

PERBANDINGAN KINERJA BROWSER PADA PEMANFAATAN ANIMASI SVG (SCALABLE VECTOR GRAPHIC)

Ichwan Kurniawan¹⁾, Much. Rifqi Maulana²⁾, Christian Yulianto Rusli³⁾
STMIK Widya Pratama¹²³⁾
ichwan.ana10@gmail.com¹⁾, rifqi@stmik-wp.ac.id²⁾, cyr.tata@gmail.com³⁾

Abstrak

Penggunaan Motion Graphic (MG) tidak hanya digunakan sebagai sarana media seni, animasi menjadi bagian yang terpenting dalam pembuatan MG, Salah satu metode untuk menghemat waktu rendering adalah dengan mengurangi kompleksitas gerakan terdapat metode lain untuk menghemat waktu render, yaitu menggunakan pemrograman komputer. Salah satu pemrograman komputer yang dapat digunakan untuk membuat animasi adalah dengan memanfaatkan Scalable Vector Graphic (SVG), SVG menggunakan bahasa markup yang berjalan pada browser. Tujuan dari penelitian ini adalah apakah terdapat perbedaan kinerja browser pada pemanfaatan animasi SVG dengan perbedaan atribut durasi dalam perbedaan gerakan. Hasil dari Perbandingan Kinerja Browser pada Animasi SVG adalah, bahwa tidak terdapat perbandingan kinerja browser yang menonjol, animasi SVG dapat ditampilkan dan dijalankan dalam lingkungan browser dengan normal. Dapat dilihat pada kinerja rendering dan painting. Keseluruhan durasi animasi rata-rata lama waktu pada kinerja render sebanyak 245ms atau 4,78% dan rata-rata lama waktu pada kinerja painting sebanyak 161ms atau 3.14%. Kemudian Jika dilihat dari lama waktu GPU (Graphics Processing Unit) tidak bekerja dalam browser, tiap durasi memiliki lama waktu yang berbeda-beda. Waktu tercepat terdapat pada durasi 4.55s dengan waktu 4849ms dan waktu terlama terdapat pada durasi 0.05s dan 1.55s dengan waktu 4921ms. Artinya bahwa durasi animasi SVG tidak mempengaruhi kinerja browser dalam menjalankan animasi SVG. Hal ini dikarenakan bahwa dokumen SVG yang dijalankan melalui browser dijalankan dalam perintah (values animation) yang sama, jadi browser hanya menjalankan perintah yang sama sera berulang-ulang. Jika terjadi selisih kinerja browser kemungkinan dikarenakan karena terdapat pekerjaan yang lain yang dilakukan oleh perangkat pemroses dalam satu kegiatan.

Kata kunci: Penbadingan, SVG, Animation, Kinerja Browser

1. Pendahuluan

Motion Graphic (MG) digunakan sebagai sarana ekspresi bagi kreator seni (Babic, Pibernik dan Mrvac 2008), penggunaan MG tidak hanya digunakan sebagai sarana media seni, namun perkembangannya MG dapat dimanfaatkan di berbagai bidang. Baik bidang pendidikan, periklanan, hukum dan sebagainya, MG dapat dimanfaatkan sebagai media pembelajaran yang lebih menarik (Sari, Rante dan Arini 2021). Pemanfaatan MG dalam proses pembelajaran dapat digunakan sebagai media pembelajaran, melalui MG diharapkan pengguna dapat memahami materi pembelajaran. Secara visual MG diharapkan dapat menghasilkan gerakan yang lebih objektif dan transisi gambar yang hasil.

Salah satu bagian terpenting dalam pembuatan MG adalah animasi (Xiao 2020), secara konsep MG memanfaatkan animasi sebagai pergerakan gambar yang saling berurutan. Pembuatan animasi, 2 dimensi maupun 3 dimensi diciptakan berdasarkan rentang waktu animasi dalam satu detik, hal ini disebut dengan *frame rate/frame per second(FPS)* (Kurniawan, Maulana dan Siregar 2020). Terdapat beberapa cara dalam pembuatan animasi salah satunya menggunakan *key time line*, namun dengan menggunakan cara ini membutuhkan waktu yang relatif cukup lama. Hal ini dikarenakan dalam pembuatan animasi, membutuhkan banyak gambar yang dimasukkan kedalam *key frame*. Selain itu dengan cara ini membutuhkan sumber daya komputasi yang cukup besar.

Salah satu metode untuk menghemat waktu *rendering* dengan mengurangi detail gerakan, hal ini akan sedikit memengaruhi hasil akhir animasi (V. Patil and L. Deshpande 2016). Selain mengurangi kompleksitas gerakan terdapat metode lain untuk menghemat waktu render, yaitu menggunakan pemrograman komputer. Perbandingan dua metode *frame by frame* dan *expression* adalah untuk *rendering process*, metode *expression (script)* lebih unggul 13 menit lebih cepat dibandingkan dengan metode *frame by frame* pada komputer dengan spesifikasi menengah, sedangkan untuk spesifikasi tinggi sama-sama 17 menit (Pramono, Suyanto and Sofyan 2017). Dalam penelitian tersebut belum diuji pada animasi *Motion Graphic (MG)*, apakah dengan menggunakan pemrograman komputer proses animasi dapat dijalankan melalui script program di setiap *frame* (Kramer, et al. 2016). Hal ini dapat memungkinkan proses pembuatan *MG* dapat diselesaikan dengan lebih cepat, karena gerakan dijalankan melalui intruksi program komputer. Selain itu dengan memanfaatkan script program memungkinkan meminimalkan kerja komputer dalam proses rendering.

Scalabel Vector Graphic (SVG) dapat digunakan untuk menghidupkan proses pembuatan diagram *MG* yang ditampilkan pada browser web (Goto, et al. 2010). Animasi dapat diciptakan dengan memanfaatkan script pemrograman *markup language SVG*, hal ini bertujuan agar animasi dapat tercipta dengan presisi nilai yang sesuai dengan kenyataan.

Penelitian yang terkait dengan Pemanfaatan *Scalabel Vector Graphic (SVG)* antarlain adalah:

Liao dan kawan-kawan dalam papernya yang berjudul *SVG engine design and optimization*, menyatakan bahwa penggunaan *SVG* sebagai bahasa *XML* dapat digunakan sebagai bagian utama dalam membuat karya visual grafik yang dinamis (Liao, et al. 2010). Dalam hal ini pembuatan karya visual dinamis dalam hal ini animasi dapat menggunakan pemrograman komputer.

Goto dan kawan-kawan dalam paper yang berjudul *Automatic Generation of SVG Program Documents with Animation Based on Attribute Graph Grammars*, menyatakan bahwa dengan memanfaatkan *SVG* dapat digunakan untuk

membuat diagram secara otomatis. File *SVG* dapat menghidupkan proses perhitungan kompleksitas siklometrik dari diagram (Goto, et al. 2010). Penelitian ini penggunaan pemrograman *SVG* dapat digunakan untuk membuat diagram yang interaktif.

Pemaparan di atas bahwa dalam pembuatan animasi *MG*, selain menggunakan teknik *keyframe*, animasi dapat dibuat dengan memanfaatkan script program. Oleh karena itu dengan memanfaatkan *SVG* dalam pembuatan animasi *MG* dapat meminimalkan waktu *render* tanpa mengurangi kualitas *render*, dan dapat membuat animasi dengan presisi nilai yang sesuai dengan kenyataannya.

Berdasarkan latar belakang dan riset terdahulu maka dapat diambil rumusan masalahnya adalah “Apakah terdapat perbedaan kinerja *browser* pada pemanfaatan animasi *SVG* dengan perbedaan atribut durasi dalam perbedaan gerakan?”. Sedangkan tujuan dari penelitian ini adalah membandingkan kinerja *browser* pada pemanfaatan animasi *SVG* dengan perbedaan atribut durasi dan perbedaan gerakan. Kemudian manfaat dari penelitian ini adalah memberikan gambaran bagi pengembang *frontend* mengenai kinerja *browser* pada pemanfaatan animasi *SVG*.

2. Metode Penelitian

Jenis penelitian ini adalah penelitian experimental dengan metode penelitian sebagai berikut:

- a. Pengumpulan Data
Penentuan masalah penelitian adalah dengan menggunakan studi literatur dan studi lapangan.
- b. Penentuan Computing Approach
Computing approach pada penelitian ini adalah penggunaan metode pembuatan animasi dengan memanfaatkan bahasa pemrograman komputer, *Scalabel Vector Graphic (SVG)* dipilih dengan pemikiran bahwa, *SVG* dapat digunakan sebagai alternatif pembuatan *Interactive Motion Graphic Browser Platform*.
- c. Implementasi Kinerja *browser* pada animasi *SVG* dengan menggunakan *performance DevTools Chrome*.
- d. Eksperimen dengan melakukan perbandingan kinerja *browser* dengan menggunakan

DecTools Chrome pada pemanfaatan animasi *SVG* dengan perbedaan atribut durasi dan perbedaan gerakan.

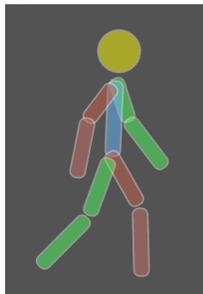
- e. Evaluasi dengan melakukan komparasi dengan data empiris. Pengujian Pemanfaatan *SVG* sebagai alternatif pembuatan animasi pada lingkungan *browser*, dilakukan melalui Perbandingan kinerja *browser* pada pemanfaatan animasi *SVG* dengan perbedaan atribut durasi dan perbedaan gerakan.

3. Hasil dan Pembahasan

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah berhubungan dengan masalah yang diteliti, data tersebut adalah data *Script* program animasi *SVG*. Data ini nantinya akan dilakukan eperimen pengrusnya terhadap kinerja *browser*. Data *Script* program animasi *SVG*, dibuat degan menggunakan *Text Editor Sublime*, dengan gambaran human character. Yaitu *character* yang mempunyai sifat seperti manusia, terdapat kepala, badan tangan dan kaki.

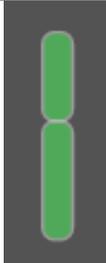
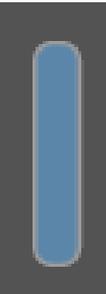
Script Character



Gambar 1. *Character Animation*

Tabel 1. *Markup Character SVG Script*

Visual	Script
 Kepala	<pre><g id="kepala"> <circle cx="520" cy="400" r="70" style="fill:#fffc00;stroke: #e6e6e6;stroke- width:5;opacity:0.5"/> </g></pre>

Visual	Script
 Tangan Kanan/Kiri	<pre><g id="A_Tangan_ki"> <rect x="500" y="500" rx="20" ry="20" width="50" height="150" style="fill:#cc6248;stroke: #e6e6e6;stroke- width:5;opacity:0.5"/> <rect id="B_Tangan_ki" x="500" y="650" rx="20" ry="20" width="50" height="200" style="fill:#cc7263;stroke: #e6e6e6;stroke- width:5;opacity:0.5"/> </g></pre>
 Kaki Kanan/Kiri	<pre><g id="A_Kaki_ki"> <rect x="500" y="750" rx="20" ry="20" width="50" height="200" style="fill:#cc6248;stroke: #e6e6e6;stroke- width:5;opacity:0.5"/> <rect id="B_Kaki_ki" x="500" y="950" rx="20" ry="20" width="50" height="225" style="fill:#cc7263;stroke: #e6e6e6;stroke- width:5;opacity:0.5"/> </g></pre>
 Badan	<pre><g id="badan"> <rect id="badan" x="500" y="500" rx="20" ry="20" width="50" height="250" style="fill:#66b9ff;stroke: #e6e6e6;stroke- width:5;opacity:0.5"/> </g></pre>

Script Animation

```
<animateTransform attributeName="transform"
id="anim_N"
xlink:href="#N"
```

```
attributeType="XML"
type="rotate"
values="Rot, x, y"
repeatCount="indefinite"
dur="1.5s"
fill="freeze"/>
```

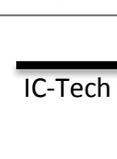
Script 1. Script Animation

3.2 Penentuan Computing Approach

Computing approach pada penelitian ini adalah penggunaan metode pembuatan animasi dengan memanfaatkan bahasa pemrograman komputer, Scalabel Vector Graphic (SVG) dipilih dengan pemikiran bahwa, SVG dapat digunakan sebagai alternatif pembuatan Interactive Motion Graphic Browser Platform dengan melakukan perbandingan kinerja browser.

Script Animation

Tabel 2. Markup Animation SVG Script

Visual	Values
	B_Tangan_Ki : -10, 500, 650; B_Tangan_Ka : -40, 500, 650; A_Tangan_Ki : -20, 500, 500; A_Tangan_Ka : 30, 500, 500; B_Kaki_Ki : 20, 500, 950; B_Kaki_Ka : -5, 500, 950; A_Kaki_Ki : 20, 500, 750; A_Kaki_Ka : -20, 500, 750;
	B_Tangan_Ki : -20, 500, 650; B_Tangan_Ka : -25, 500, 650; A_Tangan_Ki : -20, 500, 500; A_Tangan_Ka : 35, 500, 500; B_Kaki_Ki : 25, 500, 950; B_Kaki_Ka : 30, 500, 950; A_Kaki_Ki : 20, 500, 750; A_Kaki_Ka : -30, 500, 750;
	B_Tangan_Ki : -25, 500, 650; B_Tangan_Ka : -20, 500, 650; A_Tangan_Ki : 0, 500, 500; A_Tangan_Ka : 20, 500, 500; B_Kaki_Ki : 30, 500, 950; B_Kaki_Ka : 30, 500, 950; A_Kaki_Ki : -20, 500, 750; A_Kaki_Ka : -10, 500, 750;
	B_Tangan_Ki : -15, 500, 650;

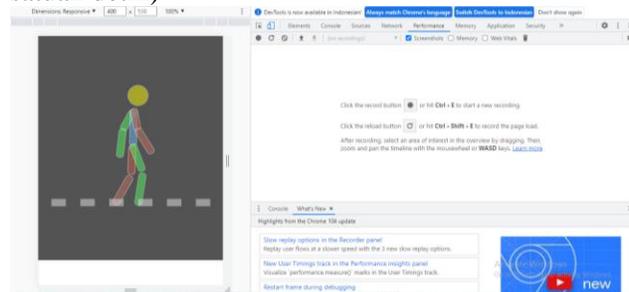
Visual	Values
	B_Tangan_Ka : -15, 500, 650; A_Tangan_Ki : 10, 500, 500; A_Tangan_Ka : 10, 500, 500; B_Kaki_Ki : 20, 500, 950; B_Kaki_Ka : 5, 500, 950; A_Kaki_Ki : -25, 500, 750; A_Kaki_Ka : 20, 500, 750;
	B_Tangan_Ki : -40, 500, 650; B_Tangan_Ka : -10, 500, 650; A_Tangan_Ki : 30, 500, 500; A_Tangan_Ka : -20, 500, 500; B_Kaki_Ki : -5, 500, 950; B_Kaki_Ka : 20, 500, 950; A_Kaki_Ki : -20, 500, 750; A_Kaki_Ka : 20, 500, 750;
	B_Tangan_Ki : -25, 500, 650; B_Tangan_Ka : -20, 500, 650; A_Tangan_Ki : 35, 500, 500; A_Tangan_Ka : -20, 500, 500; B_Kaki_Ki : 30, 500, 950; B_Kaki_Ka : 25, 500, 950; A_Kaki_Ki : -30, 500, 750; A_Kaki_Ka : 20, 500, 750;
	B_Tangan_Ki : -20, 500, 650; B_Tangan_Ka : -25, 500, 650; A_Tangan_Ki : 20, 500, 500; A_Tangan_Ka : 0, 500, 500; B_Kaki_Ki : 30, 500, 950; B_Kaki_Ka : 30, 500, 950; A_Kaki_Ki : -10, 500, 750; A_Kaki_Ka : -20, 500, 750;
	B_Tangan_Ki : -15, 500, 650; B_Tangan_Ka : -15, 500, 650; A_Tangan_Ki : 10, 500, 500; A_Tangan_Ka : 10, 500, 500; B_Kaki_Ki : 5, 500, 950; B_Kaki_Ka : 20, 500, 950; A_Kaki_Ki : 20, 500, 750; A_Kaki_Ka : -25, 500, 750;
	B_Tangan_Ki : -10, 500, 650 B_Tangan_Ka : -40, 500, 650 A_Tangan_Ki : -20, 500, 500 A_Tangan_Ka : 30, 500, 500 B_Kaki_Ki : 20, 500, 950

Visual	Values
B_Kaki_Ka	: -5, 500, 950
A_Kaki_Ki	: 20, 500, 750
A_Kaki_Ka	: -20, 500, 750

3.3 Implementasi Kinerja Animasi SVG

Implementasi perbandingan kinerja *browser* pada pemanfaatan animasi SVG dilakukan menggunakan *DevTools Chrome* (Google. inc 2022), pada penelitian ini akan dilihat perbandingan kinerja *browser* dengan menambah *attribute dur* (*duration*) pada tag *animateTransform*.

Perbandingan kinerja *browser* yang akan diukur adalah kecepatan *browser* dalam merespon *script*, *attribute dur* SVG yang digunakan adalah 0.05, 0.55, 1.05, 1.552, 2.05, 2.55, 3.05, 3.55, 4.05 dan 4.55 (dalam satuan detik)



Gambar 2. Tampilan *DevTools Chrome*

3.4 Pembahasan

Kinerja *browser* dapat diukur melalui cepat atau lambanya *browser* dalam menampilkan halaman web, dalam *DevTools Chrome* kinerja *browser* dapat dilihat dari kecepatan dalam menampilkan dalam rentang waktu rata-rata 0 sampai 5 detik .

Kinerja *browser* tersebut adalah kinerja dalam *loading* (lama memuat)/A , *Scripting* (lama membuat script)/B, *Rendering* (lama render)/C, *Painting* (lama painting)/D, *System* (Sistem) E dan *Idle* (lama tidak ada aktifitas)/F dalam satuan *milisecond(ms)*, Selain itu kinerja *browser* dapat dilihat dalam lama penggunaan *Graphic Processing Unit (GPU)*.

1. Kinerja Browser

Tabel 3. Kinerja *Browser*

Dur(s)	A	B	C	D	E	F	Total
0.05	3	56	258	189	459	4221	5186
0.55	3	28	215	134	296	4419	5094
1.05	2	14	202	134	326	4433	5111
1.55	2	36	282	171	405	4281	5177
2.05	2	29	260	179	358	4283	5111
2.55	2	28	225	157	302	4388	5102
3.05	2	29	280	153	351	4312	5127
3.55	2	29	293	205	366	4246	5141
4.05	2	29	209	132	281	4452	5105
4.55	3	31	229	155	322	4369	5109

Hasil dari kinerja *browser* dapat dilihat bahwa, perbedaan durasi animasi menentukan kinerja *browser* dalam menjalankan file pada lingkungan pada *browser* pada rentang waktu 0 sampai 5 detik, terdapat perbedaan.

Kinerja *browser* dalam memuat/load file, dapat dilihat dari durasi yang diuji hanya terdapat perbedaan pada durasi animasi 4.55s yaitu selama 3ms. Selain durasi 4.55s hanya membutuhkan waktu memuat/load hanya 2ms. Dapat dilihat bahwa waktu terlama adalah pada durasi 0.05s dan 4.55s.

Tabel 4. Kinerja *Browser Load*

Dur	0,05	0,55	1,05	1,55	2,05	2,55	3,05	3,55	4,05	4,55
A	3	2	2	2	2	2	2	2	2	3

Jika dilihat dari kinerja *browser* dalam menjalankan *script*, terdapat 4 durasi memiliki waktu yang sama yaitu durasi 2.05s, 3.05s, 3.55s dan 4.05s dengan waktu 29ms. Sedangkan yang lain durasi 0.05s dengan waktu 56ms, 0.55s dan 2.55s dengan waktu 28ms, durasi 1.05s dengan waktu 14ms, durasi 1.55s dengan waktu 36ms dan durasi 4.55s dengan waktu 31ms. Dapat dilihat bahwa waktu terlama pada pengukuran kinerja *browser* terdapat pada durasi 1.55s dengan waktu 36ms dan waktu tercepat pada durasi 1.05s dengan waktu 14ms.

Tabel 5. Kinerja *Browser Scripting*

Dur	0,05	0,55	1,05	1,55	2,05	2,55	3,05	3,55	4,05	4,55
B	56	28	14	36	29	28	29	29	29	31

Jika dilihat dari kinerja *browser* dalam menjalankan *Rendering*, tiap durasi memiliki lama waktu yang berbeda-beda. Waktu tercepat terdapat pada durasi 1.05s dengan waktu 202ms dan waktu terlama terdapat pada durasi 3.55s dengan waktu 293ms.

Tabel 6. Kinerja *Browser Rendering*

Dur	0,05	0,55	1,05	1,55	2,05	2,55	3,05	3,55	4,05	4,55
C	258	215	202	282	260	225	280	293	209	229

Jika dilihat dari kinerja *browser* dalam menjalankan *Painting*, terdapat waktu yang sama pada durasi 0.55s dan 1,05s dengan waktu 134ms. Kemudian durasi waktu yang lain memiliki lama waktu yang berbeda-beda. Waktu tercepat terdapat pada durasi 4.05s dengan waktu 132ms dan waktu terlama terdapat pada durasi 3.55s dengan waktu 205ms.

Tabel 7. Kinerja *Browser Painting*

Dur	0,05	0,55	1,05	1,55	2,05	2,55	3,05	3,55	4,05	4,55
D	189	134	134	171	179	157	153	205	132	155

Jika dilihat dari kinerja *browser* dalam penggunaan *System*, tiap durasi memiliki lama waktu yang berbeda-beda. Waktu tercepat terdapat pada durasi 4.05s dengan waktu 281ms dan waktu terlama terdapat pada durasi 0.05s dengan waktu 459ms.

Tabel 8. Kinerja *Browser System*

Dur	0,05	0,55	1,05	1,55	2,05	2,55	3,05	3,55	4,05	4,55
E	459	296	326	405	358	302	351	366	281	322

Jika dilihat dari kinerja *browser* dalam penggunaan *Idle*, tiap durasi memiliki lama waktu yang berbeda-beda. Waktu tercepat terdapat pada durasi 0.05s dengan waktu 4221ms dan waktu terlama terdapat pada durasi 4.05s dengan waktu 4452ms.

Tabel 9. Kinerja *Browser Idle*

Dur	0,05	0,55	1,05	1,55	2,05	2,55	3,05	3,55	4,05	4,55
F	4221	4419	4433	4281	4283	4388	4312	4246	4452	4452

2. Kinerja GPU dalam Browser

Tabel 10. Kinerja *GPU* dalam *Browser*

Dur(s)	GPU	Idle	Total
0.05	265	4921	5186
0.55	260	4833	5093
1.05	217	4894	5111
1.55	257	4921	5178
2.05	248	4863	5111
2.55	243	4858	5101
3.05	264	4861	5125
3.55	270	4870	5140
4.05	250	4855	5105
4.55	261	4849	5110

Jika dilihat dari lama waktu *GPU* dalam *browser*, tiap durasi memiliki lama waktu yang berbeda-beda. Waktu tercepat terdapat pada

durasi 1.05s dengan waktu 217ms dan waktu terlama terdapat pada durasi 3.55s dengan waktu 270ms.

Tabel 11. Kinerja *Browser GPU*

Dur	0,05	0,55	1,05	1,55	2,05	2,55	3,05	3,55	4,05	4,55
GPU	265	260	217	257	248	243	264	270	250	261

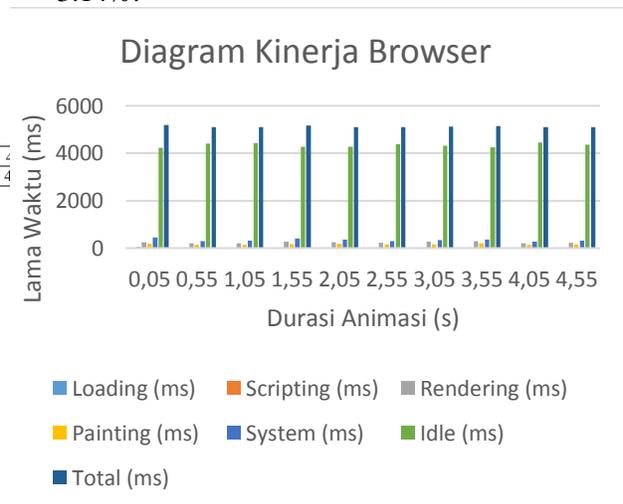
Jika dilihat dari lama waktu *GPU* tidak bekerja dalam *browser*, tiap durasi memiliki lama waktu yang berbeda-beda. Waktu tercepat terdapat pada durasi 4.55s dengan waktu 4849ms dan waktu terlama terdapat pada durasi 0.05s dan 1.55s dengan waktu 4921ms.

Tabel 12. Kinerja *Browser GPU*

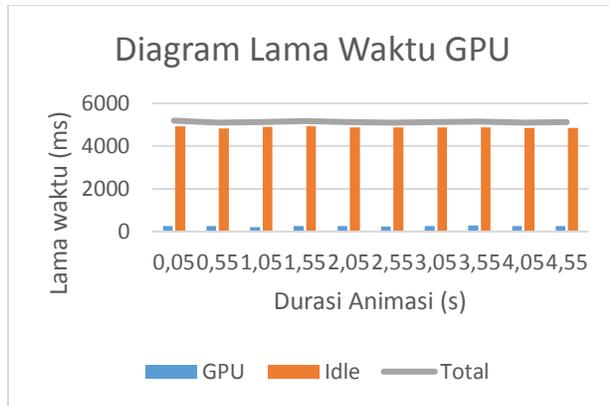
Dur	0,05	0,55	1,05	1,55	2,05	2,55	3,05	3,55	4,05	4,55
GPU	4921	4833	4894	4921	4863	4858	4861	4870	4855	4849

3.5 Evaluasi

Kinerja *browser* dalam menjalankan animasi *SVG* dalam menampilkan gambar dan menjalankan animasi, dapat dilihat pada kinerja *rendering* dan *painting*. Keseluruhan durasi animasi rata-rata lama waktu pada kinerja *render* sebanyak 245ms atau 4,78% dan rata-rata lama waktu pada kinerja *painting* sebanyak 161ms atau 3.14%.

Gambar 3. Diagram Kinerja *Browser*

Sedangkan lama waktu *GPU* dalam menjalankan animasi *SVG* dalam menampilkan gambar dan menjalankan animasi, rata-rata lama waktu *GPU* bekerja sebanyak 254ms atau 4,95% dan rata-rata lama waktu *GPU* tidak digunakan sebanyak 4873ms atau 95.05%.



Gambar 4. Diagram Lama Waktu GPU

Hasil pengujian semua durasi animasi, tidak terdapat selisih waktu yang menonjol diantara durasi pengujian. Artinya bahwa durasi animasi SVG tidak mempengaruhi kinerja *browser* dalam menjalankan animasi SVG. Hal ini dikarenakan bahwa dokumen SVG yang dijalankan melalui *browser* dijalankan dalam perintah (*values animation*) yang sama, jadi *browser* hanya menjalankan perintah yang sama sera berulang-ulang. Jika terjadi selisih kinerja *browser* kemungkinan dikarenakan karena terdapat pekerjaan yang lain yang dilakukan oleh perangkat pemroses dalam satu kegiatan.

4. Kesimpulan dan Saran

Hasil dari Perbandingan Kinerja *Browser* Pada Pemanfaatan Animasi SVG (*Scalable Vector Graphic*), bahwa tidak terdapat perbandingan kinerja *browser* yang menonjol dan animasi SVG dapat ditampilkan dan dijalankan dalam lingkungan *browser* dengan normal.

Sedangkan pada tahap evaluasi terdapat Hasil pengujian semua durasi animasi, tidak terdapat selisih waktu yang menonjol diantara durasi pengujian. Artinya bahwa durasi animasi SVG tidak mempengaruhi kinerja *browser* dalam menjalankan animasi SVG. Hal ini dikarenakan bahwa dokumen SVG yang dijalankan melalui *browser* dijalankan dalam perintah (*values animation*) yang sama, jadi *browser* hanya menjalankan perintah yang sama sera berulang-ulang. Jika terjadi selisih kinerja *browser* kemungkinan dikarenakan karena terdapat pekerjaan yang lain yang dilakukan oleh perangkat pemroses dalam satu kegiatan.

Kajian berikutnya untuk melihat kembali komponen SVG animasi dan *browser* yang digunakan, dengan menambah detail komponen SVG animasi akan menambah kinerja *browser* dan dapat dimungkinkan dengan *browser* yang berbeda mempunyai kinerja yang berbeda-beda.

DAFTAR PUSTAKA

- Babic, Niksa, Jesenka Pibernik, dan Nikola Mrvac. 2008. "Media study: Motion graphics." 2008 50th International Symposium ELMAR. Borik Zadar: IEEE. <https://ieeexplore.ieee.org/document/4747552>.
- Google. inc. 2022. *Google Developers*. 11 9. Diakses 9 11, 2022. <https://developer.chrome.com/docs/devtools/>.
- Goto, Takaaki, Tomoo Sumida, Tadaaki Kirishima, Takeo Yaku, dan Kensei Tsuchida. 2010. "Automatic Generation of SVG Program Documents with Animation Based on Attribute Graph Grammars." 2010 IEEE/ACIS 9th International Conference on Computer and Information Science. Yamagata: IEEE. doi:10.1109/ICIS.2010.120.
- Kramer, Jan-Peter, Michael Hennings, Joel Brandt, dan Jan Borchers. 2016. "An Empirical Study of Programming Paradigms for Animation." 2016 IEEE/ACM Cooperative and Human Aspects of Software Engineering (CHASE). Austin: IEEE.
- Kurniawan, Ichwan, Much. Rifqi Maulana, dan Dicke JSH Siregar. 2020. "PEMANFAATAN SCRIPT PYTHON UNTUK MEMBUAT 3D ANIMATION PADA OPEN-SOURCE SOFTWARE." *IC-TECH Journal of Informatic and Computer Technology* (P3M STMIK Widya Pratama) 15: 97-109.

doi:<https://doi.org/10.47775/ictech.v15i1.98>.

- Liao, Chia-Wei, Huai-Che Lee, Kai-Wei Chen, Zong-Hong Lyu, Jih-Rong Chen, Wen-Jen Ho, dan Jung Hong Chuang. 2010. "SVG engine design and optimization." *IEEE International Symposium on Consumer Electronics (ISCE 2010)*. Braunschweig. doi:10.1109/ISCE.2010.5522688.
- Pramono, Waris, M Suyanto, dan Amir Fatah Sofyan. 2017. "PERBANDINGAN METODE FRAME BY FRAME DAN EXPRESSION DALAM PEMBUATAN ANIMASI DUA DIMENSI." *Prosiding SEMNASTEK Fakultas Teknik Universitas Muhammadiyah Jakarta*. Jakarta: Universitas Muhammadiyah Jakarta.
- Sari, Anisa Rayinda, Hestiasari Rante, dan Nu Rhahida Arini. 2021. "Implementation of Liquid Animation Techniques for Developing Renewable Energy Motion Graphics." *2021 International Electronics Symposium (IES)*. Surabaya: IEEE. doi:10.1109/IES53407.2021.9593956.
- V. Patil, Ganesh, dan Santosh L. Deshpande. 2016. "Distributed rendering system for 3D animations with Blender." *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*. Pune: IEEE.
- Xiao, Chen. 2020. "A Study of Differences of Resource Deformation Tools and Modules in Animate throughout the Motion Graphics Creation." *2020 International Conference on Innovation Design and Digital Technology (ICIDDT)*. Zhenjing: IEEE. doi:10.1109/ICIDDT52279.2020.00113.