

# PENGGAMBARAN KURVA LISSAJOUS MENGGUNAKAN COMPILER BORLAND C++ DENGAN LIBRARY OPENGL

Anindito Yoga Pratama

Universitas Gunadarma, anindito@staff.gunadarma.ac.id

## ABSTRAK

*Kurva Lissajous merupakan sebuah penampakan pada layar osiloskop yang mencitrakan perbedaan atau perbandingan Beda Fase, Frekuensi & Amplitudo dari 2 gelombang inputan pada probe osiloskop. Penelitian ini membahas cara pengembangan bagaimana cara penggambaran kurva lissajous dengan menggunakan library OpenGL pada compiler Borland C++ yang berdasarkan inputan nilai dari variabel oleh user, yang mana biasanya penggambaran kurva lissajous menggunakan osiloskop. Dengan adanya penggambaran menggunakan library OpenGL pada compiler Borland C++ memudahkan pengguna agar tidak lagi susah untuk mencari alat osiloskop untuk melakukan penggambaran lissajous.*

*Kata kunci: Kurva Lissajous, OpenGL, Borland C++*

## PENDAHULUAN

Pada era teknologi informasi, kita dapat menggambarkan rumus matematika, macam-macam kurva dan pemodelan lainnya. Kita tidak perlu membeli alat untuk menggambarkan hal tersebut, kita hanya melakukan pengkodean menggunakan library OpenGL pada compiler Borland C++ dan kita dapat menggambarkan sesuai yang kita inginkan, disini kita akan menggambarkan kurva Lissajous (Gökalp, 2018).

OpenGL merupakan suatu library grafis standar yang mendefinisikan sebuah cross-bahasa, cross platform API untuk menulis aplikasi yang menghasilkan komputer 2D dan 3D grafis. OpenGL bersifat Open-Source, multi-platform, dan multi language. Semua bahasa pemrograman mendukung OpenGL tetapi pada makalah ini kita menggunakan bahasa pemrograman C++ dalam pembuatannya. OpenGL bisa bekerja dalam lingkungan Windows, Linux, dll (Angel, E., 2000)(Kempf, R., 2004).

Kurva Lissajous definisinya sederhana saja, yaitu adalah sebuah penampakan pada layar osiloskop yang mencitrakan perbedaan atau perbandingan Beda

Fase, Frekuensi & Amplitudo dari 2 gelombang inputan pada probe osiloskop. Dari rumus Kurva Lissajous pada umumnya kemudian kita implementasikan ke dalam bahasa pemrograman C++ dengan menggunakan library OpenGL untuk grafik komputer, sehingga kita tidak perlu menggunakan alat seperti osiloskop hanya menginput nilai dalam program tersebut dan akan muncul penggambaran dari kurva tersebut.

## METODE PENELITIAN

Dalam penggambaran kurva lissajous berdasarkan persamaan parametrik diatas yang kemudian diimplementasikan dalam bahasa pemrograman C++ adalah penggambaran kurva lissajous yang pada dasarnya menggunakan alat osiloskop. Program C++ yang dibuat dan dimodifikasi dengan menggunakan compiler Borland C++ yang berasal dari *source code* yang menggunakan compiler GCC Linux ini, akan menggambarkan kurva lissajous dengan sebuah inputan untuk memberikan nilai dari variabel-variabel yang ada di dalam persamaan parametrik sehingga akan muncul gambar kurva lissajous

sesuai dengan inputan. Setelah melakukan pemodifikasian atas *source code* kemudian melakukan analisa terhadap perubahan apa saja yang dilakukan dan menjalankan program yang sudah dilakukan modifikasi.

## HASIL DAN PEMBAHASAN

Berdasarkan penilitan yang sudah dilakukan Burkardt (2008) pengguna menentukan informasi tentang jumlah titik, dan bentuk dari dua kurva sinus yang terlibat. Bergantung pada parameter ini, bentuk yang dihasilkan mungkin atau mungkin bukan kurva tertutup. Pengguna juga menentukan rentang untuk parameter T yang mendasarinya. Sekali lagi, tergantung pada parameter, nilai ini dapat mengakibatkan kurva tertutup ditarik beberapa kali, atau kurva tertutup hanya ditarik sebagian. Tetapi yang dilakukan di dalam penelitian oleh Burkardt (2008) adalah menggunakan Open GL untuk sistem operasi OS X sehingga tidak semua orang memiliki sistem operasi tersebut, sehingga di dalam paper ini penulis membuat penggambaran kurva lissajous menggunakan Borland C++ untuk lebih memudahkan pengguna yang menggunakan sistem operasi Windows dengan penambahan dan pengubahan dari *source code* yang sudah ada (Burkardt, J., 2008). Berikut ini

merupakan *source code* yang telah dilakukan perubahan menggunakan Borland C++ sebagai berikut:

### Header

Dari Tabel 1 di atas dapat kita lihat bahwa untuk penulisan *script* untuk compiler Borland C++ dapat langsung menghapuskan huruf 'c' yang terdapat di dalam <...>

### Program Utama

Dari perbandingan Tabel 2 dapat kita lihat bahwa untuk Bagian 1 – 3 hanya merubah untuk *output* menjadi Bahasa Indonesia, untuk Bagian 4 merubah parameter *r8\_pi* menjadi *pi*, untuk Bagian 5 adalah merubah ketebalan dan warna dari kurva.

Selanjutnya setelah melakukan modifikasi seperti penjelasan di atas, apabila kita menjalankan program ini pertama kali akan muncul adalah sebuah inputan seperti pada Gambar 1.

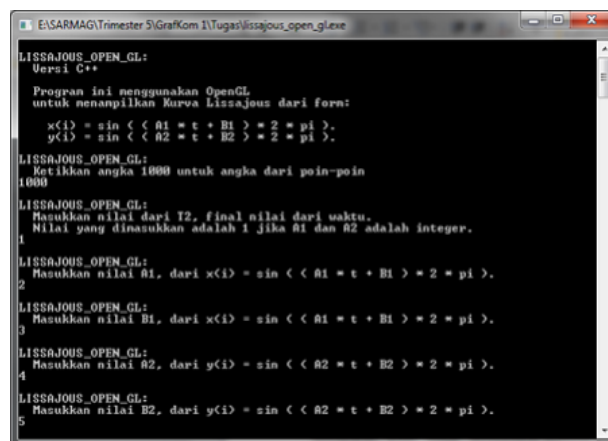
Kemudian setelah kita melakukan inputan, maka akan muncul kurva yang sesuai dengan nilai yang kita ingin untuk diinputkan seperti pada Gambar 2. Dapat dilihat dari Gambar 2 adalah hasil yang digambarkan sesuai dengan apa yang sudah kita lakukan perubahan ke dalam program seperti warna dan ketebalan kurva.

**Tabel 1.**  
**Perbandingan Header**

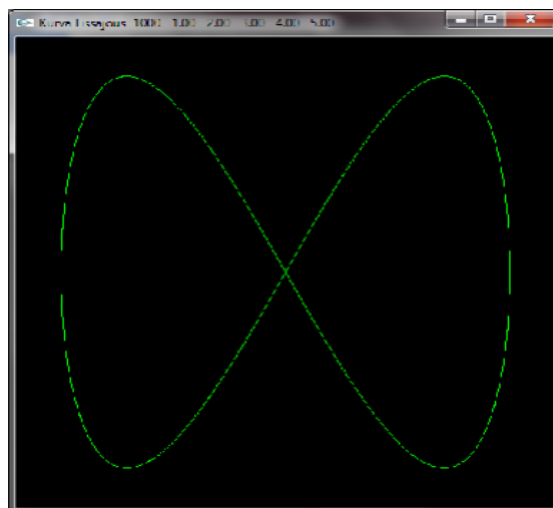
Source Code	Modifikasi
# include <cmath>	# include <stdio>
# include <cstdlib>	# include <stdlib>
# include <cstdio>	# include <math>

**Tabel 2.**  
**Perbandingan Program Utama**

	Source Code	Modifikasi
Bagian 1	<pre>cout &lt;&lt; "\n"; cout &lt;&lt; "LISSAJOUS_OPENGL:\n"; cout &lt;&lt; " C++ version\n"; cout &lt;&lt; "\n"; cout &lt;&lt; " This is a program which uses OpenGL\n"; cout &lt;&lt; " to display a Lissajous figure of the form:\n"; cout &lt;&lt; "\n"; cout &lt;&lt; " x(i) = sin ( ( A1 * t + B1 ) * 2 * pi ).\n"; cout &lt;&lt; " y(i) = sin ( ( A2 * t + B2 ) * 2 * pi ).\n";</pre>	<pre>cout &lt;&lt; "\n";cout &lt;&lt; "LISSAJOUS_OPEN_GL:\n"; cout &lt;&lt; " Versi C++\n\n"; cout &lt;&lt; " Program ini menggunakan OpenGL\n"; cout &lt;&lt; " untuk menampilkan Kurva Lissajous dari form:\n\n"; cout &lt;&lt; " x(i) = sin ((A1 * t + B1) * 2 * pi).\n"; cout &lt;&lt; " y(i) = sin ((A2 * t + B2) * 2 * pi).\n";</pre>



Gambar 1. Tampilan Input Program



Gambar 2. Tampilan Output Program

<b>Bagian</b> 2	<pre> if ( 2 &lt;= argc ){n = atoi ( argv[1] );} else { cout &lt;&lt; "\n";cout &lt;&lt; "LISSAJOUS_OPENGL:\n"; cout &lt;&lt; " Please enter N, the number of points.\n";cout &lt;&lt; " A reasonable number might be 1000.\n";cin &gt;&gt; n; } if ( 3 &lt;= argc ){t2 = atof ( argv[2] );} else { cout &lt;&lt; "\n";cout &lt;&lt; "LISSAJOUS_OPENGL:\n";cout &lt;&lt; " Please enter T2, the final value of time.\n"; cout &lt;&lt; " A value of 1 is reasonable if A1 and A2 are integers.\n";cin &gt;&gt; t2; } if ( 4 &lt;= argc ){a1 = atof ( argv[3] );} else { cout &lt;&lt; "\n";cout &lt;&lt; "LISSAJOUS_OPENGL:\n";cout &lt;&lt; " Please enter A1, for x(i) = sin ( ( A1 * t + B1 ) * 2 * pi ).\n";cin &gt;&gt; a1; } if ( 5 &lt;= argc ){b1 = atof ( argv[4] );} else { cout &lt;&lt; "\n";cout &lt;&lt; "LISSAJOUS_OPENGL:\n";cout &lt;&lt; " Please enter B1, for x(i) = sin ( ( A1 * t + B1 ) * 2 * pi ).\n";cin &gt;&gt; b1; } if ( 6 &lt;= argc ){a2 = atof ( argv[5] );} } else { cout &lt;&lt; "\n";cout &lt;&lt; "LISSAJOUS_OPENGL:\n";cout &lt;&lt; " Please enter A2, for y(i) = sin ( ( A2 * t + B2 ) * 2 * pi ).\n";cin &gt;&gt; a2; } if ( 7 &lt;= argc ){b2 = atof ( argv[6] );} else { cout &lt;&lt; "\n";cout &lt;&lt; "LISSAJOUS_OPENGL:\n";cout &lt;&lt; " Please enter B2, for y(i) = sin ( ( A2 * t + B2 ) * 2 * pi ).\n";cin &gt;&gt; b2; } } </pre>	<pre> if (2 &lt;= argc) {n = atoi (argv[1]);} else{cout &lt;&lt; "\nLISSAJOUS_OPEN_GL:\n"; cout &lt;&lt; " Ketikkan angka 1000 untuk angka dari poin-poin\n"; cin &gt;&gt; n;} if (3 &lt;= argc) {t2 = atof (argv[2]);} else{cout &lt;&lt; "\nLISSAJOUS_OPEN_GL:\n"; cout &lt;&lt; " Masukkan nilai dari T2, final nilai dari waktu.\n"; cout &lt;&lt; " Nilai yang dimasukkan adalah 1 jika A1 dan A2 adalah integer.\n"; cin &gt;&gt; t2;} if (4 &lt;= argc) {a1 = atof (argv[3]);} else{cout &lt;&lt; "\nLISSAJOUS_OPEN_GL:\n"; cout &lt;&lt; " Masukkan nilai A1, dari x(i) = sin ((A1 * t + B1) * 2 * pi).\n"; cin &gt;&gt; a1;} if (5 &lt;= argc) {b1 = atof (argv[4]);} else{cout &lt;&lt; "\nLISSAJOUS_OPEN_GL:\n"; cout &lt;&lt; " Masukkan nilai B1, dari x(i) = sin ((A1 * t + B1) * 2 * pi).\n"; cin &gt;&gt; b1;} if (6 &lt;= argc){a2 = atof (argv[5]);} else{cout &lt;&lt; "\nLISSAJOUS_OPEN_GL:\n"; cout &lt;&lt; " Masukkan nilai A2, dari y(i) = sin ((A2 * t + B2) * 2 * pi).\n"; cin &gt;&gt; a2;} if (7 &lt;= argc){b2 = atof (argv[6]);} else{cout &lt;&lt; "\nLISSAJOUS_OPEN_GL:\n"; cout &lt;&lt; " Masukkan nilai B2, dari y(i) = sin ((A2 * t + B2) * 2 * pi).\n"; cin &gt;&gt; b2;} </pre>
--------------------	---	---

Bagian 3	<pre> sprintf ( title, "Lissajous %d %5.2f %5.2f %5.2f %5.2f %5.2f", n, t2, a1, b1, a2, b2 ); </pre>	<pre> sprintf (title, "Kurva Lissajous %d %5.2f %5.2f %5.2f %5.2f %5.2f" ,n, t2, a1, b1, a2, b2); </pre>
Bagian 4	<pre> int i; int k; float r8_pi = 3.14159265358979; float t; float *xy; xy = new float[2*n]; k = 0; for ( i = 0; i &lt; n; i++ ) { t = ( float ) ( i ) * t2 / ( float ) ( n - 1 ); xy[k] = r1 * sin ( ( a1 * t + b1 ) * 2.0 * r8_pi ); xy[k+1] = r2 * sin ( ( a2 * t + b2 ) * 2.0 * r8_pi ); k = k + 2; } </pre>	<pre> int i; int j; int k; float pi = 3.14159265358979; float t; float *xy; xy = new float[2*n]; k = 0; for (i = 0; i &lt; n; i++) { t = (float)(i) * t2 / (float)(n - 1); xy[k] = r1 * sin ((a1 * t + b1)* 2.0 * pi); xy[k+1] = r2 * sin ((a2 * t + b2)* 2.0 * pi); k = k + 2; } </pre>
Bagian 5	<pre> glColor3f ( 0.0, 0.0, 1.0 ); glClearColor ( 1.0, 1.0, 1.0, 1.0 ); glPointSize ( 5.0 ); gluOrtho2D ( -1.1, 1.1, -1.1, 1.1 ); </pre>	<pre> glColor3f ( 0.0, 1.0, 0.0 ); glClearColor (0.0, 0.0, 0.0, 0.0); glPointSize ( 50.0 ); gluOrtho2D ( -1.2, 1.2, -1.2, 1.2 ); </pre>

## KESIMPULAN DAN SARAN

Kode yang diberikan dapat digunakan dalam sistem operasi windows dengan menggunakan compiler Borland C++. Sehingga dalam melakukan penggambaran kurva lissajous tidak perlu lagi menggunakan alat osiloskop untuk melakukan penggambaran kurva lissajous, cukup hanya menggunakan program ini kita melakukan inputan nilai yang kita inginkan dan terciptalah hasil dari penggambaran kurva lissajous. Program tersebut dapat dikembangkan kedepannya dengan lebih interaktif dengan adanya penambahan menu dan menambahkan penggambaran kurva lainnya.

## DAFTAR PUSTAKA

- Angel, E. (2000). *Interactive Computer Graphics, a Top-Down Approach with OpenGL*. Addison-Wesley.
- Burkardt, J. (2008). *Display a lissajous figure using opengl*. [http://people.sc.fsu.edu/~jburkardt/cpp\\_src/lissajous\\_open\\_gl/lissajous\\_open\\_gl.html](http://people.sc.fsu.edu/~jburkardt/cpp_src/lissajous_open_gl/lissajous_open_gl.html).
- Gökalp, M. S. (2018). *Multidimensional Perceptions of Physics*. *International Journal of Innovation in Science and Mathematics Education*, 26(1), 27–45.
- MacTutor History of Mathematics Archive. *Lissajous curves*. <http://www-groups.dcs.st-and.ac.uk/~history/Curves/Lissajous.html>.

Kempf, R., Frazier C. (2004).  
*OpenGL Reference Manual*. Addison-

Wesley, 4th edition