada gilirannya mungkin menggunakan skema *portioning* spasial untuk menyisihkan jarak agar keluar dari hambatan dan menentukan apakah karakter bola bersinggungan dengan silinder^[1].

2.3. Collision detection (Deteksi Tabrakan)

Collision detection atau pendeteksian tumbukan adalah proses pengecekan apakah beberapa buah objek spasial saling bertumbuk atau tidak^[2,4,5].

2.3.1. Sphere Plane Detection

Metode *sphere plane detection* merupakan salah satu cara untuk mendeteksi tabrakan dalam dunia virtual 3D adalah dengan menggunakan metode *sphere plane detection*. Pada gambar 1 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* ^[3].



Gambar 1 Sphere Plane Detection

$$\text{Jarak} = \text{sphere.position} \cdot \text{plane.normal} \quad (4)$$

$$\text{atau } d = (C - P0) \cdot n \quad ; D = -P0 \cdot n \quad (4)$$

$$\text{maka: } D = n \cdot C + D \quad (5)$$

dimana: C = titik pusat obyek *sphere*,
p₀=setiap

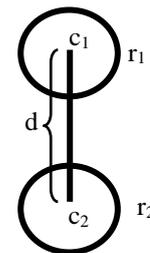
n=unit normal pada *plane*

$$n = \frac{N}{|N|} = \frac{(A,B,C)}{\sqrt{A^2+B^2+C^2}} \quad (6)$$

Persamaan plane dengan vektor normal N=(A,B,C) melalui titik (x, y, z). adalah A(x-x₀) + B(y - y₀) + C(z -z₀) = 0 atau Ax + By + Cz + D = 0, maka nilai D = -Ax₀ -By₀ -Cz₀. Dengan demikian besar dari unit normal vektor dengan notasi n.

2.3.2. 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.^[3]



Gambar 2 Sphere-Sphere Detection

Dapat ditulis dalam bentuk persamaan untuk menentukan jarak antara obyek di atas adalah :

$$D = (C2 - C1) \cdot n$$

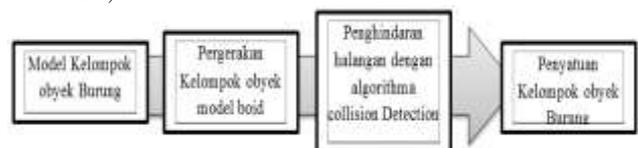
$$d = \sqrt{(D.x)^2 + (D.y)^2 + (D.z)^2}$$

Maka tabrakan akan terjadi jika :

$$d < (r1 + r2) \quad (7)$$

3. Desain Sistem

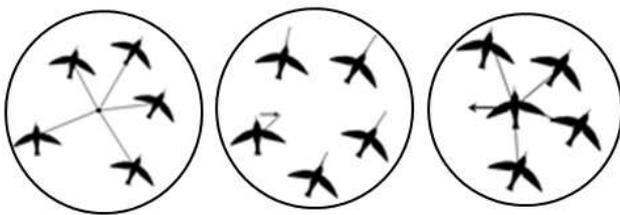
Pergerakan obyek kelompok burung yang akan menghindari penghalang dengan algoritma *collision detection*,



Gambar 3 Blok diagram Umum

Pada obyek kelompok burung pergerakan masing-masing obyek teratur dalam pergerakan sekelompok burung dan Algorihtma *collison detection* ini sebagai pendeteksi tabrakan atau pengindaran penghalang pada saat pergerakan kelompok obyek burung. Setelah melewati halangan dengan pergerakan natural setiap objek tidak bertabrakan antara satu dengan sama lain selanjut penyatuan tiap-tiap objek bergerak secara natural kembali.

3.1. Desain Pergerakan Pada Aplikasi *Boid*

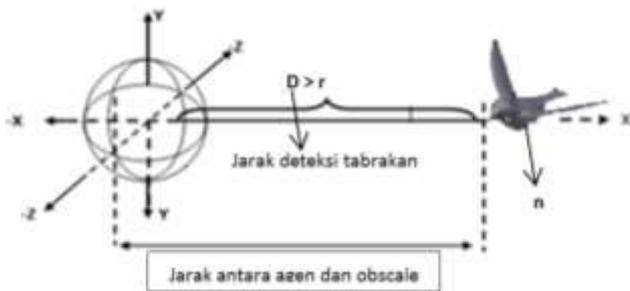


a. Cohesion b. Aligment c. Spration

Gambar 4 Ilustrasi *Boid* Untuk Menghitung Titik Koordinat Dan Kecepatan Dari Masing-Masing Agen.

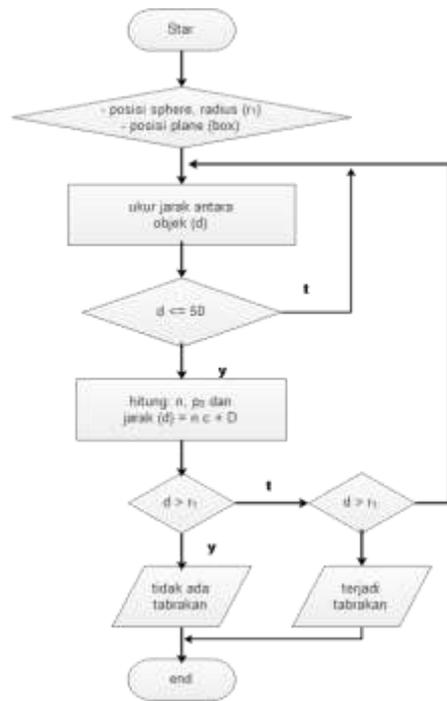
3.2. Desain Deteksi Tabrakan Agen dan Penghalang

Desain objek yang telah dibuat di antaranya yaitu *geometry box* dan *geometry sphere* yang akan di terapkan sebagai metode *sphere plane detection*.



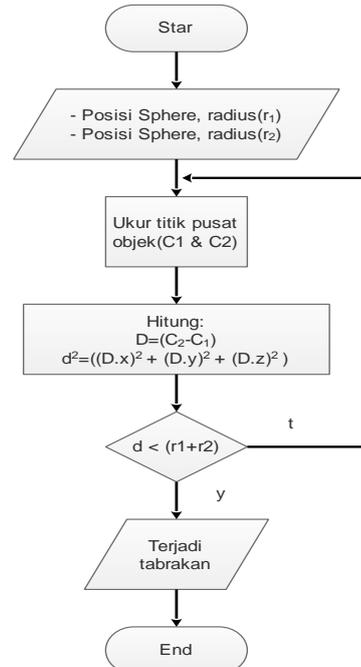
Gambar 5 Perhitungan Geometri *sphere* Terhadap *Boid*

Menghitung jarak antara pusat bola dan membandingkannya dengan jumlah jari-jari. Jika jarak ini kurang dari jumlah jari-jari, maka bola yang tumpang tindih.



Gambar 6 Flowchart *sphere plane detection*

Menentukan posisi *sphere* radius (r_1) dan posisi *plane*, mengukur jarak antara objek jika jarak objek lebih dari 50 maka mengukur kembali jarak objek, jika nilai objek kurang dari 50 maka menghitung n , P_0 dan jarak (d), jika nilai d lebih besar dari jari-jari maka tidak terjadi tabrakan dan jika d lebih kecil dari jari-jari maka terjadi tabrakan.



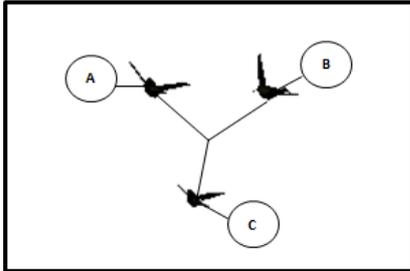
Gambar 7 Flowchart *sphere-sphere detection*.

Menentukan posisi *sphere*, radius(r_1) dan *sphere*, radius (r_2), mengukur ukuran titik pusat objek.

4. Hasil dan Pembahasa

4.1. Metode *Boid* Pada Pergerakan Burung Terbang

a. *Cohesion*



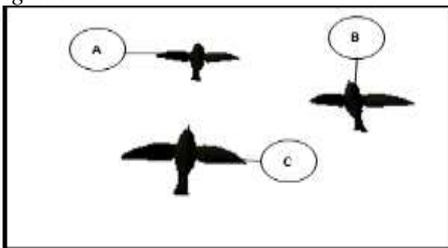
Gambar 8 Pergerakan *Cohesion* Antara *Boids*

Tabel 1 Contoh Nilai Keadaan *Boids* Untuk Titik Pusat Antara Tiga *Boids*

No	Titik Sumbu Agen		
	X	Y	Z
1	-54,4	-26.8	-20,7
2	-53,4	-9.8	-14.39
3	-55,4	-3.2	-11.04

Pada gambar 10 ketiga agen menentukan titik pusat dengan menambahkan semua posisi agen dan kemudian dibagi dengan membagi dengan vektor normal N dan nilai koordinat yang didapat ialah 3.02.

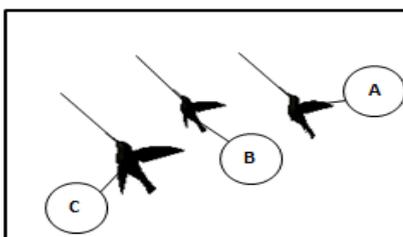
b. *Alignment*



Gambar 9 Pergerakan *Alignment* Antara *Boids*

Pada gambar 11 nilai rata-rata dari semua percepatan agen yang lain dan melakukan penyesuaian percepatan untuk pindah kearah kelompok sebesar 21.

c. *Separation*



Gambar 10 Pergerakan *Separation* Antara *Boids*

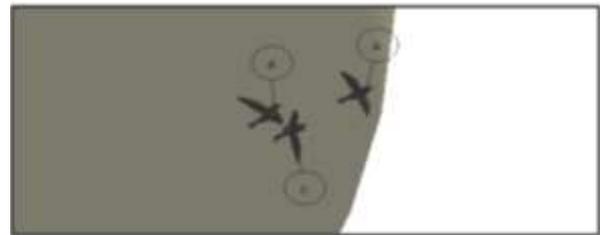
Tabel 2 Contoh Nilai Keadaan *Boids* Untuk posisi Antara Tiga *Boids*

No	Titik Sumbu Agen		
	X	Y	Z
1	-54,4	-26.8	-20,7
2	-53,4	-9.8	-14.39
3	-55,4	-3.2	-11.04

Pada kasus ketika posisi ketiga agen adalah identik, arah untuk mendorong keluar dari formasi agen tidak dapat didefinisikan dari perbedaan. Untuk lebih tepat, agen didorong keluar dalam beberapa arah ketika agen berada dalam posisi yang sama. Penyesuaian arah dan kecepatan untuk menghindari benturan sebesar - 54.15.

4.1. *Collision Detection* Pada Penghindaran Halangan

Collision detection merupakan pendeteksian tabrakan, dimana pada saat objek menghindari penghalang. Pada gambar 11 objek kerumunan burung menghindari penghalang secara teratur dan terarah dikarenakan menggunakan pensimulasian menggunakan *boid*.



Gambar 11 Pergerakan *Collision Detection* Untuk Menghindari *Obstacle*

4.1.1. Penghindaran tabrakan agen dan penghalang tunggal

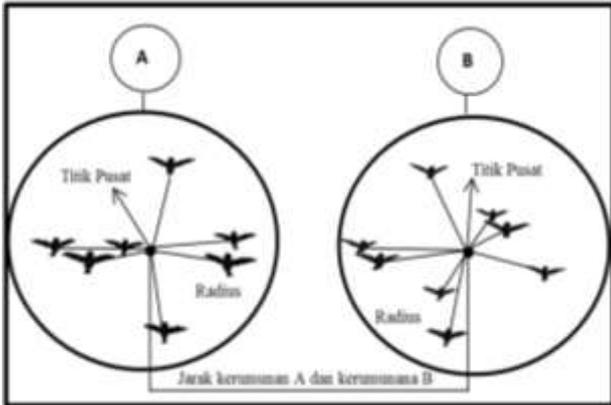
Tabel 3 Posisi terjadinya gerakan *avoidance*

No	Posisi Hambatan			Posisi Agen /Goal		
	X	Y	Z	X	Y	z
1	-3,95	0	0,21	-54,4	-26.8	20,7
2	-3,95	0	0,21	-53,4	-9.8	-14.39
3	-3,95	0	0,21	-55,4	-3.2	-11.04

Pada posisi A burung bergerak menghindar berputar menjauh menuju ke arah sumbu x, hal ini disebabkan oleh Z_{goal} & burung lebih kecil dari $Z_{hambatan}$, sehingga memberikan *repulsive field*/tolakan pada burung mengarah pada sumbu x. Pada posisi B burung bergerak menghindar menuju kearah sumbu $-x$ karena nilai Z_{goal} dan burung sama dengan $Z_{hambatan}$, sehingga memberikan *repulsive field*/tolakan pada burung mengarah sumbu $-z$. Pada posisi C burung bergerak menghindar berputar ke arah sumbu y, karena nilai Z_{goal} & burung lebih besar dari

Zhambatan, sehingga *repulsive field*/tolakan pada burung mengarah pada sumbu y positif.

4.2. Pergerakan Kawanan *Boids*



Gambar 12 Pengaturan Pergerakan Kelompok Kawanan *Boids* Dan Mengukur Titik Pusat *Boids*.

Tabel 4 Titik Pusat kerumunan

Kerumunan	Titik Pusat kerumunan			Radius
	X	Y	Z	
A	179.5	0	0	1
B	143.64	0	0	0.5

Menghitung jarak antara agen A dan agen B, jarak dinyatakan dengan d.

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

D menyatakan jumlah titik pusat dari kedua kerumunan sehingga titik pusat kedua kerumunan 6.1 Menghitung radius, dengan r1 radius kerumunan. A dan r2 radius kerumunan. B.1.5 jumlah radius dua kerumunan. Sehingga jarak antara kerumunan A dan kerumunan B adalah 192.15.

Pada gambar 16 kerumuna A berjumlah tujuh obyek agen dengan dengan titik koordinat pada sumbu X, Y, dan Z (25,4.0.0) dengan radius 1 dan kerumunan B berjumlah delapan agen dengan titik koordinat pada sumbu X, Y, dan Z (31,5.0.0) dengan radius 0.5.

5. **Simpulan**

Berdasarkan hasil analisa pada tugas akhir ini yang berjudul “Animasi 3D Gerak Burung terbang Menggunakan Model *Boids*” maka dapat diambil kesimpulan sebagai berikut :

1. Pengaturan posisi dan kecepatan agen menghasilkan tiga aturan perilaku boid *collision* (*separation*), *alignment* dan *cohesion* (pemusatan).

2. Pergerakan menghindari *obstacle* dipengaruhi oleh jarak antara agen dengan hambatan.
3. Setiap agen memiliki lebih dari satu model arah pergerakan dan pemberian *repulsive field* pada *obstacle* untuk memberikan tolakan pada agen agar menuju ke arah yang terdekat untuk sampai pada tujuan.

Daftar Pustaka

[1].Criag. Reynolds W. (1987) *Flocks, herds, and schools: a distributed behavior model*,*Proceedings of ACM SIGGRAPH*.

[2].Akuwan Saleh. (2011). Model Penghindaran Tabrakan Multi Obyek Menggunakan *Repulsive Field*, Fakultas Teknik Industri Institut Teknologi, Surabaya

[3]. Parallel Graphics, (2011). *Object-To-Object Collision Detection Interface* at. <http://www.parallelgraphics.com/developer/products/cortona/extensions/collision/>

[4]. Circle rectangle, (2009), Circular Collision Detection, at http://lazyfoo.net/SDL_tutorials/lesson19/index.php.

[5]. Game, (2011), Game Programming Video Tutorial in Python: Collision detection and cresolution <http://fossvideos.blogspot.com/2011/04/game-programming-video-tutorial-6-in.html> game control. Chalmers University of Technology, Sweden, Mei 2004.