

## VISUALISASI MODEL 3D DINAMIS BERBASIS WEB MENGGUNAKAN WEBGL

**Moch Fachri, Risanto Darmawan**

Universitas Krisnadwipayana, Indonesia

Email: moch.fachri@unkris.ac.id, risantodarmawan@unkris.ac.id

### Abstrak

Penelitian ini memanfaatkan suatu *graphic library* yang memungkinkan visualisasi model dimensi 3(3D) pada *website*. *Graphic library* yang dimaksud adalah WebGL. Visualisasi objek di dunia nyata dengan foto biasa tidak mewakili perspektif dimensi tiga dari objek tersebut. Hal ini dikarenakan foto diambil dari satu perspektif, kemudian memproyeksikan objek dunia nyata yang memiliki ukuran volumetrik kedalam bidang planar dalam bentuk citra digital. Karena itulah WebGL sebagai *graphic library* yang bersifat *open source* dan mendukung penggunaan dalam *website*, dapat menjadi solusi dalam visualisasi 3D dalam bentuk aplikasi berbasis *web*. Hasil yang didapat menunjukkan tampilan visualisasi 3D yang tertanam pada *server website*, dan dapat ditampilkan dalam antarmuka *client* yang mengakses konten visualisasi pada *server* tersebut.

**Kata Kunci:** Visualisasi 3D; WebGL; *Graphic library*

### Abstract

*This research employs a graphic library that allows the visualization of 3D models on the website. The graphics library in question is WebGL. Visualization of objects in the real world with the digital image is unable to represent the three-dimensional perspective of the object. This is because the picture is taken from a single point of capture, then projects a real-world object that has a volumetric size into a planar plane in the form of a digital image. That's why WebGL as a graphic library that is open source and supports use on websites, can be a solution in 3D visualization in the form of web-based applications. The results obtained show the 3D visualization display that is embedded on the website server and can be displayed in the client interface that accesses the visualization content on the server.*

**Keywords:** 3D Visualization; WebGL; *Graphic library*

### Pendahuluan

Visualisasi dalam bentuk objek dimensi tiga memiliki kelebihan dalam menyajikan informasi visual suatu objek dibanding visualisasi dengan gambar dimensi dua. Kelebihan ini yaitu ada pada representasi objek dengan berbagai titik perspektif serta informasi lengkap dimensi ukuran objek yang divisualisasikan (Storz et al.,2012);

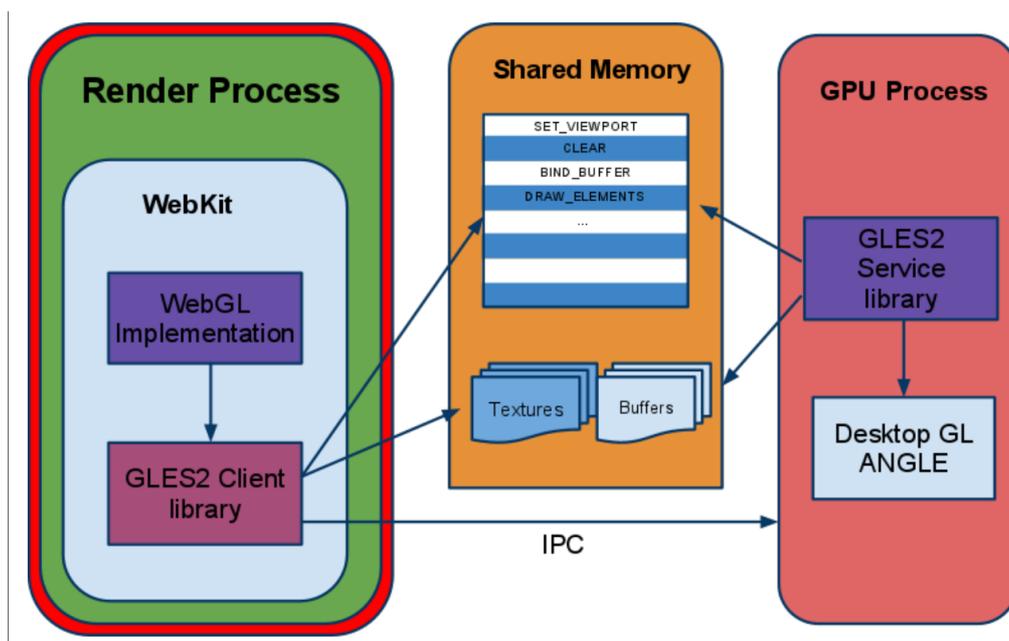
---

<b>How to cite:</b>	Moch Fachri, Risanto Darmawan (2022) Visualisasi Model 3d Dinamis Berbasis Web Menggunakan WebGL, Jurnal Syntax Literate: Jurnal Ilmiah Indonesia, (7) 10
<b>E-ISSN:</b>	2548-1398
<b>Published by:</b>	Ridwan Institute

---

(Brown et al., 2012); (Alverson et al., 2004). Hal tersebut tidak bisa didapatkan dalam visualisasi 2D (Satish and Mahendran,2019).

Visualisasi 3D masih memiliki tantangan dalam penggunaan berbasis *web*. Konten-konten yang ada pada *website* masih berfokus dengan visualisasi 2D (Speiginer and Macintyre, 2019). Untuk menjawab tantangan ini, maka telah banyak pengembang yang membuat *graphic library* untuk digunakan dalam tampilan *website*. Salah satu *graphic library* ini adalah WebGL yang dikembangkan oleh Khronos Group Inc (The Khronos Group Inc,2010). Dibandingkan dengan *graphic library* lain yang dikembangkan untuk *web* seperti VRML (Mi et al.,2019). X3D (Stefan et al., 2018), O3D (Boutsi et al., 2019), WebGL memiliki kelebihan dalam integrasi dengan konten web yang *seamless*(Saunders,2019). Selain itu bila teknologi visualisasi yang *pixel based seperti canvas* memiliki masalah performa pada visualisasi resolusi tinggi (Hristov and Petkov, 2019) dimana *hit* performa akibat WebGL tidak signifikan(Hristov and Petkov, 2019); (Liu et al.,2019). WebGL juga memiliki *modularity* dengan kombinasi *library* berbasis web lainnya (Hristov and Petkov,2019). Dengan kemampuan WebGL dalam memvisualisasikan objek 3D berbasis web, diharapkan mampu menyajikan konten pada *website* dengan dinamis.



Gambar 1

Proses WebGL dalam menampilkan render objek(The Khronos Group Inc,2020)

WebGL (*Web Graphics Library*) adalah *graphics library* yang memanfaatkan API (Application Programming Interface) dari JavaScript untuk melakukan pemrosesan grafik interaktif pada konten-konten yang diakses melalui *web browser* (Mobeen and Feng,2012). WebGL terintegrasi sepenuhnya ke semua web browser yang mendukung HTML5. Karena itulah elemen pada WebGL dapat dicampur dengan elemen HTML

serta dikombinasikan dengan bagian dari suatu halaman *web*. WebGL dirancang dan dikelola oleh organisasi non-profit Khronos Group Inc (The Khronos Group Inc,2020).

WebGL memiliki proses dalam menampilkan visualisasi objek dalam *web* seperti ditunjukkan pada gambar 1. Proses *render* dilakukan dalam oleh prosesor sentral dalam host tempat konten WebGL diimplementasikan dan oleh dikirim ke *client* berupa data-data yang dibutuhkan dalam di *web browser client* untuk penampilan visualisasi. Dimana proses service dari visualisasi di-handle oleh GPU (*Graphic Processing Unit*)(Mobeen and Feng,2012). *Web browser* dan GPU memproses data-data tersebut dalam *shared memory* yang berisi data-data 3D objek yang divisualisasikan.

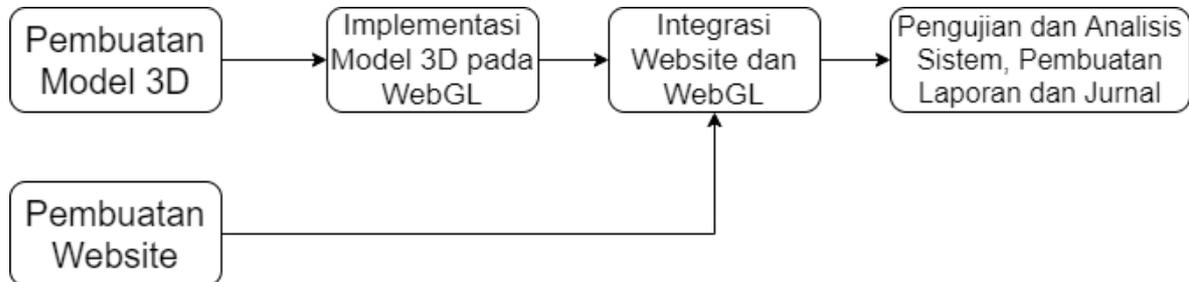
Untuk visualisasi sendiri WebGL memiliki kegunaan dalam visualisasi 3D secara luas. Salah satu contoh pada visualisasi macromolecular menunjukkan kelebihan WebGL dibandingkan alat visualisasi yang umum digunakan pada bidang tersebut (Yuan et al.,2017). Pada studi tersebut, WebGL mampu menjadi solusi *shared visualisation* secara *real-time* antar banyak pemakai di tempat berbeda secara *online* (Yuan et al.,2017). Studi lain dalam visualisasi struktur protein menunjukkan kemampuan interaktif WebGL serta kemampuannya dalam perangkat *mobile* (Shi et al.,2017). WebGL juga mampu memenuhi kebutuhan visualisasi 3D skala besar. Pada studi visualisasi tampilan geografis secara 3D, WebGL dapat menangani format 3D untuk keperluan visualisasi model geografis dengan okupasi ruang penyimpanan relatif rendah dan transmisi cepat (Miao,et al.,2017). Rerata *response time* dalam memproses *request* dari *user* kurang dari 1 ms untuk visualisasi geografis skala sebuah kota menunjukkan potensi besar WebGL ini yang memang didesain untuk aplikasi *web* yang bisa diakses berbagai *platform* (Miao,et al.,2017). Pada studi penggunaan WebGL beserta *graphics library* lain untuk visualisasi polutan udara, WebGL mampu memvisualisasikan efek partikel dengan jumlah banyak tanpa ada masalah performa yang signifikan dibandingkan dengan *graphics library* lainnya (Liu et al.,2019).

## Metode Penelitian

Tahapan pertama yang perlu dilakukan dalam penelitian ini adalah pembuatan model 3D yang akan divisualisasikan pada tampilan *web*. Pembuatan model 3D ini direncanakan menggunakan model berdasarkan benda-benda kesenian dan objek-objek lain yang memiliki *value of interest*. Contoh objek tersebut seperti vas, guci, dll. Selain itu dibutuhkan juga model lingkungan virtual sebagai latar tempat visualisasi 3D model tersebut. Dalam prosesnya dibutuhkan survey ke lapangan untuk mengambil sampel model yang akan dibuat dalam *render* 3D. Setelah model 3D tersebut selesai, selanjutnya pengimplementasian model tersebut pada WebGL. Agar bisa terintegrasi dengan baik, model 3D yang dibuat perlu disesuaikan dengan format yang di-*support* oleh WebGL.

Karena WebGL dibuat untuk kebutuhan *graphic library* pada *web*, maka disaat bersamaan perlu juga dilakukan pembuatan *website* sebagai wadah berjalannya aplikasi visualisasi. Sebagaimana ditunjukkan pada gambar 2, pembuatan *website* ini berjalan bersamaan dengan pembuatan model 3D untuk nantinya *website* yang dibuat akan

diintegrasikan dengan WebGL. Website tempat berjalannya aplikasi visualisasi ini akan *hosting* secara lokal. *Website* yang nantinya dibuat ini oleh jasa pembuatan website perlu mensupport WebGL secara *native*, yang berarti *website* ini perlu dibuat dengan HTML5.

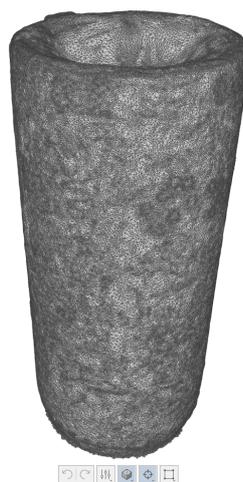


**Gambar 2**  
**Diagram Alir Penelitian**

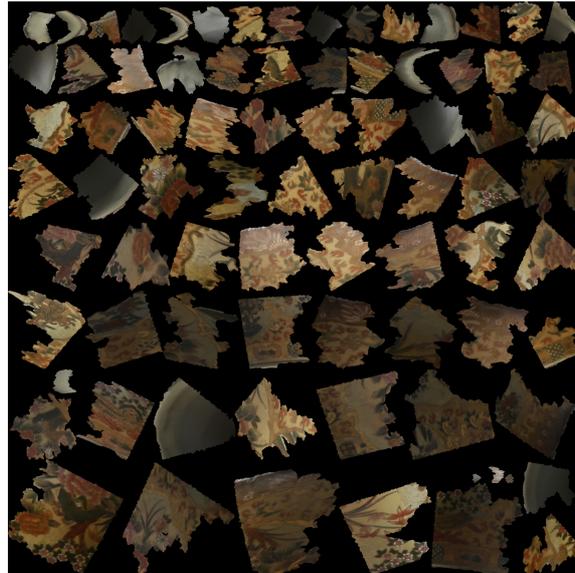
Keseluruhan proses tersebut selesai ketika aplikasi bisa berjalan dalam *webservice* lokal tanpa ada kendala dalam penampilan visualisasinya. Dengan menjalankan metode tersebut hingga tercapai integrasi dalam aplikasi berbasis *web*, konten visualisasi dapat diakses oleh pemakai dengan mengakses *web server*. Dalam penelitian ini, *web server* tempat meng-*hosting website* aplikasi dilakukan secara lokal.

### Hasil dan Pembahasan

Pembuatan model 3D didapat dari hasil 3D *scan* objek dunia nyata. Gambar 3 menunjukkan salah satu objek 3D tersebut. Model 3D pada gambar 3 menunjukkan objek guci dengan *mesh* yang belum diberi tekstur. Tekstur model tersebut diperlihatkan pada gambar 4, dimana tekstur itu nantinya akan dipasangkan pada model di gambar 3 ketika dipergunakan dalam visualisasinya.

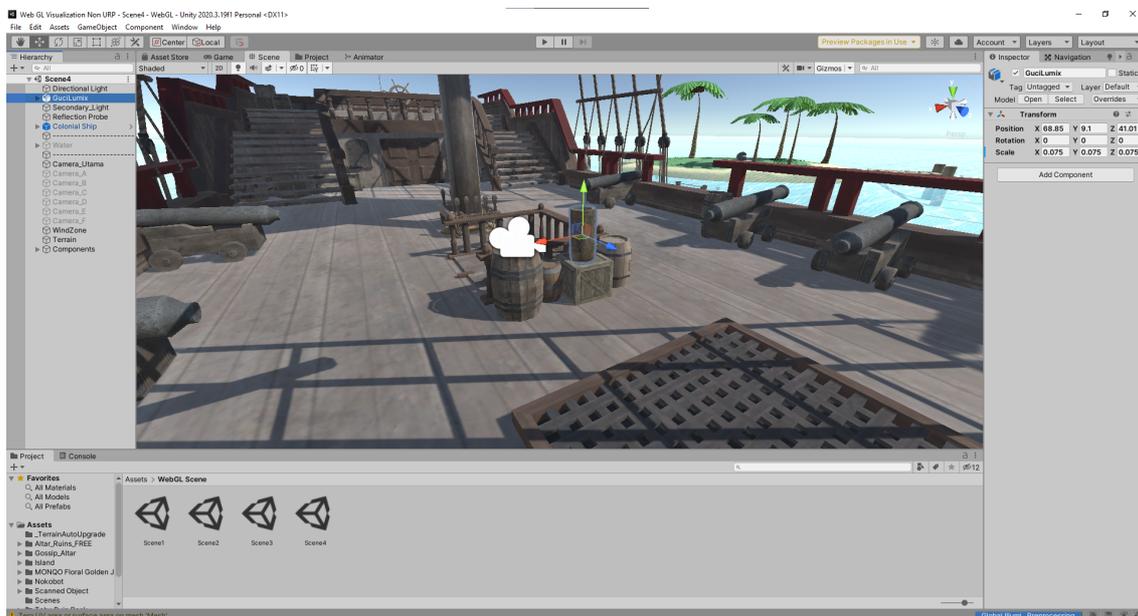


**Gambar 3**  
**Tampilan objek 3D dari hasil pemindaian objek dunia nyata**

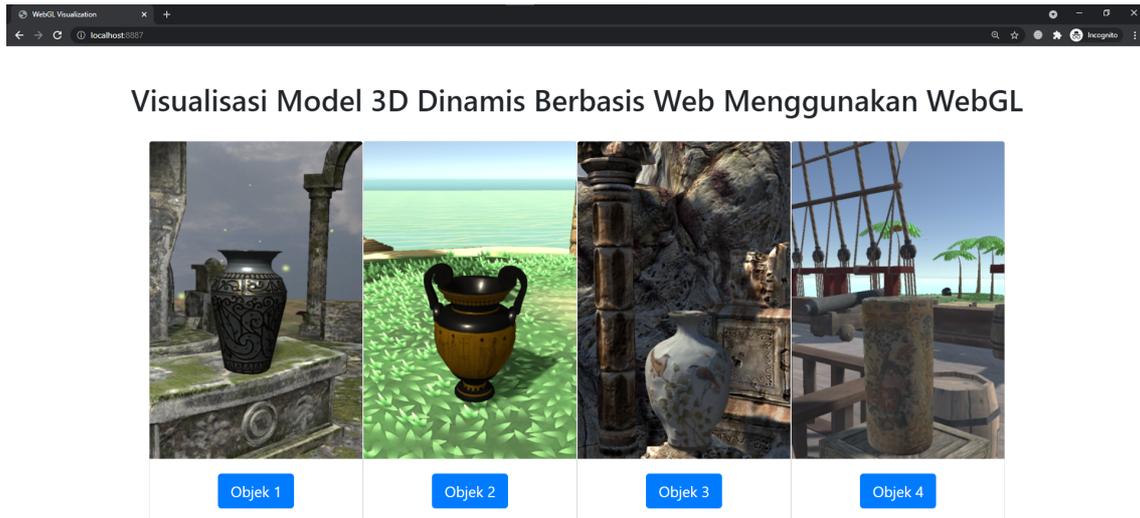


**Gambar 4**  
**Data Teksur objek 3D dari hasil pemindaian objek dunia nyata**

Untuk pengintegrasian model 3D yang akan ditampilkan dan model lingkungan, kami meutilisasi mesin permainan Unity3D yang memiliki kemampuan dalam mengkompilasi hasil 3D kedalam format yang dikenali WebGL dan men-*generate* file HTML dasar untuk diintegrasikan dalam *website* yang dibuat. Cuplikan gambar lingkungan visualisasi beserta model 3D pada mesin permainan ditunjukkan pada gambar 5.

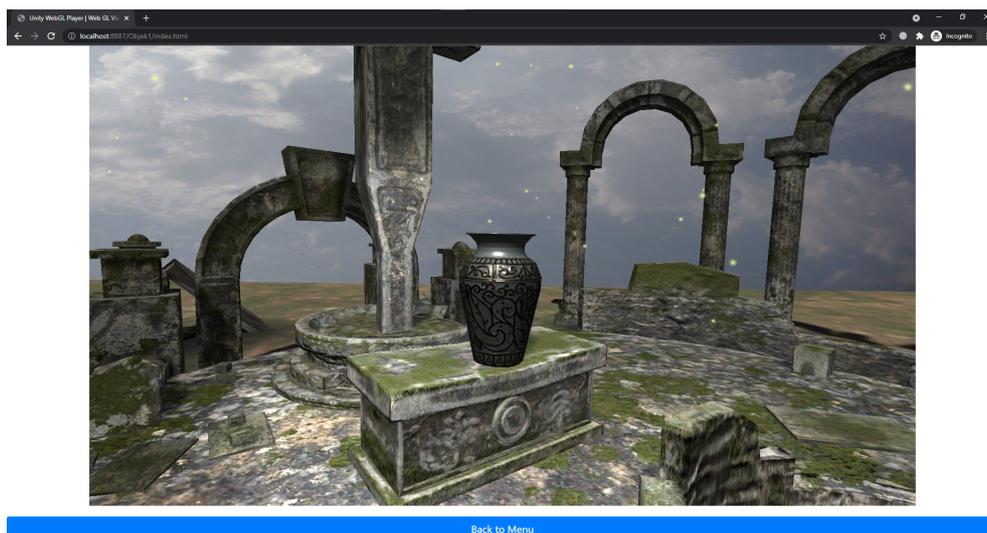


**Gambar 5**  
**Tampilan objek 3D dalam mesin permainan untuk diintegrasikan dengan WebGL**



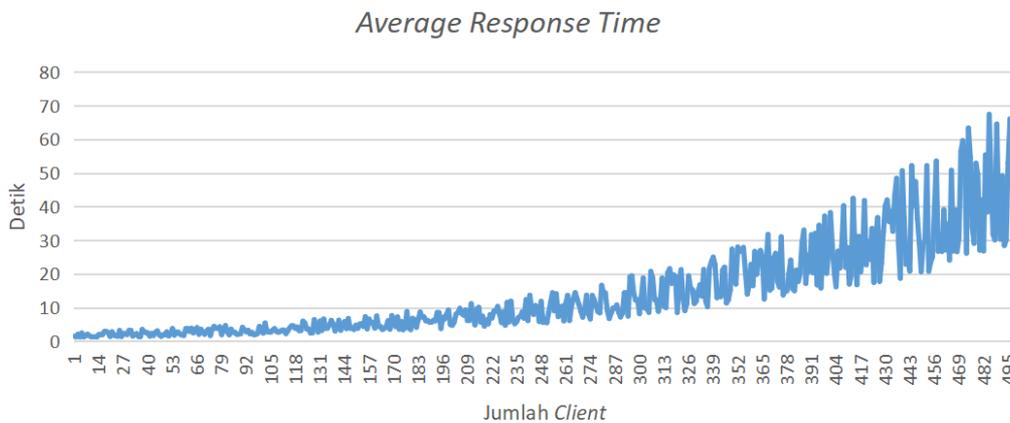
**Gambar 6**  
**Tampilan menu utama *website***

*Website* untuk menampilkan WebGL sendiri ditunjukkan pada gambar 6. pada gambar 6 tersebut menunjukkan tampilan utama dari *website* yang diakses oleh *client* terhadap *web server* secara lokal. Pengguna akan dihadapkan pada menu untuk pilihan visualisasi empat objek yang berbeda. Gambar 7 menunjukkan salah satu contoh visualisasi objek 3D. Pada contoh tersebut objek 3D beserta lingkungannya diakses menggunakan aplikasi *web browser* oleh pengguna yang mengakses *server*. Pengguna sebagai *client* disini mengirimkan *request* dalam protokol HTTPS(*Hypertext Transfer Protocol Secure*) sehingga dapat ditampilkan dengan WebGL.

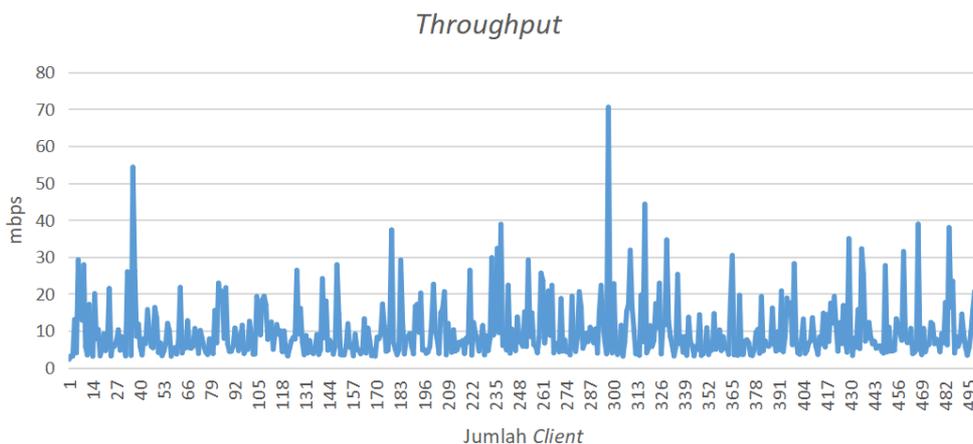


**Gambar 7**  
**Tampilan *website* ketika menampilkan visualisasi dengan WebGL**

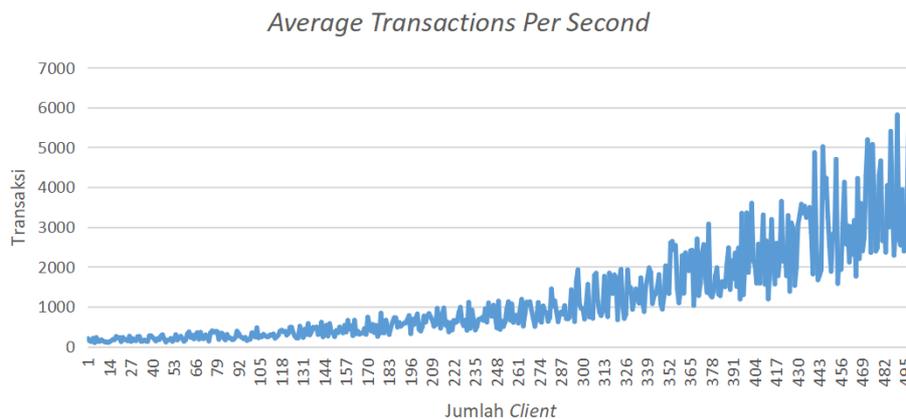
Pada pengujian ini kami menguji beban penampilan WebGL pada kemampuan *server* untuk meng-*handle* banyak *client* secara bersamaan(*concurrent*). Pengujian beban *concurrency client* dapat menggambarkan penggunaan sumber daya komputasi untuk penggunaan webGL dalam visualisasi 3D. *Web Server* dipasang secara lokal untuk terkoneksi dengan *client*. Prosesor *server* menggunakan Intel® Xeon® *processor* E3-1275 dengan memori sebesar 32GB. *Client* yang terlibat bersamaan bervariasi dari satu *client* hingga 500 *client*. Grafik pada gambar 8 menunjukkan grafik *average response time*. Nilai *average response time* menunjukkan kemampuan server dalam merespon *client*, dalam kasus ini respon terhadap request pada protokol jaringan HTTPS. Grafik gambar 8 menunjukkan kecenderungan peningkatan waktu respon seiring bertambahnya *client* yang mengakses secara bersamaan. Rerata waktu respon tertinggi selama 67,34 detik.



**Gambar 8**  
**Grafik *average response time* server webGL terhadap penambahan jumlah client**



**Gambar 9**  
**Grafik *Throughput* server webGL terhadap penambahan jumlah client**



**Gambar 10**  
**Grafik *average transaction per second* yang ditangani server webGL terhadap penambahan jumlah client**

Pada grafik *throughput* yang ditunjukkan pada gambar 9, penambahan jumlah *client* tidak menunjukkan pola. *Throughput* sendiri menunjukkan keberhasilan data yang berhasil diterima *client* dari *bandwidth* yang ada pada jaringan, nilai dari *throughput* ini lebih dipengaruhi kestabilan jaringan dan jumlah *request* dari *client* terhadap server. Data *throughput* yang kami dapat tidak menunjukkan masalah performansi *server* untuk memenuhi permintaan *client* karena tidak ada *trafik congestion* yang disebabkan proses dari *server*. *Throughput* tertinggi ada pada nilai 72,88 mbps dari *bandwidth* jaringan *gigabit ethernet* 1.000 mbps

Gambar 10 adalah rerata transaksi perdetik. *Server* semakin banyak *handle* proses transaksi data seiring bertambahnya jumlah *client*. Grafik pada gambar 10 memiliki kecenderungan yang sama dengan grafik rerata waktu respon pada gambar 8, karena semakin banyak *client* yang mengakses bersamaan, maka makin banyak transaksi data *client-server* yang ditangani sehingga waktu respon juga bertambah lama.

**Tabel 1**  
**Data Performansi Halaman Web Pada Client Yang Mengakses**

Jenis KPI(Key Performance Indicator)	Rerata Performa(detik)
<i>Time to First Byte (TTFB)</i>	1,4
<i>First Contentful Paint</i>	2,2
<i>Speed Index</i>	14,5
<i>Largest Contentful Paint</i>	13,3
<i>Time to Interactive</i>	13,4
<i>Backend Duration</i>	1,94

Adapun dari sisi *client*, responsivitas *web* yang diakses akan memengaruhi pengalaman pengguna dalam menikmati visualisasi 3D yang ditampilkan. Tabel 1 menunjukkan *key performance indicator* (KPI) dari *webserver* pada sisi *client*. *TTFB* (*Time to First Byte*) adalah jumlah total waktu dari awal permintaan hingga saat *server* menerima *byte* pertama data untuk merespon. *First Contentful Paint* menunjukkan

seberapa cepat konten seperti teks atau gambar ditampilkan ke halaman antar muka *client*. *Speed Index* menunjukkan seberapa cepat konten halaman *client* terlihat terisi sepenuhnya. *Largest Contentful Paint* menunjukkan berapa lama waktu yang dibutuhkan elemen konten terbesar untuk ditampilkan di halaman halaman antar muka *client*. *Time to Interactive* Berapa lama waktu yang dibutuhkan halaman antar muka *client* untuk bisa berinteraksi dengan *client*. *Backend Duration* adalah waktu yang dibutuhkan server untuk menghasilkan respons awal. Dari data yang didapatkan terlihat bahwa waktu *Largest Contentful Paint* mendominasi waktu yang dibutuhkan agar website yang ditampilkan di halaman *client* sudah dapat berinteraksi dengan *client* sebagai pengguna. *Largest Contentful Paint* dalam *website* yang diakses *client* tidak lain adalah konten WebGL yang merupakan konten terbesar dalam kasus layanan *web server* ini. Dari rerata waktu *Speed Index* selama 14,5 detik, WebGL menghabiskan rerata waktu untuk penampilan sebanyak 13,3 detik.

### **Kesimpulan**

Eksperimentasi yang kami lakukan mendemonstrasikan kemampuan WebGL untuk keperluan visualisasi 3D berbasis *web*. WebGL mampu menyajikan konten pada *website* dengan dinamis berkat visualisasi 3D. Visualisasi 3D yang memiliki keunggulan pada representasi objek dengan berbagai titik perspektif serta informasi lengkap dimensi ukuran objek yang divisualisasikan bila dibandingkan dengan visualisasi 2D pada citra digital. Meskipun begitu, pada studi kami terlihat bahwa konten visualisasi 3D dengan WebGL cukup membebani performa *web server* dan juga mendominasi ukuran konten dari suatu web ketika konten tersebut dimuat oleh *client*. Penggunaan visualisasi 3D dengan WebGL perlu memerhatikan kemampuan komputasi yang dimiliki *web server* serta prediksi *concurrency client* yang akan mengakses layanan *web server* tersebut.

## BIBLIOGRAFI

- Alverson, G., Eulisse, G., Muzaffar, S., Osborne, I., Taylor, L., & Tuura, L. A. (2004). IGUANA: A high-performance 2D and 3D visualisation system. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 534(1–2), 143–146. <https://doi.org/10.1016/j.nima.2004.07.036>
- Boutsis, A.-M., Ioannidis, C., & Soile, S. (2019). INTERACTIVE ONLINE VISUALIZATION OF COMPLEX 3D GEOMETRIES. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W9, 173–180. <https://doi.org/10.5194/isprs-archives-XLII-2-W9-173-2019>
- Brown, P. M., Hamilton, N. M., & Denison, A. R. (2012). A novel 3D stereoscopic anatomy tutorial. *The Clinical Teacher*, 9(1), 50–53. <https://doi.org/10.1111/j.1743-498X.2011.00488.x>
- Hristov, P., & Petkov, E. (2019). Study of 3D Technologies for Web. p. 6. in *Conference: Digital Presentation and Preservation of Cultural and Scientific HeritageAt: Bulgaria Volume: 9, Oct.2019.*
- Liu, D., Peng, J., Wang, Y., Huang, M., He, Q., Yan, Y., Ma, B., Yue, C., & Xie, Y. (2019). Implementation of interactive three-dimensional visualization of air pollutants using WebGL. *Environmental Modelling & Software*, 114, 188–194. <https://doi.org/10.1016/j.envsoft.2019.01.019>
- Mi, Q., Zhai, Z., Wang, J., Cai, Z., Cheng, L., & Wu, M. (2019). VRML-Based Investigation of Binary Compression Technology. 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2114–2117. <https://doi.org/10.1109/ICIEA.2019.8834151>
- Miao, R., Song, J., & Zhu, Y. (2017). 3D geographic scenes visualization based on WebGL. 2017 6th International Conference on Agro-Geoinformatics, 1–6. <https://doi.org/10.1109/Agro-Geoinformatics.2017.8046999>
- Mobeen, M. M., & Feng, L. (2012). High-Performance Volume Rendering on the Ubiquitous WebGL Platform. 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, 381–388. <https://doi.org/10.1109/HPCC.2012.58>
- Satish, C. J., & Mahendran, A. (2019). The effect of 3D visualization on mainframe application maintenance: A controlled experiment. *Journal of King Saud University - Computer and Information Sciences*, 31(3), 403–414. <https://doi.org/10.1016/j.jksuci.2017.03.003>

Saunders, B. V. (n.d.). Complex Variables, Mesh Generation, and 3D Web Graphics: Research and Technology Behind the Visualizations in the NIST Digital Library of Mathematical Functions. 12.

Shi, M., Gao, J., & Zhang, M. Q. (2017). Web3DMol: Interactive protein structure visualization based on WebGL. *Nucleic Acids Research*, 45(W1), W523–W527. <https://doi.org/10.1093/nar/gkx383>

SIGGRAPH LA 2010—The Khronos Group Inc. (2010). Retrieved September 1, 2020, from <https://www.khronos.org/events/siggraph-la-2010/>

Stefan, L., Hermon, S., & Faka, M. (2018). Prototyping 3D Virtual Learning Environments with X3D-based Content and Visualization Tools. 9, 15.

Storz, P., Buess, G. F., Kunert, W., & Kirschniak, A. (2012). 3D HD versus 2D HD: Surgical task efficiency in standardised phantom tasks. *Surgical Endoscopy*, 26(5), 1454–1460. <https://doi.org/10.1007/s00464-011-2055-9>

WebGL Overview—The Khronos Group Inc. (2020). Retrieved September 1, 2020, from <https://www.khronos.org/webgl/>

Yuan, S., Chan, H. C. S., & Hu, Z. (2017). Implementing WebGL and HTML5 in Macromolecular Visualization and Modern Computer-Aided Drug Design. *Trends in Biotechnology*, 35(6), 559–571. <https://doi.org/10.1016/j.tibtech.2017.03.009>

---

**Copyright holder:**

Moch Fachri, Risanto Darmawan (2022)

**First publication right:**

Syntax Literate: Jurnal Ilmiah Indonesia

**This article is licensed under:**

