

## PENGEMBANGAN EDITOR SISTEM PARTIKEL *STEREOGRAPHIC* UNTUK SIMULASI ASAP

<sup>1</sup>Bheta Agus Wardijono, <sup>2</sup>Mohammad Zulkifli Lubis

<sup>1</sup>Jurusan Sistem Informasi, STMIK Jakarta STI&K, Jalan Radio Dalam No.17, Jakarta Selatan

<sup>2</sup>Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma, Margonda  
Raya 100 Depok

bhetaagus@gmail.com, kifli.win@gmail.com

### ABSTRAK

Pemodelan asap merupakan salah satu masalah yang menarik dalam bidang grafik komputer, karena asap tidak memiliki sifat geometri yang berbentuk tetap. Salah satu cara untuk membuat simulasi asap diperlukan metode sistem partikel (*particle system*). Partikel Sistem adalah cara pemodelan obyek *fuzzy* untuk menghasilkan obyek-obyek yang partikelnya bergerak dinamis, seperti api, air, awan, asap. Untuk menghasilkan obyek seperti nyata, maka *stereographic* merupakan suatu teknik yang dapat digunakan, yaitu dengan cara menghadirkan dua gambar secara terpisah ke mata kiri dan kanan (*stereo*). Tujuan dari penelitian ini adalah untuk membuat editor sistem partikel yang dapat menampilkan simulasi asap secara *stereo*. Pembuatan aplikasi ini dilakukan dengan menggunakan bahasa pemrograman C++. Tahapan yang dilakukan dalam membangun aplikasi ini dimulai dari tahap penentuan kebutuhan aplikasi, dilanjutkan dengan perancangan aplikasi, dan perancangan antarmuka. Kemudian dilanjutkan dengan tahap pengkodean dengan menggunakan bahasa pemrograman yang telah ditentukan sebelumnya. Pada tahap terakhir akan dilakukan implementasi dan uji coba aplikasi. Berdasarkan hasil implementasi dan uji coba yang telah dilakukan, editor sistem partikel yang telah dikembangkan dapat menampilkan simulasi asap serta dapat menampilkannya secara *stereo*.

**Kata Kunci:** Simulasi Asap, Sistem Partikel, *Stereographic*.

### 1. Pendahuluan

Pemanfaatan komputer untuk digunakan sebagai media yang dapat menampilkan obyek fisik menjadi grafik 3 dimensi sudah sangat banyak. Bidang-bidang yang memanfaatkan hasil olah grafik 3 dimensi ini telah menyebar ke banyak bidang, antara lain: animasi dan simulasi, game komputer, edukasi, periklanan (advertising), dan film. Perkembangan ini didukung oleh berkembangnya perangkat lunak dan perangkat keras (khususnya adalah perangkat pengolah grafik), sehingga model-model tiga dimensi dapat dibuat sebaik dan serupa mungkin dengan obyek fisik asal.

Obyek grafik tiga dimensi (3D) dibangun oleh sejumlah titik, garis, vertex

sehingga membentuk obyek yang menirukan obyek fisik. Ada obyek yang dibuat bersifat statis, ada pula yang bersifat dinamis. Animasi dan simulasi biasanya digunakan untuk membuat obyek yang bersifat dinamis, sehingga gerakan dari obyek fisik dapat direpresentasikan. Simulasi yang berusaha menirukan obyek grafik berdasarkan pada sifat fisik dikenal dengan pemodelan berdasarkan pada fisik (*physical-based modeling*), seperti penelitian Nguyen D. (2002).

Obyek-obyek fisik yang bersifat dinamis adalah obyek yang dapat bergerak atau yang berubah keadaannya setiap waktu tertentu. Bukan hanya posisi, tetapi bentuk obyek dapat berubah sejalan dengan berjalannya waktu.

Obyek fisik yang bersifat dinamis ini antara lain, awan, asap, api, dan air yang mengalir.

Sejumlah metode untuk membuat animasi dan simulasi dari pergerakan obyek ini antara lain model Zona, metode CFD (Computational Field Dynamics), serta sistem partikel. Model Zona adalah merupakan model yang membagi model api atau asap menjadi zona yang terpisah, dimana kondisi di setiap zona diasumsikan seragam. Sedangkan, pemodelan api yang paling baru adalah penggunaan model CFD untuk memprediksi pertumbuhan api. Model ini juga sukses dalam gerakan pemodelan asap. FDS (Fire Dynamic Simulator) adalah merupakan salah satu simulator untuk menampilkan simulasi api dan asap secara 3 dimensi yang menggunakan metode CFD. Sistem Partikel merupakan cara pemodelan yang lain, yaitu memodelkan obyek secara fuzzy, dengan mengontrol sumber partikel (*emitter*) (Zhuo, 2008).

Salah satu teknologi yang memanfaatkan obyek grafik tiga dimensi yang dibangun dengan sistem partikel adalah *Virtual Reality* (VR) (Zhou, 2006). *Virtual Reality* digunakan agar *user* dapat berinteraksi dengan suatu lingkungan maya yang disimulasikan oleh komputer. Implementasi pemanfaatan model dari sistem partikel dalam bentuk simulasi evakuasi terlihat pada simulasi ketika terjadi bencana dari penelitian Ren, A (2008).

Lingkungan realitas maya ini umumnya menyajikan pengalaman visual, yang ditampilkan pada sebuah layar komputer atau sebuah penampil *Stereoscopy*. *Stereoscopy* atau *Stereographic* (juga disebut pencitraan 3-D) mengacu pada suatu teknik untuk menciptakan atau meningkatkan ilusi foto, film, atau citra dua dimensi dengan menghadirkan dua gambar secara terpisah ke mata kiri dan kanan. Dengan adanya *stereoscopy*, maka gambar atau citra dapat dilihat seperti kenyataan.

Dari sejumlah penelitian yang telah dituliskan sebelumnya, pada penelitian ini

difokuskan pada bagaimana membuat editor untuk kepentingan pembuatan obyek-obyek partikel sistem yang berdasarkan pada sifat fisiknya. Tujuannya adalah untuk memudahkan pengguna dalam memodifikasi obyek partikel sistem yang dibuat (khususnya adalah obyek asap). Pembuatan dan modifikasi obyek grafik dapat dilakukan dengan melakukan perubahan nilai parameter-parameter fisik yang dimiliki oleh obyek. Selanjutnya efek stereo akan ditambahkan sehingga obyek dapat dilihat seperti terlihat pada kenyataan yang sebenarnya.

### 1.1 Sistem Partikel (Particle System)

Sistem Partikel (*Particle System*) adalah cara pemodelan obyek grafik komputer dengan cara *fuzzy*, untuk obyek-obyek seperti api, air, awan, asap dimana obyek tersebut tidak memiliki permukaan yang halus dan tidak kaku (*rigid*) (Reeves, 1983). Sistem partikel berbeda dalam tiga cara representasi untuk sintesis suatu gambar, antara lain dalam hal:

1. Sebuah obyek tidak diwakili oleh satu set unsur-unsur permukaan primitif, misalnya, *polygon*, tetapi sebagai partikel primitif yang mendefinisikan volume.
2. Sebuah sistem partikel bukanlah entitas statis melainkan dinamis, dimana terdapat partikel yang diciptakan dan partikel yang dihancurkan/dilenyapkan.
3. Sebuah obyek diwakili oleh suatu sistem partikel. Proses stokastik digunakan untuk membuat, mengubah bentuk obyek dan mengubah penampilan dari obyek.

Partikel adalah satuan dasar dari suatu sistem partikel. Partikel memiliki sifat baik fisik maupun grafis. Sebuah partikel dapat dibentuk dari salah satu bentuk berikut: point, quad, quad tristrip, hexagon, dan line. Setiap partikel memiliki rentang hidup yang terkait dengannya. Rentang hidup adalah waktu hidup setiap

partikel yang telah dibuat. Semua partikel memiliki sifat, ukuran, alpha dan warna.

Dalam pemodelan obyek *fuzzy*, sistem partikel memiliki beberapa keuntungan. Pertama, sebuah partikel jauh lebih sederhana dari pada *polygon*. Oleh karena itu, dalam jumlah yang sama dari waktu komputasi dapat memproses lebih dan menghasilkan gambar yang lebih kompleks. Kedua adalah definisi model prosedural dan dikontrol oleh nomor acak, sehingga mendapatkan model yang sangat rinci tidak harus memerlukan banyak waktu. Ketiga, model partikel sistem yang memiliki waktu hidup, sehingga partikel dapat mengubah bentuk selama periode waktu tertentu.

## 1.2 Model Dasar Sistem Partikel

Untuk membuat sistem partikel diperlukan *modular emitter*, yaitu suatu modul yang berfungsi sebagai alat membuat keadaan awal dari sistem partikel. Suatu *modular emitter* biasanya digunakan untuk mengatur posisi awal partikel, kecepatan partikel dan berapa partikel yang harus dibuat dalam setiap frame. Untuk setiap partikel yang akan dibuat, partikel harus ditentukan sejumlah atribut, antara lain: posisi awal, kecepatan gerak awal, besar partikel, warna dari partikel, transparansi awal, bentuk partikel, dan waktu hidup partikel.

Sebuah sistem partikel adalah kumpulan dari beberapa model obyek partikel yang banyak. Untuk setiap *frame* dari urutan animasi langkah-langkah berikut dilakukan:

1. Partikel baru yang dihasilkan
2. Setiap partikel baru ditugaskan mengatur atribut sendiri
3. Setiap partikel yang ada ditentukan waktu untuk dihancurkan

4. Partikel yang tersisa diubah dan dipindahkan sesuai dengan sifat-sifat dinamis.
5. Gambar dari partikel – partikel hidup yang dimasukkan kedalam *frame buffer*.

Partikel sistem dapat diprogram untuk melaksanakan instruksi pada setiap langkah. Karena prosedural pendekatan ini dapat menggabungkan komputasi model yang menggambarkan penampilan atau dinamika obyek. Sebagai contoh, gerakan dan transformasi partikel dapat dikaitkan dengan solusi dari suatu sistem persamaan diferensial parsial, atau atribut partikel dapat diberikan pada dasar mekanika statistik.

### *Stereoscopy*

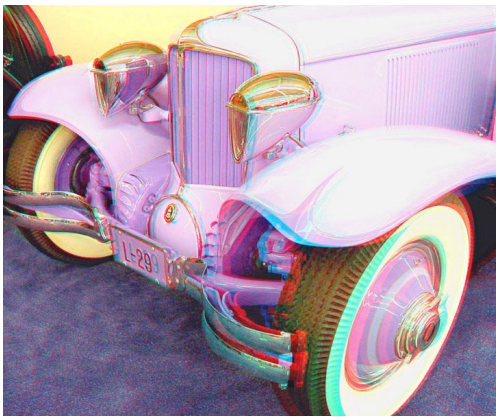
*Stereoscopy* (juga disebut pencitraan 3-D) mengacu pada suatu teknik untuk menciptakan atau meningkatkan ilusi pada foto, film, atau citra dua dimensi dengan menghadirkan dua gambar secara terpisah ke mata kiri dan kanan. Untuk mendapatkan gambar 3D dengan teknik *stereoscopy*, ada beberapa metode yang dapat dipakai seperti *anaglyphs*, *autostereoscopy*, dan lain - lain.

### *Anaglyphs*

*Anaglyphs* warna menggunakan salah satu dari sepasang filter warna komplementer untuk setiap mata. Filter warna yang paling umum digunakan adalah merah dan *cyan*. Mempekerjakan tristimulus teori, mata sensitif terhadap tiga warna primer, merah, hijau, dan biru. Filter merah mengakui hanya merah, sedangkan saringan *cyan* blok merah, melewati biru dan hijau (kombinasi biru dan hijau dianggap sebagai *cyan*).

Pengolahan gambar *anaglyph* memiliki lebih sedikit paralaks untuk mendapatkan

gambar yang lebih berguna bila dilihat tanpa filter. Gambar *anaglyph* digunakan untuk memberikan efek 3D, bila dilihat dengan kacamata dua lensa yang berbeda warna, seperti merah dan *cyan*. Gambar terdiri dari dua lapisan warna menghasilkan efek kedalaman. Biasanya subjek utama adalah di tengah, sedangkan latar depan dan latar belakang bergeser lateral dalam arah yang berlawanan. Gambar berisi dua gambar berwarna berbeda disaring untuk setiap mata. Contoh gambar *anaglyphs* yang menampilkan obyek stereo adalah seperti terlihat pada gambar 1 berikut ini.



Gambar 1. Contoh Gambar *Anaglyphs*  
Sumber: NVidia-Anaglyph (2012)

### 1.3 OpenSceneGraph (OSG) dan Delta3D

OpenSceneGraph adalah *application programmer interface* (API) yang bersifat *open source* (Martz, 2007; OpenSceneGrap, 2012; Riu, 2010). OpenSceneGraph ini adalah *tools middleware* dalam pembuatan desain data grafis. OpenSceneGraph (OSG) diciptakan untuk menangani *high performance graphic 3D* yang biasanya digunakan oleh para pengembang aplikasi dalam bidang-bidang tertentu seperti visual simulation, computer games, virtual reality, scientific visualization dan modeling. OSG berperan penting dalam pembuatan aplikasi 3D karena OSG merupakan perangkat lunak *middleware* yang posisinya berada diatas OpenGL, membuat OSG menyediakan level

rendering ke arah yang lebih tinggi, I/O, dan mengatur fungsi lainnya kedalam aplikasi 3D. Banyak aplikasi 3D membutuhkan fungsi tambahan dari *middleware library* daripada berinteraksi langsung dengan low-level rendering API.

Delta3D merupakan mesin game open source yang pada awalnya dirancang secara khusus untuk keperluan militer, baik untuk game maupun simulasi (Delta3D, 2012). Secara khusus Delta3D dirancang untuk memberikan solusi dengan biaya yang murah, mudah digunakan sebagai alternatif dari mesin game komersial dan eksklusif. Delta3D menyatukan modul-modul (*library*), seperti untuk keperluan, audio, scripting, GUI, sifat fisik, scene graph, jaringan, XML, dan lain-lain ke dalam suatu API (Application Programming Interface)

## 2. Metode Penelitian

Metode Penelitian dilakukan melalui sejumlah tahapan antara lain perancangan (meliputi perancangan alur menu dan antar muka), pemrograman (pengkodean dengan menggunakan perangkat lunak), implementasi dan uji coba.

### 2.1 Perancangan

Perancangan editor untuk simulasi asap ini dibagi menjadi dua tahap. Pada tahap pertama akan dilakukan alur aktifitas yang menggambarkan aktifitas yang dapat dilakukan oleh user dengan menggunakan diagram alur. Pada perancangan diagram alur ini akan dibagi menjadi dua bagian yaitu diagram alur untuk menu dan diagram alur untuk parameter sistem partikel.

Tahap kedua perancangan editor ini akan diakhiri dengan perancangan antarmuka (*interface*) yang akan digunakan dalam editor. Perancangan antarmuka ini digunakan sebagai

rancangan awal untuk pembuatan antarmuka user dengan editor.

Perancangan diagram alur digunakan untuk menentukan alur aktifitas yang dapat dilakukan oleh user. Perancangan diagram alur untuk aplikasi ini terbagi menjadi 2 bagian yaitu diagram alur untuk menu dan diagram alur untuk parameter sistem partikel.

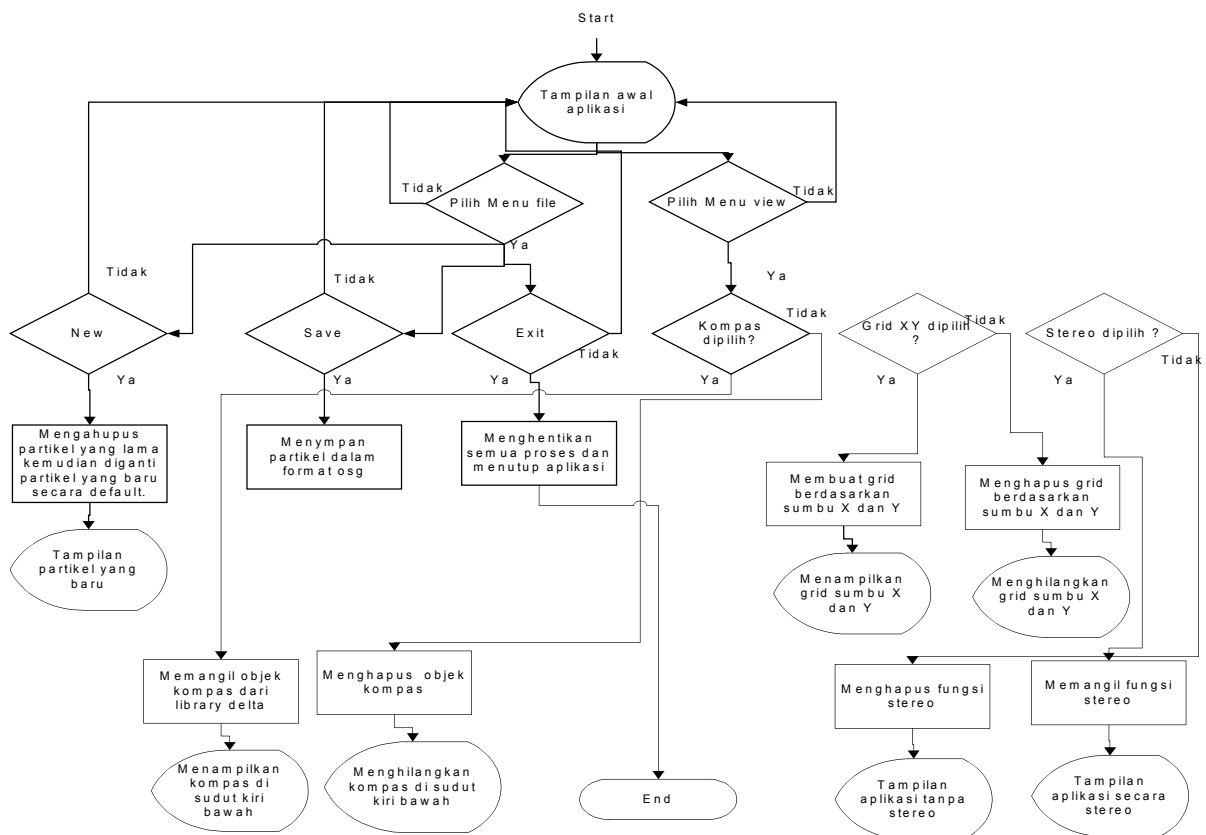
### 1. Diagram Alur Menu

Diagram alur menu adalah suatu diagram alur yang menggambarkan alur kegiatan yang dapat dilakukan oleh user pada aplikasi ini. Kegiatan utama yang dapat dilakukan oleh user dengan menggunakan aplikasi ini adalah memilih menu - menu yang dapat merubah tampilan output dari aplikasi ini.

User dapat memilih menu – menu yang tersedia pada editor. Menu utama terbagi menjadi dua bagian yaitu menu file sama menu

view. Pada menu file user dapat memilih sub – sub menu yang tersedia. Sub menu pertama adalah new. Sub menu new digunakan untuk membuat nilai-nilai parameter yang lama diubah menjadi nilai parameter secara default, lalu ditampilkan kedalam layer yang tersedia. Sub menu berikutnya adalah save. Sub menu save ini digunakan untuk menyimpan obyek partikel yang sudah diubah parameternya atau masih standar ke dalam sebuah format \*.osg. Sub menu yang terakhir adalah sub menu exit. Sub menu exit ini berfungsi untuk menutup editor sistem partikel.

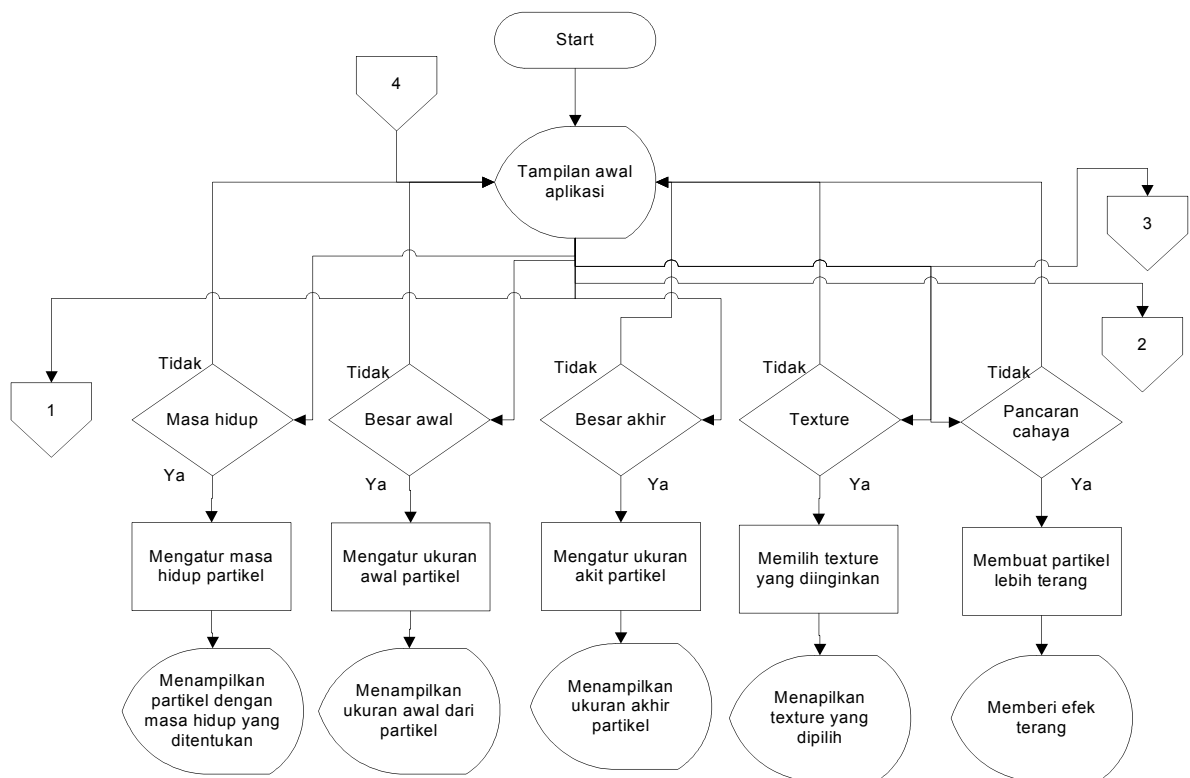
Menu berikutnya adalah menu view. Menu ini terbagi menjadi tiga buah sub menu. Sub menu yang pertama adalah sub menu kompas. Sub menu kompas ini berfungsi untuk menampilkan obyek kompas atau menghilangkan obyek kompas pada layer. Sub menu berikutnya adalah XYGrid. Sub menu



Gambar 2 Diagram Alur Menu

XYGrid ini berfungsi untuk menampilkan garis terhadap sumbu X dan sumbu Y, sub menu ini juga dapat menghilangkan garis yang ada. Sub menu yang terakhir adalah sub menu Stereo. Sub menu stereo ini berfungsi menghasilkan efek 3D pada layer, dengan menggunakan teknik *stereographic*

yaitu bagian umum, bagian pewarnaan, serta bagian *emiter*. Dari bagian – bagian tersebut jika parameternya diubah maka akan menghasilkan modifikasi pada obyek yang dihasilkan. Parameter dari sistem partikel yang digunakan untuk menghasilkan obyek yaitu: masa hidup partikel, besar partikel awal, besar partikel akhir, *texture*, dan pancaran cahaya.



Gambar 3. Diagram Alur Paramterr Umum

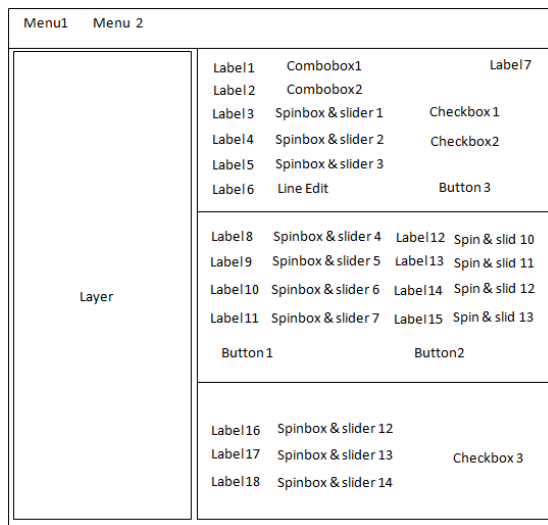
## 2. Diagram Alur Parameter Sistem Partikel

Diagram alur parameter partikel sistem adalah diagram alur yang digunakan untuk mengubah nilai – nilai parameter. Di dalam editor ini, parameter dibagi menjadi tiga bagian

### 2.2 Perancangan Antarmuka

Pada editor sistem partikel untuk simulasi asap ini hanya terdapat satu buah frame sebagai tampilan utama. Tampilan utama ini terdiri dari dua bagian pokok, yaitu frame yang digunakan

sebagai tempat menampilkan obyek (*frame*) serta sejumlah tombol dan input yang digunakan untuk mengubah parameter sistem partikel. Pengguna dapat memilih dan mengubah parameter serta melihat perubahannya pada obyek tersebut secara langsung (*real time*). Rancangan antarmuka ini terdiri dari *window viewer*, *layer*, *combo box*, *slider*, *spin box*, *push button*, dan *label*. Berikut adalah rancangan dari tampilan utama editor.



Gambar 4 Rancangan awal editor

### 2.3. Pemrograman

Pemrograman dilakukan dengan menggunakan bahasa pemrograman C++, serta menggunakan tools pengolah grafik Openscenegraph dan Delta3D. Untuk mendukung pembuatan Graphical User Interface (GUI) dipergunakan Qt (Qt, 2012).

*Layer* pada editor merupakan tempat melihat obyek atau partikel yang digunakan. Berikut ini adalah kode program untuk membuat partikel default yang ditampilkan didalam layer :

```
ParticleSystemLayer layer;
layer.mParticleSystem = newosgParticle
::ParticleSystem();
layer.mParticleSystem>setDefaultAttrib
utes("", true, false);
```

```
layer.mpParticle=newosgParticle::Parti
cle();
layer.mParticleSystem->setDefault
ParticleTemplate (*layer.mpParticle);
layer.mGeode = new osg::Geode();
```

Pada kode program tersebut sistem partikel dapat dibentuk, mulai dari penentuan atribut, penentuan tempat, dan penentuan besar suatu partikel berdasarkan pada fungsi yang ada pada Openscenegraph. Kode program `layer.mParticleSystem = new osgParticle :: ParticleSystem();` berfungsi untuk memasukan sistem partikel kedalam layer yang telah dibuat. Layer tadi akan menampilkan partikel dengan settingan default.

Setelah partikel terbentuk selanjutnya disetting *counter*, *placer* dan *shoter* yang berfungsi untuk menentukan posisi suatu partikel akan mengarah kemana. Sistem partikel dapat dirubah parameternya mulai dari ukuran partikel hingga warna dari partikel tersebut. Untuk melihat suatu partikel lebih nyata diperlukan teknik stereo. Pada teknik stereo user dapat melihat partikel dalam bentuk 3D. berikut ini adalah kode program untuk membuat teknik stereo dapat ditampilkan di dalam layer.

```
layer.mDisplaySetting =
osg::DisplaySettings::instance();
if (mDisplaySetting == NULL)
{
mDisplaySetting=newosg::DisplaySettings
;
mDisplaySetting->setDefaults();
GetCamera()->GetOSGCamera()->setDisplay
Settings(mDisplaySetting);
}
mDisplaySetting->setStereoMode
(osg::DisplaySettings::QUAD_BUFFER);
mDisplaySetting->setStereoMode
(osg::DisplaySettings::ANAGLYPHIC);
mDisplaySetting->setEyeSeparation
(0.01f);
mDisplaySetting->setStereo(true);
mDisplaySetting->setDisplayType
(osg::DisplaySettings::MONITOR);
```

Kode program diatas digunakan untuk menampilkan stereo ke dalam layer. `mDisplaySetting->setStereo(true);` digunakan untuk mengaktifkan tampilan stereo pada layer. Stereo yang ditampilkan berupa *anaglyphic*.

Stereo *anaglyphic* membuat partikel atau obyek terlihat lebih nyata. User dapat melihat jika menggunakan kacamata 3D.

Kedua kode program diatas merupakan inti dari editor ini. Kode program pertama digunakan untuk membuat partikel, sedangkan kode program kedua digunakan untuk membuat tampilan stereo.

### 3. Hasil Dan Uji Coba

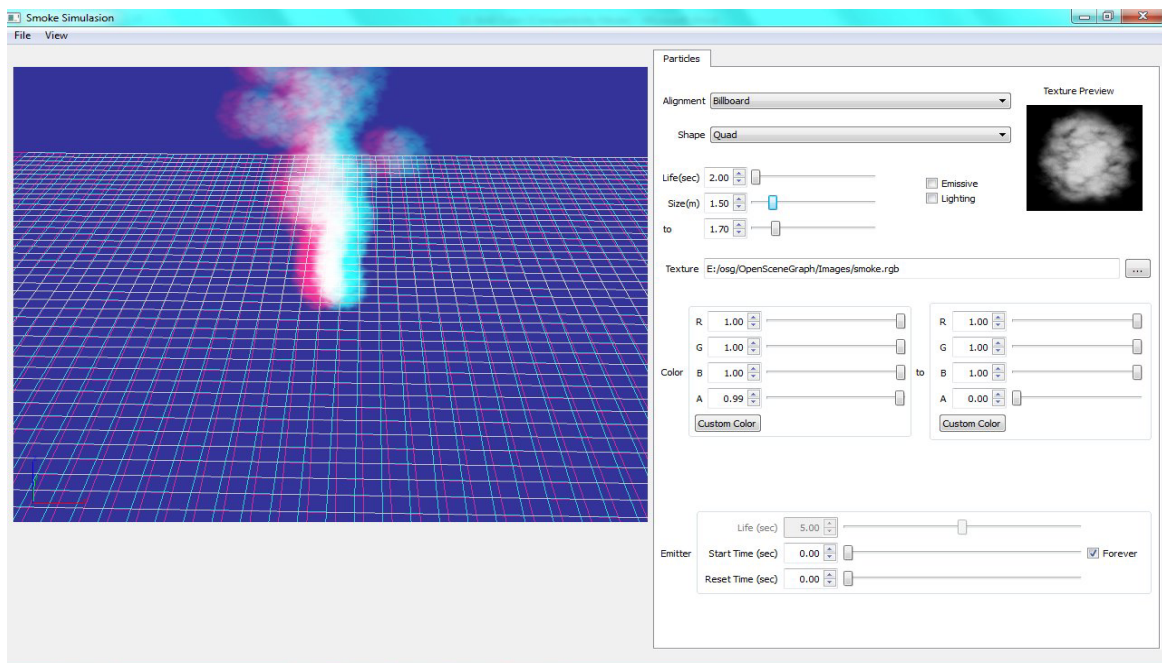
Hasil pembuatan editor *stereoscopy* yang telah sesuai dengan rancangan selanjutnya diimplementasikan. Implementasi ini dilakukan pada komputer dengan spesifikasi prosesor Intel Core i3 dengan memori 4GB. Menu yang ada dalam editor yang ada dapat digunakan untuk menampilkan obyek berupa partikel asap. Disamping itu fungsi dari menu dalam editor tersebut dapat diaktifkan, antara lain untuk mengatur masa hidup partikel, besar partikel, pancaran cahaya, serta texture. Fungsi stereo yang telah dikembangkan juga dapat digunakan sehingga efek stereo terlihat.

Uji coba stereo dilakukan untuk menampilkan obyek secara stereo. Ketika opsi dari viewer dipilih stereo, maka obyek akan menampilkan hasil stereo. Untuk melihat tampilan stereo ini diperlukan sebuah kacamata 3D agar partikel asap terlihat nyata. Hasil dan uji coba dari editor yang dikembangkan untuk menampilkan partikel secara stereo terlihat pada gambar 5.

### 4. Simpulan Dan Saran

#### 4.1 Simpulan

Editor sistem partikel telah berhasil dirancang, dibuat dan diimplementasikan untuk menampilkan obyek asap. Obyek yang telah dihasilkan dapat dimodifikasi melalui fungsi-fungsi yang digunakan untuk mengubah parameter pembentuk partikel. Berdasarkan hasil uji coba, editor ini telah dapat menghasilkan sistem partikel dalam berbagai bentuk, mulai dari bentuk *point*, *quad*, *quad tri strip*, *hexagon*, dan *line*.



Gambar 5. Tampilan stereo partikel asap dalam editor



Melalui editor ini dapat dimasukan *texture* asap sehingga partikel akan terlihat seperti asap serta mampu menampilkan simulasi asap secara stereo. Graphical User Interface yang telah dibangun mempermudah user untuk mengubah parameter-parameter pembentuk partikel.

#### 4.2 Saran

Untuk pengembangan selanjutnya, diharapkan aplikasi ini dapat memiliki fasilitas yang lebih lengkap lagi, seperti fasilitas untuk mengarahkan arah partikel, serta memindahkan atau merotasi partikel. Selain itu, dapat menampilkan efek 3D dengan teknik lain. Hal ini dikarenakan teknik *anaglyphs* dapat merusak warna asli dari obyek yang terbentuk.

#### Daftar Pustaka

1. Delta3D Game Engine, <http://www.delta3d.org/>
2. Martz, Paul. 2007. *OpenSceneGraph Quick Start Guide*. Computer Graphics Systems Development Corporation. California.
3. Nguyen, D. Q., Fedkiw, R., and Jensen, H. W. 2002. "Physically based modeling and animation of fire". In SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 721\_728.
4. NVidia-Anaglyph, <http://3dvision-blog.com/tag/nvidia-anaglyph/>
5. OpenSceneGraph, <http://www.openscenegraph.org/projects/osg/wiki/About/Introduction>
6. Qt, <http://www.qt.nokia.com>
7. Reeves, W.T., 1983. "Particle Systems – A Technique for Modelling a Class of Fuzzy Objects". ACM Transactions on Graphics.
8. Ren, A., Chen C., Shi J., and Zou L. 2008. "Application of Virtual Reality Technology to Evacuation Simulation in Fire Disaster". Tsinghua Science & Technology, Volume 13, Issue 5, Pages 674-680.
9. Riu, W., and Xuelie, Q., 2010. *OpenSceneGraph 3.0 Beginner's Guide*. Packt Publishing. Birmingham
10. Zhou, S., Sun, Y., Lu, L., and Chen, Z. 2006. "Fire Simulation Model Based on Particle System and Its Application in Virtual Reality" The 16th International Conference on Artificial Reality and Telexistence.
11. Zhuo, N., and Rao, Y. 2008. "Real time dense smoke simulation based particle system". In Intelligent Information Technology Application Workshops. IITAW '08. pp. 809-813.