

ANALISIS MEMORY PERFORM BROWSER PADA PENGGUNAAN AUDIO WEB PROGRAMMING

Ichwan Kurniawan¹⁾, Much. Rifqi Maulana²⁾, Arochman Arochman³⁾

STMIK Widya Pratama¹²³⁾

ichwan.ana10@gmail.com¹⁾, rifqi@stmik-wp.ac.id²⁾, arochman.aryanta@gmail.com³⁾

Abstrak

Sebuah game dikembangkan dengan melibatkan banyak asset, salah satunya pemanfaatan audio dalam pengembangan game. Pembuatan efek suara dalam sebuah game dilakukan dengan merekamnya secara terpisah dengan proses pengembangan game, hal ini dapat memakan waktu dan sangat tidak efisien untuk diselesaikan. Oleh karena itu, pembuatan efek suara dinamis diperlukan untuk mempersingkat waktu produksi dan menyelesaikan game lebih cepat. Dengan mengkombinasikan audio dengan pemrograman memungkinkan penyajian audio dalam sebuah game dapat disajikan secara dinamis, hal ini dikarenakan dengan algoritma pemrograman audio dapat dikontrol berdasarkan waktu, kejadian dan perilaku tertentu dalam sebuah game. Hasil dari Analisis perform memory browser dalam menjalankan file notasi suara maupun kode frekuensi notasi suara adalah jika dilihat perbandingan rata-rata browser google chrome pada saat penggunaan array memiliki rata-rata 405,92 kb untuk browser google chrome, hal ini menunjukkan bahwa terdapat perbedaan yang sangat besar terhadap penggunaan memory array dalam menjalankan file dan frekuensi. Penggunaan file audio menunjukkan bahwa dalam file load audio membutuhkan memory array yang cukup besar, sedangkan dalam penggunaan frekuensi audio menunjukkan bahwa load audio tidak membutuhkan memory array yang tidak begitu besar.

Kata Kunci : *Audio Programming, Sound Effect, Gama Development.*

1. Pendahuluan

Pengembangan game moderen peran *audio* memainkan peran yang penting, penggunaan *audio* dapat mengekspresikan momen atau tindakan pada waktu tertentu dalam permainan dan menekankan pengalaman visual dari apa yang dilihat pemain, dengan demikian, dapat meningkatkan kesan imersif bagi pemain (Pramana, Yuniarno, Nugroho, & Suprpto, 2022). Soundtrack video game dapat memengaruhi persepsi pemain secara signifikan, terutama bila terdapat hubungan intuitif antara fitur musik dan game (Makhmutov, Brown, Surkov, Timchenko, & Timchenko, 2022). Selain visual yang menonjol pada sebuah game, penambahan audio menjadi hal yang penting dalam keberhasilan game. Penggunaan audio dapat mempengaruhi respon psikis suasana tindakan pemain dalam bermain game.

Audio adalah faktor yang sangat diperlukan dan memandu emosi pemain dalam game (Huang & Cheng, 2022). Pemanfaatan audio yang baik, memungkinkan meningkatkan kesan pemain terhadap sebuah permainan. Kombinasi

audiovisual yang baik akan mempengaruhi nilai permainan dalam permainan, tantangan dalam sebuah game akan terabaikan dengan kenikmatan mendengarkan audio yang nyaman. Namun, menciptakan ketepatan audio dengan visual pada sebuah game menjadi hal yang meropotkan, kejadian dan suasana dikontrol dalam waktu tertentu.

Penggunaan audio dalam sentral permainan dapat menjadikan permainan menyenangkan, menarik, dan pada akhirnya instruktif (Aline, Dombrowsky, & Sporka, 2021). Audio adalah faktor yang sangat diperlukan dan memandu emosi pemain dalam game (Huang & Cheng, 2022). Pemanfaatan audio yang baik, memungkinkan meningkatkan kesan pemain terhadap sebuah permainan. Kombinasi audiovisual yang baik akan mempengaruhi nilai permainan dalam permainan, tantangan dalam sebuah game akan terabaikan dengan kenikmatan mendengarkan audio yang nyaman. Namun, menciptakan ketepatan audio dengan visual pada sebuah game menjadi hal yang

meropotkan, kejadian dan suasana dikontrol dalam waktu tertentu.

Pembuatan efek suara dalam sebuah game dilakukan dengan merekamnya secara terpisah dengan proses pengembangan game, hal ini dapat memakan waktu dan sangat tidak efisien untuk diselesaikan. Oleh karena itu, pembuatan efek suara dinamis diperlukan untuk mempersingkat waktu produksi dan menyelesaikan game lebih cepat (Pramana, Yuniarno, Nugroho, & Suprpto, 2022). Control audio dalam pengemasan permainan memanfaatkan sebuah algoritma pemrograman untuk menentukan waktu audio dijalankan (Khan, Nguyen, Dai, & Thawonmas, 2022). Dengan mengkombinasikan audio dengan pemrograman memungkinkan penyajian audio dalam sebuah game dapat disajikan secara dinamis, hal ini dikarenakan dengan algoritma pemrograman audio dapat dikontrol berdasarkan waktu, kejadian dan perilaku tertentu dalam sebuah game.

Uraian di atas menunjukkan bahwa dalam menciptakan *game sound effect* dilakukan diluar pengembangan *game* dan membutuhkan sumber daya *audio* yang tidak sedikit. Hal ini berdampak pada peningkatan penggunaan sumber daya komputer yang digunakan. Penelitian ini akan dilakukan analisis *memory perform browser* pada penggunaan *audio web programming*.

Berdasarkan latar belakang maka dapat diambil rumusan masalahnya adalah “Apakah Penggunaan *audio web programming* dapat mempengaruhi *memory perform browser*?”. Sedangkan tujuan dari penelitian ini adalah Analisis *Memory Perform Browser* pada Penggunaan *Audio Web Programming*. Kemudian manfaat dari penelitian ini adalah memberikan gambaran kepada pengembangan *game* tentang pembuatan *audio assets game* dengan menggunakan *audio web programming*.

2. Metode Penelitian

Penelitian ini adalah penelitian eksperimen dengan metode penelitian sebagai berikut:

- Penentuan masalah penelitian adalah dengan menggunakan studi literatur dan studi lapangan.
- Penentuan *Computing Approach* penelitian ini dipilih studi literatur mengenai Pembuatan efek suara dalam sebuah game dilakukan

dengan merekamnya secara terpisah dengan proses pengembangan game, hal ini dapat memakan waktu dan sangat tidak efisien untuk diselesaikan .

- Analisis *Audio Pemrograman* untuk Pengembangan *Game Sound Effect Berbasis Web*
- Evaluasi dengan melakukan komparasi dengan data empiris.
- Pengujian dengan melakukan perbandingan *perform memory browser* pada saat menjalankan *game audio effect* pada lingkungan *browser google chrome*

3. Hasil dan Pembahasan

3.1 Pengumpulan Data

Populasi Data yang digunakan dalam penelitian ini adalah berhubungan dengan masalah yang diteliti, data tersebut adalah data yang berhubungan dengan sound pada lingkungan browser. Data sound dalam hal ini berupa data nilai frekuensi notasi atau data file notasi pada lingkungan browser, data nota nada yang digunakan merupakan 12 notasi nada, dalam hal ini nada C, C#, D, D#, E, F, F#, G, G#, A, A# dan B.

Penggunaan notasi nada dengan frekuensi menggunakan tingkat oktav ke 4, detail data yang digunakan sebagai berikut.

Tabel 1. Data Frekuensi dan File Notasi Nada

Key	Note	Freq	*.Wav
A	C	130.8	40
W	C#	138.6	41
S	D	146.8	42
E	D#	155.6	43
D	E	164.8	44
F	F	174.6	45
T	F#	185.0	46
G	G	196.0	47
Y	G#	207.7	48
H	A	220.0	49
U	A#	233.1	50
J	B	246.9	51

3.2 Penentuan *Computing Approach*

Computing Computing approach pada penelitian ini adalah implementasikan penggunaan *audio programming* dalam pengembangan *game sound effect* pada lingkungan *browser google chrome* dan *microsoft edge*. Penggunaan *audio programming* dipilih dengan alasan bahwa *sound asset game* tidak harus dilakukan secara terpisah dalam pengembangan *game*, dengan menggunakan *audio programming, sound asset game* dalam dilakukan bersamaan dalam pengkodean bahasa pemrograman.

Penggunaan perintah *javascript* untuk memainkan notasi file **.wav* sebagai berikut.

```
function playNote(e) {
  const audio = document.querySelector('audio[data-key=${e.keyCode}]');
  key = document.querySelector(`.key[data-key=${e.keyCode}]`);

  if (!key) return;

  const keyNote = key.getAttribute("data-note");

  key.classList.add("playing");
  note.innerHTML = keyNote;
  audio.currentTime = 0;
  audio.play();
}
```

Gambar 1 Script Fungsi *playNote File*

Script di atas digunakan untuk memainkan file **.wav* pada saat key tombol keyboard ditekan. Fungsi ini menerima variabel yang dikirim dari halaman HTML dalam bentuk *querySelector, selector* tersebut menyimpan nilai kunci keyboard dan notasi nada yang akan dibunyikan.

Sedangkan penggunaan perintah *javascript* untuk memainkan frekuensi nada sebagai berikut.

```
function playNote(e) {
  const key = document.querySelector(`.key[data-key=${e.keyCode}]`);

  if (!key) return;

  const keyNote = key.getAttribute("data-note");
  const k_Note = key.getAttribute("data-note_k");

  key.classList.add("playing");
  note.innerHTML = keyNote;
  task_note_melody(k_Note, type);
}
```

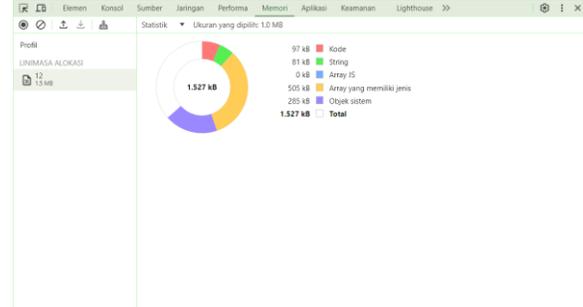
Gambar 2 Script Fungsi *playNote Frequence*

3.3 Implementasi

Implementasi *audio programming* dengan *javascript*, dalam memainkan 12 notasi nada, C, C#, D, D#, E, F, F#, G, G#, A, A# dan B. Akan dibandingkan kinerja *browser memory* baik menggunakan *google chrome* maupun *microsoft Edge* dalam penggunaan frekuensi notasi

maupun audio file notasi (**.wav*). Kinerja yang akan diukur dalam hal ini adalah kapasitas penggunaan *object array* dalam pemrograman.

Tools yang digunakan untuk pengukuran perform *browser* dalam hal ini adalah *Google Chrome DevTools*.



Gambar 3 *Google Chrome DevTools*

Analisis yang akan dilakukan dalam kinerja *memory* dalam lingkungan *browser* adalah dalam penggunaan *array* dalam satuan *kilobytes (kb)*.

3.4 Pembahasan

Tabel 2. Hasil Pengukuran Kinerja *Browser Google Chrome* dengan File (*wav*).

No	Key	Note	Array (Kb)
1	A	C	449,00
2	W	C#	455,00
3	S	D	460,00
4	E	D#	475,00
5	D	E	480,00
6	F	F	485,00
7	T	F#	491,00
8	G	G	496,00
9	Y	G#	501,00
10	H	A	507,00
11	U	A#	500,00
12	J	B	505,00

Tabel 3. Hasil Pengukuran Kinerja *Browser Google Chrome* dengan Frekuensi Notasi

No	Key	Note	Array (Kb)
1	A	C	42,00
2	W	C#	48,00
3	S	D	54,00
4	E	D#	58,00
5	D	E	64,00
6	F	F	82,00
7	T	F#	87,00
8	G	G	93,00
9	Y	G#	85,00
10	H	A	104,00
11	U	A#	109,00
12	J	B	107,00

Pada pengukuran kinerja *memory browser* pada *google chrome* kecenderungan meningkat penggunaan *memory* dalam penggunaan *array*. Penggunaan file *.wav dengan penggunaan frekuensi terlihat lebih rendah dalam penggunaan *memory browser* baik *google chrome*.

3.5 Evaluasi

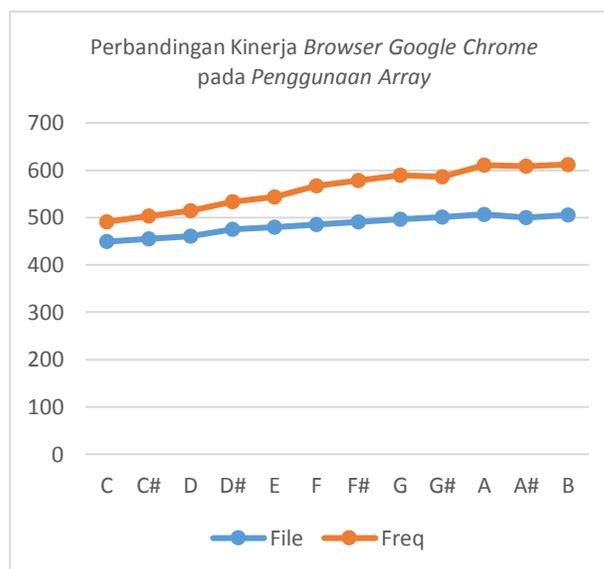
Tahap pengujian ini akan dilakukan beberapa tahap, dengan menguji kapasitas *memory* dalam penggunaan *browser* dengan menggunakan *browser google chrome*, pengujian selanjutnya akan diuji dalam penggunaan notasi suara, yaitu penggunaan notasi dengan file *.wav atau dengan kode frekuensi notasi.

Tabel 4. Perbandingan Kinerja *Browser Google Chrome* pada Penggunaan *Array*

No	Note	Array (Kb)		Perbandingan
		File	Freq	
1	C	449,00	42,00	407,00
2	C#	455,00	48,00	407,00
3	D	460,00	54,00	406,00
4	D#	475,00	58,00	417,00
5	E	480,00	64,00	416,00
6	F	485,00	82,00	403,00
7	F#	491,00	87,00	404,00
8	G	496,00	93,00	403,00

No	Note	Array (Kb)		Perbandingan
		File	Freq	
9	G#	501,00	85,00	416,00
10	A	507,00	104,00	403,00
11	A#	500,00	109,00	391,00
12	B	505,00	107,00	398,00
Rata-Rata		483,67	77,75	405,92

Sedangkan dalam penggunaan *array* dalam penggunaan file (*.wav) rata-rata kapasitas *memory* yang digunakan *browser* adalah 483,67 kb dalam menjalankan 12 notasi nada. Sedangkan dengan penggunaan kode frekuensi rata-rata kapasitas *memory* yang digunakan *browser* adalah 77,75 kb, kemudian rata-rata perbandingan kapasitas *memory* yang digunakan *browser* adalah 405,92 kb. Kemungkinan hal ini disebabkan karena pada saat pemanggilan file notasi membutuhkan program membutuhkan *object array*, sedangkan dengan kode frekuensi notasi tidak membutuhkan *object array*.

Gambar 4 Grafik Kinerja *Browser Chrome* pada Penggunaan *Array*

4. Kesimpulan dan Saran

Hasil dari Analisis perform *memory browser* dalam menjalankan file notasi suara maupun kode frekuensi notasi suara adalah jika dilihat perbandingan rata-rata *browser google chrome* pada saat penggunaan *array* memiliki rata-rata 405,92 kb untuk *browser google chrome*, hal ini

menunjukkan bahwa terdapat perbedaan yang sangat besar terhadap penggunaan *memory array* dalam menjalankan file dan frekuensi. Penggunaan *file audio* menunjukkan bahwa dalam *file load audio* membutuhkan *memory array* yang cukup besar, sedangkan dalam penggunaan frekuensi *audio* menunjukkan bawah frekuensi *load audio* tidak membutuhkan *memory array* yang tidak begitu besar.

Hasil evaluasi yang telah dilakukan menunjukkan bahwa, penggunaan file notasi (*.wav) mengkonsumsi banyak kapasitas *memory* dibanding dengan penggunaan kode frekuensi notasi, dalam hal ini penggunaan *object array* dengan menggunakan *google chrome*.

Daftar Pustaka

- Aline, H., Dombrowsky, A., & Sporcka, A. (2021). Soundoku: A sound puzzle game. *2021 IEEE Conference on Games (CoG)*. Copenhagen, Denmark: IEEE. doi:10.1109/CoG52621.2021.9618984
- Huang, M.-T., & Cheng, C.-W. (2022). Influence of Sound Direction and Distance on Immersive Experience in VR Gaming. *2022 IEEE 5th International Conference on Knowledge Innovation and Invention (ICKII)*. Hualien, Taiwan: IEEE. doi:10.1109/ICKII55100.2022.9983604
- Khan, I., Nguyen, T., Dai, X., & Thawonmas, R. (2022). DareFightingICE Competition: A Fighting Game Sound Design and AI Competition. *2022 IEEE Conference on Games (CoG)*. Beijing, China: IEEE. doi:10.1109/CoG51982.2022.9893624
- Makhmutov, M., Brown, J. A., Surkov, M., Timchenko, A., & Timchenko, K. (2022). Adaptive Game Soundtrack Tempo Based on Players' Actions. *2022 IEEE Conference on Games (CoG)*. Beijing, China: IEEE. doi:10.1109/CoG51982.2022.9893604
- Pramana, F. S., Yuniarno, M. E., Nugroho, S. M., & Suprpto, Y. K. (2022). Dynamic Sound Effect Based on Distance of Interactive Object using Linear Interpolation Method in Unity3D. *2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*. Surabaya, Indonesia: IEEE. doi:10.1109/CENIM56801.2022.10037563