



SOLUSI PERMASALAHAN KOMPUTASI SKALA BESAR PADA SOFTWARE ADOBE FLASH DALAM PERANCANGAN MEDIA PEMBELAJARAN PENCARIAN *EULERIAN CIRCUIT**

Reza Wafdan*, Mahyus Ihsan, Marwan Ramli dan Hafnani

Grup Riset Multimedia Jurusan Matematika, Magister Matematika
Universitas Syiah Kuala, Darussalam Banda Aceh
*Email: redjawavdhan11@gmail.com

Abstract. *Adobe Flash is a software that is used to build an interactive contents that can be attached in other things, such as presentations, games and e-learning. The Eulerian Circuit Search Instructional Media is one of examples in using Adobe Flash in building an interactive instructional media to solve graph theory problems. Visually, Adobe Flash is better than other programming software in general and it is the most suitable one for graph theory, as it uses graphics and visuals to solve problems. The weakness of Adobe Flash is when doing large scale calculations; such as when testing algorithms to find Eulerian circuits. Then an idea came about to complete the large scale calculations in Adobe Flash, which is commonly known as script time limit, with using other methodologies by changing script time limit on the settings, algorithms, partition the processes, and utilize the timers within the objects in the actionscript that handles the search algorithms of searching Eulerian circuit.*

Keywords: Adobe Flash, e-learning, Eulerian circuit, actionscript, script time limit

I. PENDAHULUAN

Media pembelajaran elektronik atau yang sering dikenal dengan *e-learning* memiliki potensi untuk mengubah penonton (*audience*) dari penerima informasi pasif (*passive recipients*) menjadi peserta yang aktif (*active participants*) dalam proses belajar mengajar dengan memanfaatkan multimedia interaktif [1, 2]. Peran multimedia dalam bidang pendidikan ini juga berperan sebagai alat bantu yang dapat memperkaya nilai belajar secara konvensional melalui pengayaan konten dan pengembangan teknologi pendidikan yang lebih menarik dan interaktif. Dengan alat bantu belajar ini, setiap orang dapat belajar kapanpun dan dimanapun, sehingga dapat menghilangkan sikap pasif dalam proses belajar. Media pembelajaran elektronik merupakan solusi yang tepat dalam menyajikan sebuah materi pembelajaran yang rumit jika menggunakan metode konvensional. Salah satu materi pembelajaran yang sangat tepat untuk diterapkan dalam media pembelajaran elektronik

yaitu permasalahan perhitungan jumlah *Eulerian circuit* pada sebuah graf sederhana yang merupakan permasalahan dalam bidang teori graf. Sebuah graf sederhana $G = (V, E)$ terdiri dari V yang merupakan himpunan tidak kosong dari *vertex* (atau *node*) dan E yang merupakan himpunan *edge*, dimana setiap *edge* menghubungkan dua *vertex* yang berbeda dan dimana tidak ada dua *edge* menghubungkan sepasang *vertex* yang sama [3]. *Eulerian circuit* adalah sebuah *circuit* yang memuat setiap *vertex* dan *edge* pada sebuah graf [4]. Kasus perhitungan jumlah *Eulerian circuit* ini menarik untuk diangkat karena sampai sekarang belum ada algoritma ataupun teorema yang dapat memastikan jumlah *Eulerian circuit* pada sembarang graf sederhana, kecuali pada graf-graf khusus seperti graf lengkap K_n . Graf lengkap dengan n *vertex* yang dilambangkan dengan K_n merupakan graf sederhana yang setiap dua *vertex* yang berbeda dihubungkan oleh tepat satu *edge* [4]. Karena itu, timbullah sebuah gagasan untuk merancang sebuah algoritma pencarian *Eulerian circuit* yang mampu menampilkan daftar semua *Eulerian circuit* dan

menghitung jumlah *Eulerian circuit* pada sembarang graf sederhana yang disajikan dalam bentuk media pembelajaran elektronik. Permasalahan pencarian *Eulerian circuit* ini merupakan permasalahan dalam bidang teori graf yang kebanyakan menggunakan ilustrasi graf secara visual, maka untuk merancang media pembelajarannya harus memilih *software* yang mampu menampilkan konten secara visual dan interaktif, salah satunya yaitu *software* Adobe Flash. Adobe Flash adalah lingkungan *authoring* paling canggih yang dilengkapi dengan bahasa pemrograman *actionscript* untuk membuat dan menampilkan konten interaktif digital, *web* dan *mobile platform* [5]. Adobe Flash tidak hanya digunakan untuk membuat konten pada situs *web* yang canggih dan menarik, namun Adobe Flash juga sudah mulai digunakan dalam penyusunan presentasi, game, *e-learning courseware* dan lainnya yang dapat disebar via internet, DVD, CD, atau jaringan lokal [6]. Dilihat dari segi kemampuan visualnya yang bahkan telah digunakan dalam perancangan sebuah game, maka Adobe Flash merupakan *software* yang tepat untuk digunakan dalam perancangan sebuah media pembelajaran elektronik, salah satunya media pembelajaran pencarian *Eulerian circuit* ini, apalagi dengan dilengkapi bahasa pemrograman *actionscript*, sehingga dapat digunakan untuk membuat media pembelajaran yang lebih dinamik.

Namun, karena semua *Eulerian circuit* dicari dengan metode pencarian manual dan dalam satu graf bahkan bisa memuat jutaan *Eulerian circuit*, langkah perancangan media pembelajaran pencarian *Eulerian circuit* ini pun terhenti saat diketahui bahwa *software* yang dikembangkan oleh perusahaan Adobe ini tidak mendukung untuk komputasi dalam skala besar. Berbeda dengan *software* pemrograman pada umumnya seperti Matlab, Borland C++ atau Pascal. Meskipun Adobe Flash telah berhasil dalam perancangan game, Adobe Flash ternyata tidak digunakan dalam perancangan game yang menggunakan perhitungan rumit dan berat, namun hanya pada game-game ringan saja. Karena itu, dalam penelitian ini penulis mencoba membahas metode yang tepat untuk menyelesaikan permasalahan komputasi skala besar pada Adobe Flash, khususnya dalam perancangan media pembelajaran pencarian *Eulerian circuit*. Metode ini tidak hanya dapat digunakan pada kasus perancangan media pembelajaran ini saja, namun juga dapat menjadi solusi pada kasus-kasus rumit lainnya seperti pada perancangan game dengan komputasi rumit dan berat.

II. METODOLOGI

Permasalahan komputasi skala besar ini muncul saat dilakukan pengujian media pembelajaran

pencarian *Eulerian circuit* menggunakan *software* Adobe Flash, yaitu saat memeriksa jumlah *Eulerian circuit* dari graf lengkap dengan tujuh *vertex* (K_7). Sehingga untuk mengetahui pokok permasalahan dari masalah komputasi pada Adobe Flash ini, ada baiknya terlebih dahulu mengetahui sedikit teori tentang jumlah *Eulerian circuit* pada sebuah graf sederhana dan langkah demi langkah sampai kepada batasan dalam *software* Adobe Flash sehingga menghambat proses pengujian media pembelajaran ini. Perhitungan jumlah *Eulerian circuit* pada sembarang graf sederhana merupakan permasalahan yang rumit. Sehingga belum ada algoritma ataupun teorema yang dapat memastikan jumlah *Eulerian circuit* pada sembarang graf sederhana, kecuali pada graf-graf khusus seperti pada graf K_n . Pada graf lengkap berarah K_n untuk n ganjil, jumlah *Eulerian circuit* dapat dihitung dengan menggunakan rumus (1) di bawah ini. Graf lengkap K_n hanya memiliki *Eulerian circuit* pada graf lengkap dengan n ganjil, sedangkan untuk n genap tidak memiliki *Eulerian circuit* menurut Teorema Euler karena setiap *vertex*-nya memiliki derajat ganjil yaitu $(n-1)$ [4].

$$Eul(n) = \left(\frac{1}{2}n - \frac{3}{2}\right)!^n e_n \quad (1)$$

Rumus untuk menghitung jumlah *Eulerian circuit* pada graf lengkap berarah yang ditunjukkan pada rumus (1) di atas menggunakan *asymptotic enumeration methods* dengan menggunakan nilai e_n pada Tabel 1 berikut yang dijabarkan oleh McKay dalam penelitiannya [7].

Tabel 1. Nilai e_n untuk rumus (1)

| N | e_n |
|----|---|
| 3 | 2 |
| 5 | 264 |
| 7 | 1.015.440 |
| 9 | 90.449.251.200 |
| 11 | 169.107.043.478.365.440 |
| 13 | 6.267.416.821.165.079.203.599.360 |
| 15 | 4.435.711.276.305.905.572.695.127.676.467.200 |

Sumber: McKay (1995)

Walaupun rumus (1) di atas mampu menghitung jumlah *Eulerian circuit*, namun hanya mampu pada graf lengkap, sedangkan media pembelajaran pencarian *Eulerian circuit* ini mampu menghitung pada semua jenis graf sederhana serta menampilkan daftar semua *Eulerian circuit*-nya. Kemudian yang menjadi masalah lain dari rumus (1) di atas yaitu, rumus tersebut digunakan untuk menghitung graf lengkap berarah, sedangkan kasus pada penelitian ini pada graf lengkap tidak berarah. Namun, jumlah *Eulerian circuit* pada graf lengkap tidak berarah

juga dapat dihitung dengan rumus tersebut karena jumlah *Eulerian circuit* pada graf lengkap tidak berarah adalah setengah dari jumlah *Eulerian circuit* pada graf lengkap berarah, karena pada graf lengkap berarah dua *Eulerian circuit* yang memiliki arah berlawanan dianggap berbeda sedangkan pada graf lengkap yang tidak berarah dianggap sama asalkan urutan *edge*-nya tetap. Sehingga dengan menggunakan rumus (1) diperoleh hasil seperti yang ditunjukkan pada Tabel 2 berikut ini.

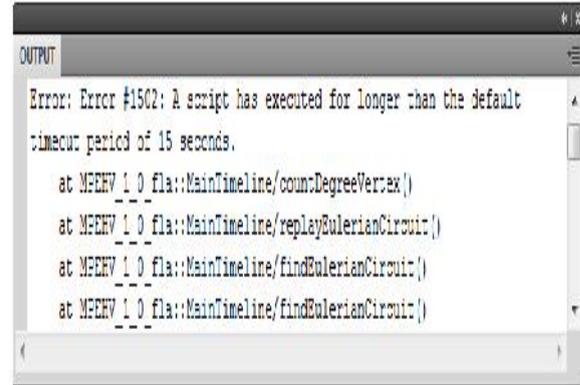
Tabel 2. Jumlah *Eulerian circuit* pada graf lengkap tidak berarah K_n

| Jumlah Vertex (n) | e_n | $Eul(n)/2$ |
|-------------------|---|----------------|
| 3 | 2 | 1 |
| 5 | 264 | 132 |
| 7 | 1.015.440 | 64.988.160 |
| 9 | 90.449.251.200 | 4,55760 x 1017 |
| 11 | 169.107.043.478.365.440 | 1,28663 x 1032 |
| 13 | 6.267.416.821.165.079.203.599.360 | 3,35286 x 1051 |
| 15 | 4.435.711.276.305.905.572.695.127.676.467.200 | 1,60665 x 1076 |

Dalam mencari dan menghitung jumlah *Eulerian circuit* pada graf sederhana, media pembelajaran ini menggunakan sebuah algoritma yang merupakan hasil modifikasi algoritma *Fleury*. Algoritma *Fleury* adalah salah satu algoritma sederhana yang digunakan untuk menelusuri *Eulerian circuit* pada sembarang graf sederhana yang ditemukan oleh Fleury pada tahun 1883 [4], algoritma ini hanya mampu menemukan satu *Eulerian circuit* pada graf sederhana. Setelah semua tahapan perancangan media pembelajaran pencarian *Eulerian circuit* ini selesai, mulai dari penyusunan algoritma, merancang *user interface*, pembuatan *storyboard* sampai pengkodean program dengan menggunakan bahasa pemrograman *actionscript*, maka tibalah pada tahap pengujian program. Pengujian program dilakukan untuk memastikan bahwa program pada media pembelajaran ini berjalan lancar dan mampu mendapatkan hasil yang sesuai dengan yang diharapkan dengan cara membandingkan antara hasil yang diperoleh program dengan hasil dari rumus.

Pengujian program dengan Adobe Flash untuk graf lengkap K_n dengan $n = 3$ dan $n = 5$ dapat menghasilkan jumlah *Eulerian circuit* yang benar yaitu berjumlah satu *Eulerian circuit* untuk $n = 3$ dan 132 *Eulerian circuit* untuk $n = 5$ dalam waktu kurang dari satu detik. Namun, saat pencarian dan perhitungan *Eulerian circuit* pada graf lengkap K_7 yang memiliki 64.988.160 *Eulerian circuit*, Adobe Flash menampilkan pesan kesalahan seperti pada Gambar 1 berikut. Pesan tersebut menandakan bahwa *runtime* program pencarian dan perhitungan

Eulerian circuit ini melebihi batas waktu yang ditentukan pada *software* Adobe Flash atau dinamakan dengan *script time limit* [8]. *Script time limit* ini bertujuan untuk menghindari masalah dalam pemrograman seperti *circular logic* dan *endless loops* [9].



Gambar 1 Pesan kesalahan pada Adobe Flash

Ada empat metode yang akan dicoba untuk menyelesaikan masalah *script time limit* ini yaitu mengubah nilai *script time limit* pada pengaturan, mengubah algoritma secara total, mempartisi proses dalam beberapa bagian, mempartisi proses dan menyisipkan objek *timer* ke dalam algoritma. Dengan metode tersebut diharapkan dapat menyelesaikan permasalahan *script time limit* ini sehingga pengujian program untuk graf lengkap K_n untuk n ganjil di atas lima dapat diselesaikan. Adapun spesifikasi komputer yang digunakan dalam pengujian media pembelajaran pencarian *Eulerian circuit* ini yaitu seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Spesifikasi komputer yang digunakan

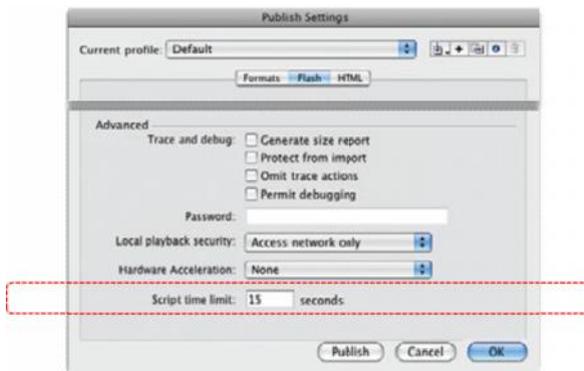
| | |
|-------------|---|
| Processor | Intel® Core(TM) i3-2310M CPU @ 2.10 GHz |
| RAM | DDR 4 GB |
| System Type | 32-bit Operating System |
| OS | Windows 7 Ultimate |

III. HASIL DAN PEMBAHASAN

Pengujian program jumlah *Eulerian circuit* menggunakan *software* Adobe Flash dapat diselesaikan dengan terlebih dahulu menyelesaikan permasalahan *script time limit* dan telah dicoba menggunakan empat metode penyelesaian yaitu mengubah nilai *script time limit* pada pengaturan, mengubah algoritma secara total, mempartisi proses dalam beberapa bagian, mempartisi proses dan menyisipkan objek *timer* ke dalam algoritma. Keempat metode tersebut dan hasil yang diperoleh akan dibahas pada sub pembahasan berikut ini.

Mengubah Nilai *Script Time Limit* pada Pengaturan Adobe Flash

Salah satu metode yang mungkin dilakukan untuk mengatasi masalah *script time limit* pada Adobe Flash ini yaitu mengubah nilai *default* dari *script time limit* dengan nilai yang diinginkan. Nilai *default* dari *script time limit* pada *actionscript* adalah 15 detik. Nilai *script time limit* tersebut dapat diubah dari kotak dialog *Publish Settings*. Kotak dialog ini dapat dibuka pada menu *File*, pada tab *Flash* dan pada bagian *Advanced* seperti yang terlihat pada Gambar 2. Nilai *script time limit* dapat langsung diubah dengan mengganti nilai 15 pada *textbox script time limit* dengan nilai yang diinginkan. Namun, nilai yang dimasukkan tidak boleh melebihi nilai maksimum dari *script time limit* yang telah diatur oleh Adobe yaitu 60 detik [10]. Selanjutnya setelah nilai *script time limit* diatur maksimum yaitu 60 detik dan dicoba pada program pencarian dan perhitungan *Eulerian circuit* tersebut, program sudah mampu berjalan dan bertahan dalam waktu lama, namun tiba-tiba pesan kesalahan yang sama tetap muncul pada layar yang diperkirakan waktunya sesuai dengan nilai yang dimasukkan yaitu 60 detik. Dengan kata lain, waktu yang dibutuhkan saat proses pencarian dan perhitungan *Eulerian circuit* ini melebihi waktu maksimal yang diizinkan pada *software* Adobe Flash tersebut, sehingga metode ini bukan salah satu solusi yang tepat untuk kasus komputasi skala besar pada media pembelajaran pencarian *Eulerian circuit*.



Gambar 2. Tampilan kotak dialog Publish Settings

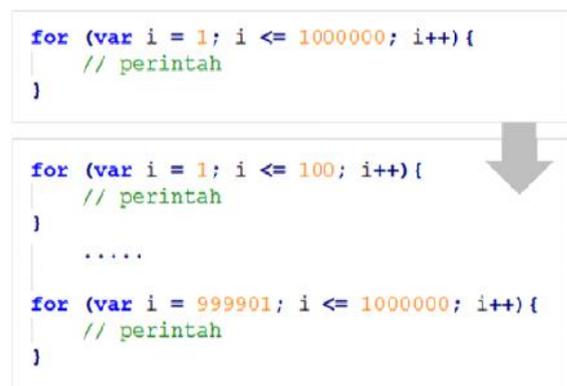
Mengubah Algoritma secara Total

Kemungkinan solusi yang kedua yaitu mengubah total algoritma yang telah dirancang. Ini bisa saja menjadi salah satu solusi, namun bukan hal yang mudah, karena selain algoritma telah dipasang dan diterapkan pada media pembelajarannya, merancang algoritma kasus pencarian semua *Eulerian circuit* merupakan kasus yang sulit, jika mengubah total algoritma, maka tidak jauh beda dengan merancang ulang algoritma baru. Kemudian

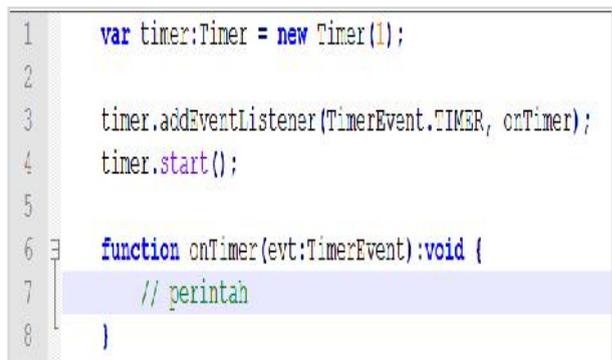
walaupun telah diubah, namun belum tentu algoritma baru tersebut mampu bekerja untuk mendapatkan hasil dalam waktu 60 detik karena kompleksitas algoritma ini adalah eksponensial, misalkan untuk graf lengkap K_{15} yang memiliki $1,60665 \times 10^{76}$ *Eulerian circuit*, kemungkinan akan membutuhkan waktu di atas 60 detik, apalagi untuk K_n dengan n di atas 15. Sehingga metode ini juga bukan salah satu solusi yang efisien untuk kasus komputasi skala besar pada media pembelajaran pencarian *Eulerian circuit*.

Mempartisi Proses Pencarian dan Perhitungan *Eulerian Circuit*

Metode yang ketiga yaitu mempartisi proses pencarian dan perhitungan *Eulerian circuit*. Mempartisi proses berarti membagi atau memecahkan sebuah program menjadi beberapa bagian program lainnya yang jika dijalankan tetap menghasilkan hasil yang sama dengan program asalnya. Gambaran sederhana dalam mempartisi sebuah proses dapat dilihat pada Gambar 3. Setelah cara ini diterapkan pada program pencarian dan perhitungan *Eulerian circuit* pada graf lengkap K_7 , ternyata Adobe Flash juga masih menampilkan pesan kesalahan yang sama karena tetap dianggap dalam satu proses yang sama. Sehingga metode ini juga bukan salah satu solusi yang tepat untuk kasus komputasi skala besar pada media pembelajaran pencarian *Eulerian circuit* ini.



Gambar 3. Mempartisi proses



Gambar 4. Mendeklarasi objek *timer*

Merpartisi Proses dan Menyisipkan Objek Timer

Solusi terakhir untuk menyelesaikan permasalahan komputasi skala besar yang disebabkan *script time limit* pada Adobe Flash ini yaitu mempartisi proses serta menyisipkan objek *timer* dalam setiap prosesnya. Untuk membuat sebuah objek *timer* dapat mengikuti *code* pada Gambar 4 berikut [11]. Dalam Gambar 4, baris pertama yaitu “var timer: Timer = new Timer(1)” artinya mendefinisikan sebuah objek *timer* dengan *delay* satu *millisecond* dan dengan nama “timer”.

Kemudian baris ketiga yaitu:

“timer.addEventListener(TimerEvent.TIMER, onTimer)” artinya membuat sebuah *timer event* untuk objek *timer* dengan nama “onTimer”. Sedangkan “timer.start()” artinya memulai perhitungan untuk objek *timer*. Baris enam sampai delapan merupakan deklarasi fungsi *timer event* dari fungsi “onTimer” yang dapat diisi dengan sebuah perintah. *Code* pada Gambar 4 dapat langsung digunakan untuk menyelesaikan permasalahan *script time limit* dengan mengubah menjadi seperti yang terlihat pada Gambar 5. *Code* pada Gambar 5 merupakan penggunaan objek *timer* pada algoritma pencarian *Eulerian circuit*.

```
1 var timer:Timer = new Timer(1);
2 var temp:int, jumlahEcir:int, maxCounter:int;
3
4 timer.addEventListener(TimerEvent.TIMER, onTimer);
5 timer.start();
6
7 E function onTimer(evt:TimerEvent):void {
8     maxCounter = maxCounter + 50;
9     temp = jumlahEcir;
10    findEulerianCircuit(); // Untuk setiap 50 Eulerian circuit
11 E    if(jumlahEcir == temp){
12        timer.sleep();
13    }
14 }
```

Gambar 5. Menyisipkan objek *timer*

Pada baris kedua, variabel bertipe integer “temp” berfungsi untuk menampung jumlah *Eulerian circuit* yang telah dihitung dari variabel “jumlahEcir” yang ditunjukkan pada baris delapan di Gambar 5. Sedangkan variabel “jumlahEcir” yang juga bertipe integer berfungsi untuk merekam jumlah *Eulerian circuit* yang ditemukan. Selanjutnya variabel “maxCounter” berfungsi untuk menampung maksimal jumlah *Eulerian circuit* yang boleh didapatkan dalam satu kali proses pencarian. Baris kelima berfungsi memanggil fungsi “onTimer” dalam beberapa kali pemanggilan, pada setiap kali pemanggilan fungsi, proses akan berhenti satu *millisecond* atau dinamakan *delay* dan akan berhenti total ketikan perintah “timer.stop()” dijalankan. Baris ke-10 yaitu pemanggilan fungsi “findEulerianCircuit()” yang merupakan sebuah fungsi untuk mencari maksimal 50 *Eulerian circuit* pada graf beserta

menghitung jumlahnya setiap kali pemanggilan, dengan kata lain, proses pencarian dan perhitungan jumlah *Eulerian circuit* dipartisi dengan setiap bagian partisi hanya mencari 50 *Eulerian circuit*. Nilai maksimal *Eulerian circuit* dalam satu kali proses pemanggilan fungsi ini harus disesuaikan dengan lama proses pencariannya, sebaiknya satu kali proses pencarian harus selesai dalam waktu maksimal satu detik untuk meringankan kinerja komputer. Sehingga karena untuk mencari 50 *Eulerian circuit* membutuhkan waktu mendekati satu detik, maka nilai maksimal *Eulerian circuit* tidak berlaku untuk semua jenis graf, untuk pencarian dan perhitungan *Eulerian circuit* pada graf lengkap K_n dengan n ganjil di atas 15, mungkin tidak lagi dapat diselesaikan dalam waktu satu detik, sehingga nilai 50 harus diganti dengan nilai lain dibawah 50 pada *code* di baris ke-8. Baris ke-11 sampai baris ke-13 berfungsi untuk menghentikan proses jika jumlah *Eulerian circuit* pada pencarian berikutnya yang disimpan pada variabel “jumlahEcir” sama dengan jumlah *Eulerian circuit* pada pencarian sebelumnya yang disimpan pada variabel “temp”, dengan kata lain tidak ada lagi *Eulerian circuit* yang ditemukan.

Penggunaan objek *timer* berhasil untuk memecahkan masalah *script time limit* pada Adobe Flash. Dengan menggunakan objek *timer* ini, Adobe Flash sudah mampu untuk menghitung jumlah *Eulerian circuit* dalam komputasi berat yang membutuhkan waktu proses melebihi waktu 60 detik pada Adobe Flash. Sehingga jumlah *Eulerian circuit* yang diperoleh untuk graf lengkap K_7 yaitu 64.988.160 *Eulerian circuit* dalam waktu sekitar 50 jam dan sesuai dengan hasil yang diperoleh dari rumus (1). Sehingga dapat disimpulkan metode ini merupakan salah satu solusi yang tepat untuk kasus komputasi skala besar pada media pembelajaran pencarian *Eulerian circuit* ini. Namun, adapun kekurangan dari metode ini yaitu, waktu proses bertambah sebanyak 0,1 % dari waktu yang sebenarnya jika tanpa menggunakan *delay* satu *millisecond*.

KESIMPULAN

Adapun hasil yang menjadi kesimpulan dari penelitian ini yaitu, metode mempartisi proses dan menyisipkan objek *timer* merupakan salah satu solusi yang tepat untuk kasus komputasi skala besar menggunakan *software* Adobe Flash pada media pembelajaran pencarian *Eulerian circuit* pada graf sederhana. Kelebihan dari metode ini yaitu mampu melakukan proses perhitungan yang melebihi nilai maksimum dari *script time limit* yang telah diatur oleh Adobe yaitu 60 detik, sehingga proses perhitungan dapat diselesaikan walaupun waktu proses bertambah 0,1 % dari

waktu yang sebenarnya jika tanpa menggunakan objek *timer* tersebut.

REFERENSI

1. S.M. Rahman, *Multimedia Technologies: Concepts, Methodologies, Tools, and Applications*. IGI Global, United State of America, 2008.
2. B. Holmes and J. Gardner, *E-Learning: Concept and Practice*. SAGE Publications, California, 2006.
3. K.H. Rosen, *Discrete Mathematics and Its Applications* 7th ed. The McGraw-Hill Companies, Inc., New York, 2012.
4. E.G. Goodaire and M.M. Parmenter, *Discrete Mathematics with Graph Theory*, Prentice-Hall, United State of America, 2002.
5. P. Milbourne, *The Essential Guide to Flash CS4 with ActionScript*. Spinger –Verlag New York, Inc., United State of America, 2009.
6. F. Gerantabee and Aquent Creative Team, *Flash Professional CS5 Digital Classroom*. Wiley Publishing, Inc., Indiana, 2009.
7. B. McKay and R.W. Robinson, "Asymptotic Enumeration of Eulerian Circuits in The Complete Graph", *Combinatorica* 10 (4) (1995), 367–377.
8. R. Reinhardt and S. Dowd, *Adobe® Flash® CS4 Professional Bible*, Wiley Publishing, Inc., Indiana, 2009.
9. R. Shupe, *Learning Flash CS4 Professional*, O'Reilly Media, Inc., Canada, 2009.
10. Adobe, *Using Adobe® Flex® 4.6*, Adobe System, Inc., United State of America, 2015.
11. R. Shupe and Z. Rosser, *Learning ActionScript 3.0 A Beginner's Guide*. O'Reilly Media, Inc., Canada, 2008.