

STIMULATOR SSVEP-BCI DENGAN OPENGL

Indar Sugiarto¹

¹ Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Kristen Petra

¹ Jl. Siwalankerto 121-131, Surabaya

¹ indi@petra.ac.id

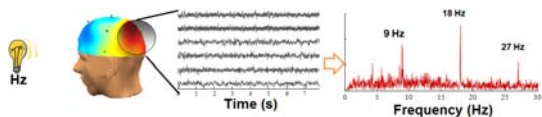
Abstrak

Dalam paper ini, sebuah program stimulator untuk aplikasi sistem BCI dengan teknik SSVEP yang memanfaatkan fitur-fitur pemrograman grafis OpenGL telah dibuat dan diuji performansinya. Program stimulator ini menghasilkan animasi berupa gambar 2D yang berkedip dengan frekuensi stabil di bawah 30Hz. Kestabilan frekuensi ini sangat penting bagi sistem BCI dengan teknik SSVEP karena teknik ini bekerja dengan cara mengklasifikasi kandungan frekuensi dalam sinyal EEG yang dirangsang oleh sebuah stimulator visual. Hasil pengujian ini juga memperkuat kesimpulan dari hasil penelitian sebelumnya bahwa frekuensi maksimum yang bisa disintensis adalah setengah dari frekuensi refresh-rate monitornya.

Kata kunci : *bci, ssvep, stimulator, OpenGL*

1. Pendahuluan

Sistem BCI (*brain-computer interface*) adalah sistem yang dirancang untuk memungkinkan komunikasi secara langsung antara manusia dengan komputer/mesin tanpa menggunakan saluran normal dari sistem saraf otak manusia [1]. Dalam sebuah sistem BCI yang memanfaatkan fitur SSVEP (*steady-state visual evoked potential*) yang diekstrak dari sinyal-sinyal EEG (*electroencephalograph*), fungsi dari stimulator eksternal sangatlah penting [2]. Stimulator eksternal tersebut bisa berupa sumber cahaya yang berkedip dengan frekuensi konstan. Frekuensi dari kedipan tersebut akan membuat sel-sel otak manusia juga akan memiliki ritme dengan frekuensi yang sama. Frekuensi kerja dari sel-sel otak tersebut akan terekam dari sinyal-sinyal EEG terutama bila sensor-sensor EEG diletakkan di bagian *ocipital region* [3]. Gambar berikut menjelaskan prinsip dasar dari sistem BCI yang menggunakan teknik SSVEP.



Gambar 1. Prinsip dasar dari SSVEP

Karena sistem BCI yang dibuat dengan teknik SSVEP sangat tergantung kestabilan frekuensi dari rangsangan yang masuk, maka stimulator yang digunakan harus memiliki performa yang bagus

dalam menghasilkan frekuensi kedipan tersebut [4]. Stimulator untuk sistem BCI dengan teknik SSVEP pada umumnya dibuat dengan dua macam cara: menggunakan lampu atau LED (*light emitting diode*) diskrit [5], atau menggunakan layar monitor untuk mengemulasikan kedipan sebuah lampu [6-8]. Pada penelitian sebelumnya, sebuah stimulator untuk sistem BCI berbasis SSVEP telah dibuat dan dikembangkan menggunakan DirectX [9]. Paper ini menjelaskan alternatif lain dari implementasi stimulator dengan menggunakan OpenGL. Performansi dari stimulator yang menggunakan OpenGL akan dibandingkan dengan stimulator sebelumnya yang menggunakan DirectX.

2. Arsitektur Stimulator

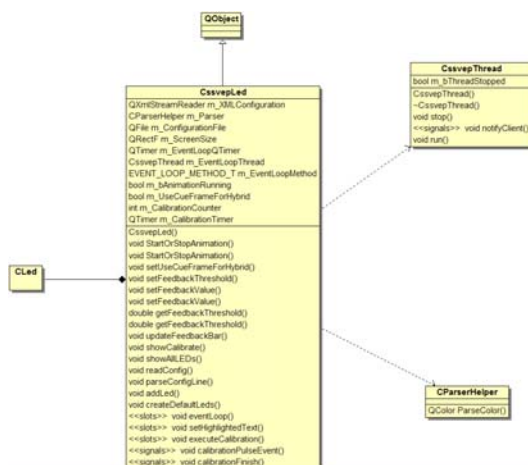
Stimulasi visual yang digunakan untuk sistem SSVEP-BCI dalam penelitian ini ditujukan untuk ditampilkan dengan menggunakan layar monitor komputer. Program dibuat untuk bisa dijalankan dalam lingkungan sistem operasi Windows XP dan Vista. Untuk menghasilkan sebuah program stimulator yang *robust*, ada tiga hal penting yang perlu diperhatikan:

- Program harus berjalan dalam lingkungan sistem operasi windows yang *multitasking* dan tidak bersifat *real-time*, yang terdiri dari puluhan *process* dan ratusan *thread* yang berjalan secara *concurrent*. Program stimulator akan menjadi salah satu *thread*

yang akan dijalankan oleh windows secara *concurrent*.

- Program tersebut harus memanfaatkan timer dari sistem operasi yang berjalan. Dalam lingkungan sistem operasi windows, ada dua tipe timer yang bisa digunakan, yaitu: standard timer (dengan resolusi sekitar 50ms) dan multimedia timer (dengan resolusi 1ms).
- Program tersebut harus bisa mengoptimalkan fungsi-fungsi API (*application program interface*) yang berhubungan dengan kartu grafis dari komputer yang digunakan. Ada dua teknologi API yang paling banyak digunakan saat ini jika berhubungan dengan pemrograman grafis: DirectX (*close-source*, dikembangkan oleh Microsoft) dan OpenGL (*open-source*, dikembangkan oleh Silicon Graphics).

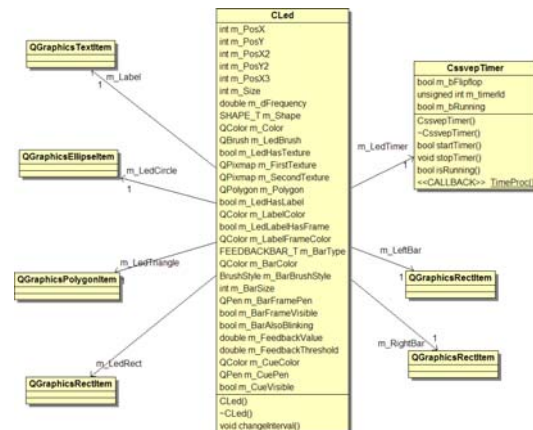
Dalam penelitian sebelumnya, program stimulator dibuat dengan bahasa pemrograman C++ dalam lingkungan Microsoft Visual Studio 2005 dan memanfaatkan DirectX SDK (*Software Development Kit*). Dalam paper ini, program stimulator dibuat dengan bahasa pemrograman C++ menggunakan g++ (GNU C++ compiler) dan juga Qt. Qt merupakan salah satu *programming framework* yang banyak dipakai dalam dunia *open-source* dan dipilih untuk pengembangan program stimulator karena Qt menyediakan fungsi-fungsi khusus yang mempermudah akses terhadap OpenGL [10]. Berikut ini adalah representasi UML (*unified modelling language*) dari program stimulator yang menggunakan OpenGL untuk *graphics rendering*.



Gambar 2. Representasi program stimulator dalam kelas CvsvepLed.

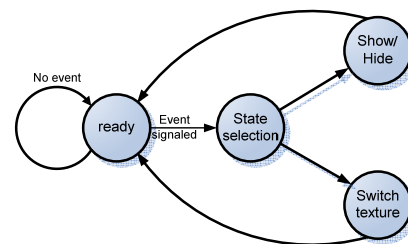
Program stimulator memiliki sebuah kelas utama yang disebut CvsvepLed. Dalam gambar 2 terlihat bahwa kelas CvsvepLed memiliki dua asosiasi: CvsvepThread (berfungsi untuk mengontrol

alur program secara *multi-threading*) dan CParserHelper (berfungsi untuk mengakses file konfigurasi yang menyimpan parameter-parameter dari animasi yang akan ditampilkan). Kelas CvsvepLed juga memiliki agregasi terhadap kelas CLed dimana kelas CLed merupakan kelas yang digunakan untuk mengatur animasi dari program stimulator. Diagram berikut menunjukkan struktur dari kelas CLed.



Gambar 3. Representasi UML dari kelas CLed. Kelas CLed merupakan kelas yang berisi instantiasi dari obyek-obyek grafis Qt untuk mengatur tampilan animasi stimulator di layar komputer.

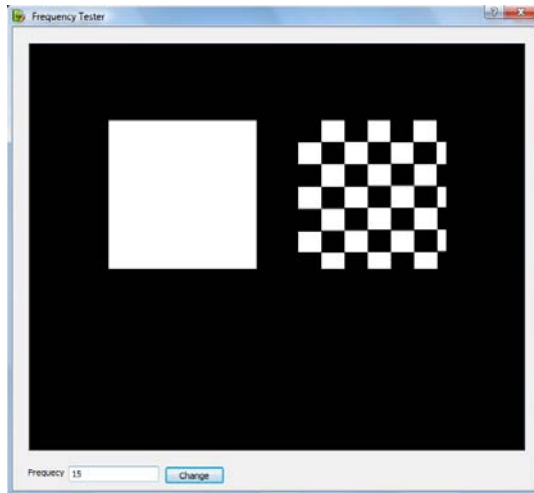
Tugas utama dari program stimulator ini adalah menghasilkan sumber cahaya yang berkedip dengan frekuensi konstan. Prinsip kerja dari program stimulator ini pada dasarnya adalah menampilkan *sprite* secara periodik menjadi sebuah animasi. Diagram berikut menunjukkan dasar kerja dari animasi program stimulator.



Gambar 4. Representasi *state machine* dari model animasi program stimulator.

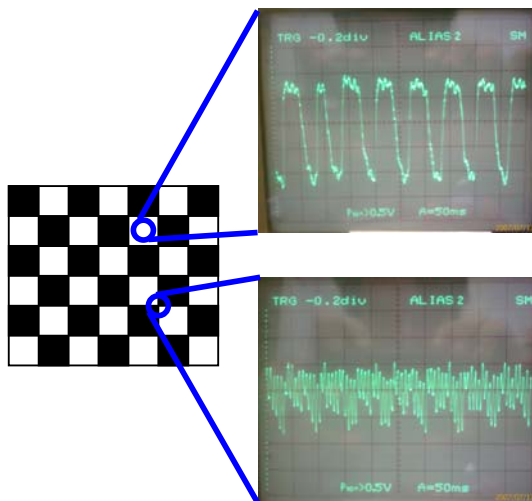
3. Pengujian

Hasil program stimulator ketika dijalankan ditunjukkan pada gambar berikut.



Gambar 5. Tampilan program stimulator ketika dijalankan. Ada dua animasi yang dipakai: animasi tanpa tekstur (gambar kiri) dan animasi dengan tekstur papan catur (gambar kanan).

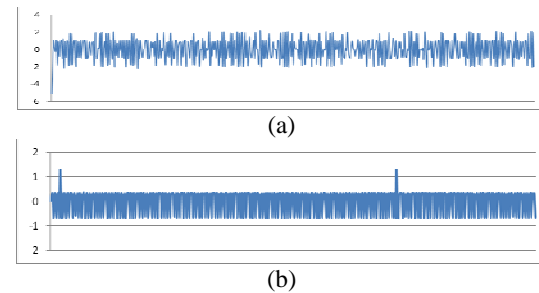
Untuk mengukur frekuensi kedipan yang dihasilkan oleh stimulator, sensor cahaya dengan komponen utama berupa photo-transistor tipe BPX-43 ditempatkan tepat di depan stimulator. Kedipan dari stimulator akan mengubah-ubah intensitas cahaya yang dipancarkan oleh layar monitor dan jika ditampilkan pada layar osiloskop akan terlihat seperti gambar berikut.



Gambar 6. Tampilan di osiloskop ketika sensor cahaya ditempatkan tepat di depan gambar animasi dari stimulator. Terlihat bahwa animasi tanpa tekstur dan animasi dengan tekstur papan catur bisa memberikan respon yang berbeda.

Selain menyediakan fungsi-fungsi yang mempermudah akses terhadap OpenGL, Qt juga menyediakan fungsi-fungsi untuk mengakses windows timer lewat obyek yang bernama QTimer. Untuk menguji performa dari QTimer dan membandingkannya dengan multimedia timer yang

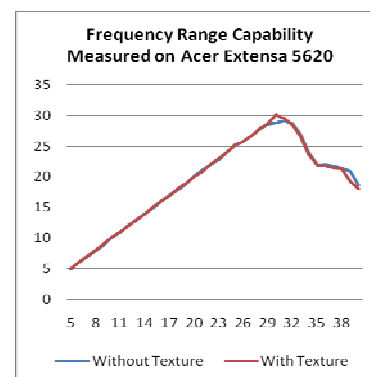
digunakan dalam penelitian sebelumnya, dilakukan pengujian dengan cara menginstantiasi QTimer dan menyampling interupsi yang dibangkitkan oleh QTimer. Grafik berikut menunjukkan deviasi jika QTimer dan Multimedia Timer dinyalakan dengan interval 33ms yang diukur dengan cara mengurangi interval interupsi dari timer dengan interval yang diinginkan. Deviasi ini menyatakan tingkat keakuratan dari timer tersebut.



Gambar 7. Grafik deviasi dari QTimer (a) dan Multimedia Timer (b) yang menunjukkan tingkat keakuratan dari masing-masing timer.

Dari grafik pada gambar 7, terlihat bahwa untuk menghasilkan interval animasi sebesar 33ms, QTimer akan memberikan deviasi sebesar 4ms sedangkan Multimedia Timer hanya memiliki deviasi sebesar 1ms. Terlihat bahwa Multimedia Timer memiliki tingkat keakuratan yang tinggi dibandingkan dengan QTimer. Itu sebabnya dalam penelitian ini, Multimedia Timer tetap digunakan sebagai pembangkit frekuensi bagi animasi dengan metode yang sama seperti dijelaskan dalam [9].

Rentang frekuensi yang dihasilkan juga diukur dalam penelitian ini. Hasilnya ditunjukkan pada gambar berikut.

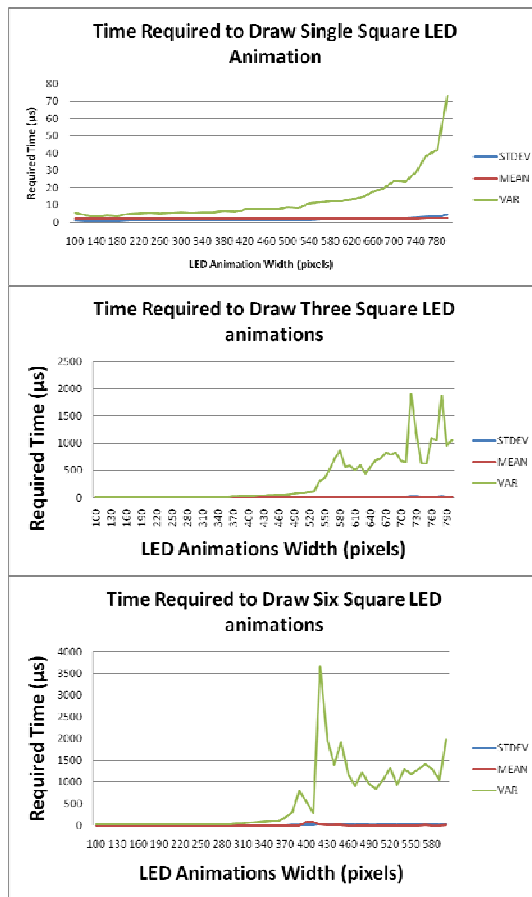


Gambar 8. Hasil pengujian rentang frekuensi yang bisa dihasilkan oleh stimulator yang diprogram dengan OpenGL. Pengukuran dilakukan baik terhadap obyek animasi bertekstur maupun tak bertekstur.

Dari hasil pengujian rentang frekuensi yang mampu dihasilkan oleh program stimulator, terlihat bahwa frekuensi tertinggi yang bisa disintesis adalah maksimum 30Hz (setengah dari *refresh-rate* monitor

yang digunakan, yaitu 60Hz). Hal ini memperkuat kesimpulan yang didapat dalam [9] bahwa frekuensi tertinggi yang bisa disintesis oleh program stimulator bagi SSVEP-BCI yang menggunakan layar monitor komputer adalah setengah dari frekuensi *refresh-rate* monitornya, yang berarti juga sesuai dengan teorema Nyquist.

Dimensi dan jumlah dari animasi yang berkedip di layar monitor juga berpengaruh terhadap kestabilan frekuensi yang dihasilkan. Untuk mengetahui pengaruhnya, dilakukan pengukuran kecepatan *rendering* dari animasi dengan mengubah-ubah ukuran dan jumlah dari animasi yang ditampilkan. Grafik berikut menunjukkan performa dari OpenGL dalam proses *rendering* animasi dengan variasi ukuran dan jumlahnya.



Gambar 9. Kecepatan *rendering* OpenGL sebagai fungsi jumlah dan ukuran animasi yang ditampilkan. Terlihat bahwa semakin banyak dan semakin besar animasi yang ditampilkan, waktu yang dibutuhkan untuk proses *rendering* juga semakin lama sehingga hasil akhir berupa frekuensi stimulator juga terpengaruh.

Grafik pada gambar 9 menunjukkan bahwa dimensi maksimum untuk sebuah animasi yang digunakan dalam program stimulator dengan konfigurasi enam buah animasi dalam satu layar

monitor seperti yang digunakan dalam [7] dan [11] adalah 340 x 340 pixel. Ukuran ini jauh lebih dari cukup karena dalam prakteknya, animasi dengan ukuran 100 x 100 pixel sudah mencukupi untuk membangkitkan sinyal EEG dengan kandungan SSVEP yang diinginkan [12].

4. Diskusi

Dari hasil-hasil pengujian pada sub-bab 3 di atas terlihat bahwa performa OpenGL tidak jauh berbeda dibandingkan dengan performa DirectX untuk *rendering* animasi berkecepatan tinggi yang digunakan sebagai stimulator bagi SSVEP-BCI. Juga terlihat bahwa program stimulator yang dibuat baik dengan DirectX maupun OpenGL akan sama-sama menghasilkan frekuensi yang stabil di bawah 30Hz. Frekuensi ini sudah lebih dari cukup mengingat bahwa dalam aplikasi SSVEP-BCI, frekuensi ideal yang bisa dipakai hanya berkisar antara 11Hz hingga 17Hz [11].

Dalam penelitian ini tidak dilakukan komparasi antara stimulator dengan metode tampilan layar monitor dengan stimulator dengan menggunakan lampu/LED diskrit. Namun secara umum, perbedaan kedua metode tersebut dapat disarikan sebagai berikut [13].

Tabel 1. Perbedaan antara stimulator menggunakan LED diskrit dan stimulator menggunakan layar monitor.

	LED-stimulator	Monitor-stimulator
Light intensity	Very high (>5000 cd/m ²) Very good even in an open area.	Low (450-500 cd/m ²) Good only in a closed room or laboratory.
Frequency resolution	High, depends on the device characteristic and capability of the embedded controller.	Low, depends on the refresh rate/response time of the monitor and also timer resolution of operating system.
Power consumption	Middle, high current is required to produce high light intensity.	High, since monitor computer will not work alone but used together with the computer itself.
Portability	Low, a dedicated	High, the

	embedded system usually employed with low level language programming.	stimulator can run on many computer types.
Design flexibility	Low, once the design is fixed it is almost impossible to change its aspects such as size, position, and orientation.	High, since computer monitor can draw almost any kind of graphical object on it.
User fatigue	Strong light intensity may make the subject quickly feel exhausted.	Modern computer monitor is designed so that a user can work with it in longer period.

5. Kesimpulan

Program stimulator untuk sistem BCI dengan teknik SSVEP telah dikembangkan dalam penelitian ini dengan memanfaatkan OpenGL. Hasil pengujian membuktikan bahwa frekuensi kedipan yang dihasilkan sangat stabil dan akurat meskipun tidak bisa lebih tinggi dari 30Hz. Frekuensi kedipan ini berasal dari animasi yang ditampilkan secara periodik di layar monitor. Hal ini memperkuat kesimpulan dari hasil penelitian sebelumnya bahwa frekuensi maksimum yang bisa disintensis adalah setengah dari frekuensi *refresh-rate* monitornya.

Daftar Pustaka:

- [1] Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., and Vaughan, T.M, 2002, "Brain-Computer Interface for Communication and Control", Clinical Neurophysiology 113.
- [2] Allison, B., McFarland, D.J., Schalk, G., Zheng, S.D., Jackson, M.M., and Wolpaw, J.R., 2008, "Towards an Independent Brain-Computer Interface Using Steady State Visual Evoked Potentials", Clinical Neurophysiology 119.
- [3] Allison, B., Sugiarto, I., Graimann, B., Gräser, A., 2008, "EEG Activity Elicited by Different Display Approaches Used in SSVEP BCI System", Neuroscience Meeting Planner, Washington, DC: Society for Neuroscience.
- [4] Herrmann, C., 2001, "Human EEG responses to 1–100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena." Exp. Brain Res. Vol.

- 137, No. 3–4. pp. 346–353.
- [5] Wang, Y., Wang, R., Gao, X., Hong, B., Gao, S., 2006, "A Practical VEP-Based Brain-Computer Interface". IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 14, No. 2.
- [6] Vidal, J., 1973, "Toward Direct Brain-Computer Communication", Annual Review of Biophysics and Bioengineering, Vol. 2, 157-80.
- [7] Lalor, EC, et al., 2005, "Steady-State VEP-Based Brain-Computer Interface Control in Immersive 3D Gaming Environment", EURASIP Journal on Applied Signal Processing, No. 19.
- [8] Müller, M., Malinowski, P., Gruber, T., Hillyard, S., 2003, "Sustained Division of The Attentional Spotlight", Letters to Nature Vol. 424.
- [9] Sugiarto, I., 2008, "Stimulator SSVEP-BCI Dengan DirectX", Industrial Electronic Seminar (IES2008), Surabaya.
- [10] Trolltech, Inc., "Creating Cross-Platform Visualization UIs with Qt and OpenGL", Trolltech Whitepaper.
- [11] Friman, O., Volosyak, I., and Gräser, A., 2007, "Multiple Channel Detection of Steady-State Visual Evoked Potentials for Brain-Computer Interfaces", IEEE Transactions on Biomedical Engineering, Vol. 54, No. 4, 742-751.
- [12] Allison, B., Sugiarto, I., Graimann, B., Gräser, A., 2008, "Display Optimization in SSVEP BCIs", Computer-Human Interaction 2008, Florence, Italy.
- [13] Sugiarto, I., 2008, "Display And Feedback Approaches for BCI Systems", Master Thesis: University of Bremen.

