

Virtualisasi Fenomena Rotasi Bumi dan Perbedaan Zona Waktu menggunakan Unity 3D

Denver Agpianka dan Anggraini Mulwinda

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang, Indonesia
kurma.holic@yahoo.com

Abstrak— Kemajuan dari perkembangan teknologi komputer grafis melahirkan sebuah media citra 3 dimensi. Salah satu perangkat lunak yang mampu digunakan untuk mengolah media 3 dimensi adalah Unity 3D. Kebebasan pengembangan perangkat lunak menggunakan Unity 3D memungkinkan pengguna untuk menerapkan konsep kehidupan nyata ke dalam Unity 3D. Konsep yang digunakan dalam penelitian ini adalah rotasi bumi dan perbedaan zona waktu karena fenomenanya tidak dapat dilihat secara langsung. Tujuan dari penelitian ini adalah untuk mengetahui apakah Unity 3D mampu membuat virtualisasi sesuai konsep. Metode yang digunakan dalam pengembangan perangkat lunak adalah Waterfall dan untuk implementasinya menggunakan eksperimen. Hasil yang diperoleh adalah Unity 3D mampu menghasilkan produk media 3 dimensi yang sesuai atau valid dengan materi fisika rotasi bumi dan zona waktu yang telah diujicobakan kepada guru SMP mata pelajaran Fisika.

Kata kunci— teknologi komputer grafis, virtualisasi

I. PENDAHULUAN

Awal mula dari perkembangan teknologi komputer hanya dibuat dengan tujuan untuk menghitung namun seiring berjalannya waktu dan kebutuhan dari berbagai pengguna maka komputer dibuat dengan berbagai macam fungsi salah satunya adalah multimedia.

Dari perkembangan segi multimedia telah melahirkan banyak produk. Salah satu produk yang banyak diminati adalah media citra 3 dimensi. Media ini dapat dibuat berkat kecanggihan dari perangkat keras pengolah citra atau GPU (*Graphic Processing Unit*).

Dalam pengembangannya, media 3 dimensi yang mampu dilihat dari berbagai sisi ini dapat dibuat dengan menggunakan berbagai perangkat lunak. Salah satu perangkat lunak yang masih terbilang baru dan perkembangannya cukup cepat adalah Unity 3D.

Sejatinya, perangkat lunak Unity 3D ini didesain untuk pengembangan game atau disebut dengan perangkat lunak *game engine*. Namun dari segi kebebasan pengembangan perangkat lunak menggunakan Unity 3D ini memungkinkan pembuatan virtualisasi yang konsepnya diambil dari kehidupan sehari-hari. Salah satu fenomena yang menarik untuk divirtualisasikan adalah fenomena rotasi bumi dan pembagian zona waktu akibat dari fenomena tersebut.

Fenomena ini dipilih karena keterbatasan manusia dalam menyaksikan fenomena ini secara langsung meskipun sudah dirasakan akibat dari rotasi bumi. Ada beberapa cara untuk mengetahuinya yaitu melihat secara langsung dari luar angkasa atau belajar melalui mata pelajaran Fisika yang diberikan di bangku pendidikan. Melihat langsung adalah bukan solusi yang bisa diterapkan secara massal oleh karena itu salah satu solusinya adalah proses belajar langsung akan tetapi kebanyakan media virtualisasi belum mampu memberikan gambaran yang seperti nyata.

Penelitian yang dilakukan oleh Yongzhe Xu dkk menggunakan Unity 3D telah berhasil membuat simulasi kebakaran gedung pencakar langit. Tujuannya adalah mencari tahu seberapa besar efek dari kebakaran jika terjadi di gedung bertingkat tinggi menggunakan fitur *Particle System*. Hal ini menyakinkan peneliti bahwa Unity 3D mampu menghasilkan virtualisasi yang sesuai dengan fenomena di dunia nyata.

Untuk membuat virtualisasi rotasi bumi dan zona waktu, penulis menggunakan fitur *animation* untuk animasi rotasi bumi dan *scripting* untuk membuat jam dunia. Dengan memanfaatkan fitur-fitur ini, penulis berharap Unity 3D sudah mampu membuat media virtualisasi rotasi bumi dan zona waktu.

II. METODE

Metode yang digunakan dalam penelitian ini terdiri atas dua tahapan utama yaitu tahap pengembangan perangkat lunak kemudian dilanjutkan dengan tahap implementasi perangkat lunak. Kedua metode ini digunakan untuk mengetahui keberhasilan pengembangan perangkat lunak baik secara fungsional maupun teori.

Pada tahap pengembangan perangkat lunak, metode yang digunakan adalah Waterfall yang diartikan oleh Pressman (2010 : 39) sebagai sebuah sistematis, pendekatan contoh pada pengembangan perangkat lunak yang dimulai dari kebutuhan pengguna dan pengerjaan melalui perencanaan, pemodelan, pembangunan, dan penyebaran, memuncak pada dukungan teruser-menerus dari perangkat lunak yang telah selesai dibuat.

Tahapan dalam metode *Waterfall* adalah:

1. Rekayasa dan Pemodelan Sistem Informasi.
2. Analisis Kebutuhan Perangkat Lunak.
3. Desain.
4. Generasi Kode.
5. Pengujian.

Selanjutnya ketika perangkat lunak sudah diujikan untuk mengetahui kinerjanya secara fungsional, langkah berikutnya adalah masuk ke metode implementasi. Dalam metode ini, instrumen yang digunakan adalah angket dan essay. Angket diberikan untuk ahli materi, yaitu guru, untuk mengukur kesesuaian teori yang diterapkan di dalam perangkat lunak virtualisasi. Analisis data untuk angket menggunakan skala *Linkert*, dengan kategori sebagai berikut :

1. Sangat Setuju 4
2. Setuju 3
3. Tidak Setuju 2
4. Sangat Tidak Setuju 1

Dengan menggunakan skala tersebut maka dapat digunakan untuk mencari *rating scale*-nya.

Kemudian implementasi dilanjutkan dengan uji coba produk, dimana perangkat lunak secara langsung didemonstrasikan kepada pengguna yaitu siswa. Setelah itu dilakukan uji tingkat pemahaman materi menggunakan instrumen essay berisi soal yang disesuaikan dengan teori yang terdapat dalam perangkat lunak virtualisasi.

Hasil dari essay dianalisis untuk mencari berapa banyak siswa yang tuntas dalam memahami materi dan tingkat pemahaman siswa dari setiap soal menggunakan teknik analisis butir soal.

III. HASIL DAN PEMBAHASAN

Hasil untuk tahapan pengembangan perangkat lunak :

1. Rekayasa dan Pemodelan Sistem Informasi
 - Kebutuhan yang harus ada dalam perangkat lunak :
 - a. Virtualisasi harus melibatkan bumi sebagai objek utama dan matahari sebagai sumber cahaya.
 - b. Bumi harus mendapat cahaya dari matahari dengan penempatan yang sesuai.
 - c. Bumi harus berotasi dengan tampilan jam yang sesuai.
 - d. Pengguna dapat berkeliling bumi.
 - e. Pengguna dapat menampilkan dan menyembunyikan tampilan jam dan negara.
 - f. Pengguna dapat mengatur kecepatan rotasi.
 - g. Analisis Kebutuhan Perangkat Lunak
 - h. Analisis Kebutuhan Perangkat Lunak

2. Analisis Kebutuhan Perangkat Lunak
 Untuk pengembangan perangkat lunak menggunakan Unity 3D :

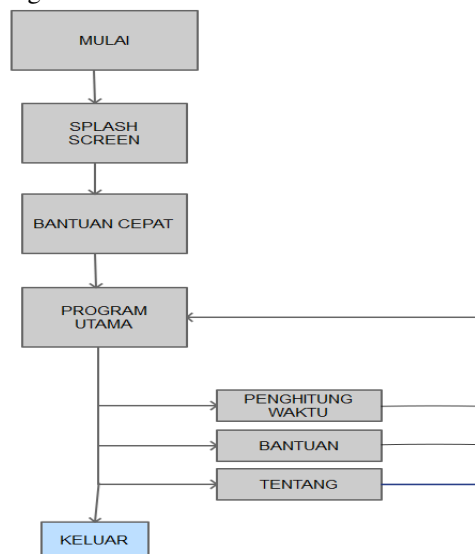
Sistem Operasi	Windows XP SP2+, 7 SP1+, 8, Mac OS X 10.6+
GPU / Kartu Grafis	Kartu Grafis dengan DX9 (Shader Model 2.0)

Untuk menggunakan perangkat lunak virtualisasi :

Sistem Operasi	Windows XP atau versi lebih tinggi
GPU / Kartu Grafis	Kartu Grafis dengan dukungan DX9 dan Pixel Shader 2.0
Resolusi Layar	1366 68 pixel

3. Desain

a. Diagram Alir



Gambar 1. Diagram Alir Perangkat Lunak

b. Layout GUI

• Splash Screen



Gambar 2. Splash screen

• Bantuan Cepat



Gambar 3. Bantuan cepat

• Program Utama



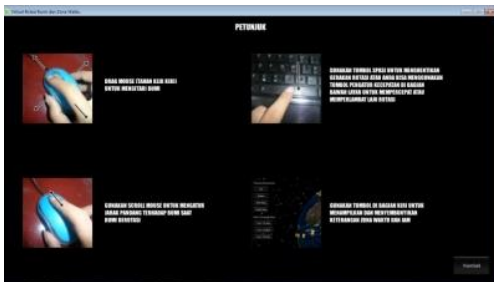
Gambar 4. Program Utama

• Penghitung Waktu



Gambar 5. Penghitung Waktu

• Bantuan



Gambar 6. Bantuan

• Tentang



Gambar 7. Tentang

4. Desain Grafik 3D

- a. Bumi
- b. Matahari
- c. Jam Digital

5. Generasi Kode

- a. GUI Script
- b. Timer Script

Script ini digunakan untuk menampilkan jam digital yang diterapkan di beberapa negara di objek bumi. Perhitungan disesuaikan dengan lokasi atau pembagian zona waktu GMT.

c. Animation

Animasi menggunakan script khusus berformat .anim. Fungsi ini dibutuhkan untuk mempresentasikan gerakan rotasi bumi. Kecepatan rotasi disesuaikan dengan timer script.

6. Pengujian

a. Pengujian Fungsional

Pengujian perangkat lunak menggunakan metode *blackbox* yang secara khusus digunakan untuk mencari kesalahan fungsi yang ada di dalam program yang sudah dibuat. Berikut adalah tabel hasil pengujian fungsional :

TABEL I. HASIL PENGUJIAN FUNGSIONAL

No.	Aspek yang Diuji	Jenis Pengujian	Hasil Pengujian
1.	Memulai Perangkat Lunak	Expected Input	Sesuai
2.	Menggunakan Perangkat Lunak	Expected Input	Sesuai
3.	Speed Control Button	Expected Input	Sesuai
4.	Side Menu	Expected Input	Sesuai
5.	Penghitung Waktu	Expected Input	Sesuai
6.	Bantuan	Expected Input	Sesuai
7.	Tentang	Expected Input	Sesuai

b. Pengujian Khusus

Pengujian ini ditujukan untuk mencari kesesuaian output dengan penghitungan yang diterapkan di dalam menu penghitung waktu dan jam dunia.

• Pengujian jam

TABEL II. PENGUJIAN SAMPEL JAM DUNIA

No.	Nama Negara	Wilayah GMT	Output Jam	Selisih	Visual	Hasil Pengujian
1.	Alaska	GMT-9	07:59	09:00	Siang	Sesuai
2.	Mexico	GMT-6	10:59	06:00	Siang	Sesuai
3.	Venezuela	GMT-5	11:59	05:00	Siang	Sesuai
4.	Argentina	GMT-3	13:59	03:00	Siang	Sesuai
5.	Inggris	GMT 0	16:59	00:00	Sore	Sesuai
6.	Arab Saudi	GMT+3	19:59	03:00	Malam	Sesuai
7.	Pakistan	GMT+5	21:59	05:00	Malam	Sesuai
8.	Bangladesh	GMT+6	22:59	06:00	Malam	Sesuai
9.	Indonesia (Jakarta)	GMT+7	23:59	07:00	Malam	Sesuai
10.	China	GMT+8	00:59	08:00	Malam	Sesuai
11.	Jepang	GMT+9	01:59	09:00	Malam	Sesuai
12.	Australia (Tasmania)	GMT+10	02:59	10:00	Malam	Sesuai

• Pengujian Penghitung Waktu

Rumus penghitung waktu :

$$f = t + (z_1 - z_2)$$

Dimana :

f = hasil hitungan

t = nilai jam yang

diinput

z_1 = zona GMT Anda

z_2 = zona GMT yang dihitung

TABEL III. PENGUJIAN PENGHITUNG WAKTU

No.	t1	Z1	Z2	f	Hasil pada Perangkat Lunak	Hasil Pengujian
1.	00:00	0	-12	12:00	12:00	Sesuai
2.			-11	13:00	13:00	Sesuai
3.			-10	14:00	14:00	Sesuai
4.			-9	15:00	15:00	Sesuai
5.			-8	16:00	16:00	Sesuai
6.			-7	17:00	17:00	Sesuai
7.			-6	18:00	18:00	Sesuai
8.			-5	19:00	19:00	Sesuai
9.			-4	20:00	20:00	Sesuai
10.			-3	21:00	21:00	Sesuai
11.			-2	22:00	22:00	Sesuai
12.			-1	23:00	23:00	Sesuai
13.			0	00:00	00:00	Sesuai
14.			+1	01:00	01:00	Sesuai
15.			+2	02:00	02:00	Sesuai
16.			+3	03:00	03:00	Sesuai
17.			+4	04:00	04:00	Sesuai
18.			+5	05:00	05:00	Sesuai
19.			+6	06:00	06:00	Sesuai
20.			+7	07:00	07:00	Sesuai
21.			+8	08:00	08:00	Sesuai
22.			+9	09:00	09:00	Sesuai
23.			+10	10:00	10:00	Sesuai
24.			+11	11:00	11:00	Sesuai
25.			+12	12:00	12:00	Sesuai
26.			+13	13:00	13:00	Sesuai
27.			+14	14:00	14:00	Sesuai

7. Hasil untuk pengujian implementasi :

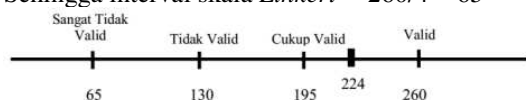
Angket validasi materi diberikan kepada 5 guru fisika SMP. Jumlah skor kriterium (bila seluruh aspek yang dinilai mendapat skor tertinggi) = $4 \times 13 \times 5 = 260$

Jumlah skor pengumpulan data = 224.

Prosentase menurut 5 responden adalah :

$$= 224/260 \times 100\% = 86,15 \%$$

Sehingga interval skala *Linkert* = $260/4 = 65$



Termasuk dalam kategori valid.

Dari hasil pengembangan perangkat lunak dapat diperoleh sebuah produk virtualisasi dari Unity 3D yang berjalan sesuai dengan input dan hasil penghitungan jam dunianya sudah sesuai.

Tingkat responsibilitas perangkat lunak bergantung pada spesifikasi komputer, mengingat pada hasil analisis kebutuhan perangkat keras dan lunak memerlukan spesifikasi yang lebih seperti dukungan kartu grafis. Jika kebutuhan minimal tidak dipenuhi maka hasilnya tidak maksimal dan akan dijumpai *lagging* atau terjadinya jeda pada pergerakan grafik dan efek *lighting* atau pencahayaan dari matahari yang kurang terang.

Hasil validasi dari lima responden dari guru mata pelajaran fisika menghasilkan kategori valid dengan prosentase kevalidan 86,15%.

IV. SIMPULAN

Dari hasil penelitian dapat disimpulkan bahwa Unity 3D sebagai game engine mampu dipakai untuk mengembangkan perangkat lunak yang berbasis pada teori kehidupan sehari-hari khususnya untuk fenomena rotasi bumi dan zona waktu. Perangkat lunak yang dihasilkan dari Unity 3D dinilai valid oleh guru Fisika dan informasi yang disampaikan melalui program virtualisasi ini sudah mampu dipahami oleh ahli materi.

REFERENSI

- [1] Azwar, Saifuddin. 2004. Metode Penelitian. Yogyakarta: PUSTAKA PELAJAR.
- [2] Barus P.K, dkk. 1993. FISIKA 1 Untuk Sekolah Lanjutan Tingkat Pertama kelas 1. Jakarta: Departemen Pendidikan dan Kebudayaan.
- [3] Barus P.K, dkk. 1996. FISIKA 1 Untuk Sekolah Lanjutan Tingkat Pertama kelas 1. Jakarta: Departemen Pendidikan dan Kebudayaan.
- [4] Hadi, Sutrisno. 2004. Metodologi Research Jilid 2. Yogyakarta: ANDI.
- [5]
- [6] Hidayati, Rima. (2015). Konsep Virtualisasi. Available at <http://ilmukomputer.org/2015/07/04/konsep-virtualisasi/> [accessed 08/26/2014].
- [7] Kartono, Agus. 2007. Seribu Pena Fisika untuk SMP/MTs Kelas IX. Penerbit Erlangga.
- [8] Perbedaan Zona Waktu pada Jam Dunia. Online at <http://www.anneahira.com/jam-dunia.htm> [accessed at 08/26/2014]
- [9] Prasetyo Bambang & Jannah, Lina Mifathul. 2007. Metode Penelitian Kuantitatif Teori dan Aplikasi. Jakarta: PT RajaGrafindo Persada.
- [10] Pressman, Roger. S. 2002. Rekayasa Perangkat Lunak: Pendekatan Praktis Jilid Pertama. Yogyakarta: ANDI.
- [11] Rolliawati, Dwi. (2015). Game Engine. Available at <http://rolliawati.dosen.narotama.ac.id/> [accessed 07/16/2014].
- [12] Sommerville, Ian. 2003. Software Engineering Jilid 2. Jakarta: Penerbit Erlangga