Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

Pembuatan Animasi Integral Sebagai Implementasi Grafika Komputer Dengan Menggunakan Morphing dan OpenGL

Herriyance[1], Chalil Al Vareel[2], Ewaldo[3], Rio Francisco Hutabarat[4], Sastra Harapan Gulo[5]

Jurusan Ilmu Komputer Fasilkom-TI Universitas Sumatera Utara
E-mail : [1]Herriyance@usu.ac.id, [2]chalilalvareel@gmail.com, [3]ewaldo296@gmail.com, [4]riofrancisco29082002@gmail.com, [5]sastraharapangulo2002@gmail.com
Email Penulis Korespondensi: chalilalvareel@gmail.com

Article History:
Received Jun 22th, 2022
Revised Jul 12th, 2022
Accepted Jul 27th, 2022

Abstrak

Penerapan grafika komputer dalam kehidupan sehari-hari sudah sering kita temukan. Salah satu contoh penerapan dari grafika komputer ini adalah penerapan teknik morphing. Morphing adalah proses transisi dari suatu bentuk objek ke bentuk objek yang lainnya. Salah satu algoritma yang dapat digunakan untuk membuat teknik morphing adalah dengan metode interpolasi linear. Tujuan kami melakukan penelitian ini adalah untuk melihat bagaimana penerapan algoritma morphing metode interpolasi linear dalam pembuatan animasi integral. Metode yang digunakan pada penelitian kali ini adalah dengan mengimplementasikan algoritma morphing metode interpolasi linear pada OpenGL dan melakukan simulasi dalam bahasa pemrograman C++ yang nantinya akan menghasilkan animasi integral.

Kata Kunci : Algoritma, Interpolasi Linear, Morphing, OpenGL, Pemrograman C++

Abstract

We often find the application of computer graphics in everyday life. One example of the application of computer graphics is the application of morphing techniques. Morphing is the process of transitioning from one object form to another. One of the algorithms that can be used to create a morphing technique is the linear interpolation method. The purpose of this research is to see how the application of the linear interpolation method morphing algorithm in making integral animations. The method used in this research is to implement the morphing algorithm linear interpolation method in OpenGL and perform simulations in the C++ programming language which will later produce integral animations.

Keyword : *Algorithm, Linear Interpolation, Morphing, OpenGL, C++ Programming*

1. PENDAHULUAN

Dewasa ini teknologi semakin maju dan berkembang semakin pesat, bahkan hampir semua bidang pekerjaan masyarakat tidak dapat lepas dari keberadaan teknologi. Salah satu teknologi yang saat ini banyak dimanfaatkan masyarakat untuk menunjang pekerjaannya adalah grafika komputer. Bidang yang paling banyak menggunakan grafika komputer adalah bidang yang langsung berinteraksi dengan manusia dan yang bersifat visual.

Grafika komputer sendiri adalah suatu bidang ilmu komputer dan matematika untuk mempresentasikan dan memanipulasi data gambar menggunakan komputer. Pembuatan animasi integral merupakan implementasi dari grafika komputer. Animasi integral ini dapat dibuat dengan menggunakan algoritma *morphing* dan metode interpolasi linear yang dijalankan dengan bahasa pemrograman C++ dan *library* OpenGL.

Mengenai penelitian terdahulu, kami tidak dapat menemukan penelitian lainnya yang serupa, namun kami menggunakan penelitian terdahulu ini sebagai referensi untuk menambah wawasan mengenai teori-teori yang akan dipaparkan pada penelitian kami.

Penelitian terdahulu pertama yang terkait dengan penelitian ini adalah sebuah penelitian yang dilakukan oleh María-Jesús dkk. pada tahun 2018. Pada penelitian terdahulu ini, terdapat penerapan *morphing* yang salah satunya adalah *image*

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

morphing untuk gambar-gambar satelit dengan memanfaatkan interpolasi linear [1]. Walaupun kami juga menggunakan metode interpolasi linear, tetapi penelitian kami bertumpu kepada *morphing* objek 2 dimensi.

Penelitian terdahulu kedua adalah penelitian yang dilakukan oleh James L. Jones pada tahun 2018 mengenai implementasi *morphing* pada objek 3 dimensi, yaitu sebuah wajah dengan ekspresi yang dapat berubah [2]. Sedangkan pada penelitian kami, kami menerapkan *morphing* hanya sebatas untuk objek 2 dimensi saja.

Penelitian terdahulu ketiga adalah penelitian oleh S. I. Vyatkin dkk. yang dilakukan pada tahun 2018. Penerapan *morphing* pada penelitian ini dikaitkan erat dengan teori perturbasi pada Fisika dan menggunakan objek 3 dimensi sebagai objek sasaran. Ada pula penggunaan interpolasi linear dalam menghitung nilai fungsi perturbasi [3].

Penelitian terdahulu keempat adalah penelitian mengenai animasi 2 dimensi yang dilakukan oleh Welly Desriyati pada tahun 2021. Pada penelitian ini, animasi 2 dimensi berfungsi sebagai media pembelajaran matematika dengan materi pengenalan bangun datar. Di samping itu, animasi dibuat dalam bentuk video [4]. Untuk penelitian kami, kami merancang animasi 2 dimensi dalam bentuk program, serta materi yang berkaitan adalah materi integral.

Penelitian terdahulu kelima adalah penelitian mengenai *morphing* pada teks yang dilakukan oleh Shaohan Huang dkk. pada tahun 2018. Meskipun penelitian kami mengimplementasikan *morphing* bukan pada teks, melainkan pada objek 2 dimensi saja, namun kedua penelitian menggunakan interpolasi linear. Yang membedakan adalah pada penelitian terdahulu ini, algoritma interpolasi linear menerima kalimat sebagai masukan [5], sedangkan interpolasi linear pada penelitian kami menerima koordinat titik pada objek sebagai masukan.

Dengan memperhatikan beberapa hal tersebut, kami membuat penelitian yang berjudul **"**Pembuatan Animasi Integral Sebagai Implementasi Grafika Komputer Dengan Menggunakan Morphing dan OpenGL**"**. Tujuan penulisan ini adalah untuk memahami bagaimana pembuatan animasi integral dengan menggunakan teori morphing sebagai implementasi dari grafika komputer. Kami berharap bahwa penelitian kami dapat menjadi inspirasi bagi penelitian-penelitian selanjutnya, dalam hal penerapan algoritma ataupun lainnya.

2. METODOLOGI PENELITIAN

2.1 Landasan Teori

a. Komputer Grafik
Grafika komputer adalah suatu bidang ilmu komputer yang berkaitan dengan pembuatan dan manipulasi gambar (visual) secara digital [6]. Bentuk sederhana dari grafika komputer adalah grafika komputer 2D yang selanjutnya dikembangkan menjadi grafika komputer 3D, pemrosesan citra (*image processing*), dan pengenalan pola (*pattern recognition*).

b. *Morphing*
Morphing adalah suatu efek khusus dimana akan melakukan transisi dari suatu bentuk objek ke bentuk objek yang lainnya [7]. Teknik *morphing* telah digunakan dalam pembuatan animasi, film, serta telah diimplementasikan di berbagai bidang seperti *video filter*.

c. OpenGL
OpenGL (*Open Graphics Library*) adalah sebuah API yang digunakan untuk pembuatan atau penerapan grafis dua dimensi (2D) maupun tiga dimensi (3D). Sifatnya yang *cross-platform* membuat OpenGL dapat digunakan di berbagai sistem operasi yang ada. OpenGL sendiri telah menjadi perbincangan di tengah kalangan-kalangan pengguna yang melakukan implementasi pemrograman grafik 2D dan 3D [8].

d. Algoritma Interpolasi linear
Algoritma interpolasi linear adalah algoritma *morphing* yang di mana terjadi proses interpolasi. Interpolasi itu sendiri adalah penyisipan, sehingga dalam algoritma ini, titik ditempatkan di antara dua *endpoint* (titik awal dan titik akhir) [7, 10]. Ada 2 rumus interpolasi linear yang umum dipakai, yang satu adalah untuk posisi titik dalam satu garis lurus dan satu lagi adalah untuk proses *morphing*.

Rumus interpolasi linear untuk pencarian posisi titik dalam garis lurus [10] adalah sebagai berikut :

$$□ = □1 + (□ - □1) * \frac{□2 - □1}{□2 - □1} \quad (1)$$

Di mana :

x1, y1 : Posisi titik di awal
x2, y2 : Posisi titik di akhir
x, y : Posisi titik pada garis lurus

Sedangkan rumus interpolasi linear untuk proses *morphing* [7] adalah sebagai berikut :

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

$□_i(□) = (1 - □)□_1 + (□)□_2$, dengan 0 < t < 1 (2)

Di mana :

$V_i(t)$: Posisi titik di antara 2 *endpoint* pada waktu ke-t
V_1 : Posisi titik di awal
V_2 : Posisi titik di akhir
t : Waktu

Tahap-tahap dalam interpolasi linear :

1. Tentukan posisi titik di awal dan di akhir
2. Tentukan waktu
3. Hitung posisi titik di antara 2 *endpoint*

Algoritma interpolasi linear sendiri telah diterapkan di berbagai bidang, seperti estimasi fungsi ambiguitas pada *wideband* yang penelitiannya dilakukan oleh Jia-Jia Jiang pada tahun 2018 [9] dan substitusi nilai yang hilang pada suatu himpunan data, yang di mana penelitiannya dilakukan oleh Khongorzul Dashdondov dkk. pada tahun 2022 [10].

e. GLUT
GLUT merupakan hasil pengembangan dari OpenGL yang diciptakan untuk aplikasi dengan level kecil hingga menengah. Selain itu, GLUT menggunakan fungsi *callback* untuk menambahkan interaksi dari user. Salah satu tujuan GLUT ialah menciptakan fleksibilitas kode platform yang dapat dijalankan lebih dari satu sistem operasi (*cross-platform*). Dengan menggunakan GLUT, *programmer* hanya memerlukan sedikit code tanpa mengetahui spesifikasi operasi [6].

f. Animasi
Animasi adalah kumpulan berbagai gambar yang diproses dengan sedemikian rupa sehingga memberikan hasil akhir berupa gerakan [11]. Animasi komputer dibagi dalam dua jenis yaitu animasi dua dimensi dan animasi tiga dimensi [12].

g. Integral
Integral adalah salah satu topik dalam matematika dan merupakan konsep anti turunan diferensial. Penerapan yang paling mendasar adalah perhitungan luas daerah yang bentuknya tidak beraturan [13].

2.2 Tahapan Penelitian

Penelitian dilakukan melalui 4 tahap, yaitu tahap perencanaan (*planning*), pengkodean (*coding*), tahap pengujian (*testing*), dan tahap evaluasi (*debugging*).

a. *Planning*
Pada tahap ini, model program direncanakan terlebih dahulu, mulai dari alur kerja program hingga desain visual program.

b. *Coding*
Pada tahap ini dilakukan proses pembuatan program diawali dengan mengetikkan kode program. Tujuannya adalah untuk mengimplementasikan model program yang telah direncanakan sebelumnya.

c. *Testing*
Pada tahap ini dilakukan pengujian terhadap program yang telah dibuat agar dapat dilihat apakah program telah berjalan dengan semestinya. Pada tahap ini juga, dapat dilakukan pelacakan kesalahan kode yang akan diperbaiki.

d. *Debugging*
Pada tahap ini apabila terdapat kesalahan-kesalahan pada program, maka dilakukan analisis terhadap *error* yang ada. Perbaikan ditujukan agar program dapat beroperasi sesuai dengan harapan.

3. HASIL DAN PEMBAHASAN

3.1 Kode program

Pseudocode untuk algoritma interpolasi linear pada program ini adalah sebagai berikut :
Judul : Program Interpolasi Linear

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

Deklarasi
var V1, V2, t, Vt : float;

Implementasi
read(V1);
read(V2);
read(t);

Vt ← (1 - t) * V1 + (t) * V2;
write(Vt);

Setelah program secara keseluruhan telah direncanakan, maka penelitian dapat dimulai dengan mengetikkan kode program sebagai implementasi perencanaan sebelumnya.

Berikut ini adalah potongan-potongan kode program yang penting :

```
#include <glut.h>
#include <math.h>
#include <vector>
using namespace std;

vector<float> CurveCoordinatesY = {};

float t = 0.5;
float ylb1 = 500, yrb1 = 500,
      ylb2 = 500, yrb2 = 500,
      ylb3 = 500, yrb3 = 500,
      ylb4 = 500, yrb4 = 500,
      ylb5 = 500, yrb5 = 500,
      ylb6 = 500, yrb6 = 500,
      ylb7 = 500, yrb7 = 500,
      ysamadenganfx    = 500;

const double pi = 3.14159265358979323846;
```

Gambar 1. Inisialisasi Library dan Variabel

Gambar 1 di atas adalah potongan kode untuk menggunakan *libraries* serta mendefinisikan variabel-variabel yang akan digunakan. *Library* yang digunakan adalah *library* bawaan C++ dan *library* OpenGL. *Library* "math.h" dipakai untuk perhitungan-perhitungan matematis, sedangkan *library* "vector" berfungsi agar *vector* dapat digunakan dalam program ini.

```
float Morphing(float V1, float V2, float t)
{
    float Vt;

    Vt = (1 - t) * V1 + (t) * V2;

    return Vt;
}
```

Gambar 2. Fungsi Morphing

Gambar 2 di atas adalah potongan kode untuk menerapkan rumus interpolasi linear yang telah disesuaikan dengan *pseudocode* dan persamaan (2), dengan parameter-parameter nya adalah V1, V2, dan t yang masing-masing bertipe data float. Hasil dari perhitungan dimasukkan ke dalam variabel Vt yang nilainya akan dikembalikan saat fungsi dipanggil.

```
void NilaiMulaMula()
{
    for (int i = 200; i <= 800; i++) {
        CurveCoordinatesY.push_back(500);
    }
}
```

Gambar 3. Fungsi Memasukkan Nilai ke Vector

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

Gambar 3 di atas adalah potongan kode untuk memasukkan nilai koordinat Y setiap titik penyusun kurva. Perlu diketahui bahwa pada program ini, kurva tersusun atas sekumpulan titik sehingga proses perubahan bentuk kurva dapat dilakukan.

```
glBegin(GL_POINTS);

for (int i = 200; i <= 800; i++) {
    glVertex2f(i, CurveCoordinatesY[i - 200]);
}

glEnd();
```

Gambar 4. Pembentukan Kurva

Gambar 4 di atas adalah potongan kode untuk menggambar kurva. Perulangan dieksekusi agar setiap titik dapat ditempatkan dan secara keseluruhan membentuk kurva yang mula-mula merupakan garis lurus.

```
void PersegiPanjang()
{
    glColor3ub(255, 0, 0);

    glBegin(GL_QUADS);

    glVertex2f(290, 100); glVertex2f(290, ylb1); glVertex2f(350, yrb1); glVertex2f(350, 100);
    glVertex2f(350, 100); glVertex2f(350, ylb2); glVertex2f(410, yrb2); glVertex2f(410, 100);
    glVertex2f(410, 100); glVertex2f(410, ylb3); glVertex2f(470, yrb3); glVertex2f(470, 100);
    glVertex2f(470, 100); glVertex2f(470, ylb4); glVertex2f(530, yrb4); glVertex2f(530, 100);
    glVertex2f(530, 100); glVertex2f(530, ylb5); glVertex2f(590, yrb5); glVertex2f(590, 100);
    glVertex2f(590, 100); glVertex2f(590, ylb6); glVertex2f(650, yrb6); glVertex2f(650, 100);
    glVertex2f(650, 100); glVertex2f(650, ylb7); glVertex2f(710, yrb7); glVertex2f(710, 100);

    glEnd();

    glColor3ub(139, 0, 0);

    glBegin(GL_LINE_LOOP);

    glVertex2f(290, 100); glVertex2f(290, ylb1); glVertex2f(350, yrb1); glVertex2f(350, 100);
    glVertex2f(350, 100); glVertex2f(350, ylb2); glVertex2f(410, yrb2); glVertex2f(410, 100);
    glVertex2f(410, 100); glVertex2f(410, ylb3); glVertex2f(470, yrb3); glVertex2f(470, 100);
    glVertex2f(470, 100); glVertex2f(470, ylb4); glVertex2f(530, yrb4); glVertex2f(530, 100);
    glVertex2f(530, 100); glVertex2f(530, ylb5); glVertex2f(590, yrb5); glVertex2f(590, 100);
    glVertex2f(590, 100); glVertex2f(590, ylb6); glVertex2f(650, yrb6); glVertex2f(650, 100);
    glVertex2f(650, 100); glVertex2f(650, ylb7); glVertex2f(710, yrb7); glVertex2f(710, 100);

    glEnd();
}
```

Gambar 5. Potongan Kode Persegi Panjang

Gambar 5 di atas adalah potongan kode untuk membentuk 7 persegi panjang, dengan masing-masing persegi panjang dikaitkan dengan variabel. Tujuannya adalah agar semua ukuran persegi panjang tersebut dapat berubah bentuk pada saat pengaturan fase *morphing*.

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

```
void specialKey(int key, int x, int y) // Mengatur fase morphing
{
    switch (key)
    {
    case GLUT_KEY_UP:
        if (t < 1) {
            t += 0.1;
        }
        break;
    case GLUT_KEY_DOWN:
        if (t > 0) {
            t -= 0.1;
        }
        break;
    }

    // Untuk kurva

    for (int i = 200; i <= 800; i++) {
        CurveCoordinatesY[i - 200]
            = Morphing(500 - 200 * sin(pi * (i - 290) / 420), 500 + 200 * sin(pi * (i - 290) / 420), t);
    }

    ysamadenganfx = Morphing(500 - 200 * sin(pi * (510) / 420), 500 + 200 * sin(pi * (510) / 420), t);

    // Untuk persegi panjang

    ylb1 = Morphing(440, 560, t); ylb2 = Morphing(380, 620, t); ylb3 = Morphing(340, 660, t); ylb4 = Morphing(300, 700, t);
    ylb5 = Morphing(340, 660, t); ylb6 = Morphing(380, 620, t); ylb7 = Morphing(440, 560, t);

    yrb1 = Morphing(440, 560, t); yrb2 = Morphing(380, 620, t); yrb3 = Morphing(340, 660, t); yrb4 = Morphing(300, 700, t);
    yrb5 = Morphing(340, 660, t); yrb6 = Morphing(380, 620, t); yrb7 = Morphing(440, 560, t);

    glutPostRedisplay();
}
```

Gambar 6. Potongan Kode Pertama Pengatur Fase Morphing

Gambar 6 di atas adalah potongan kode untuk mengatur proses perubahan bentuk kurva dan semua persegi panjang. Diperlihatkan bahwa proses dapat dimajukan dengan cara menekan tombol panah atas pada papan ketik, dan begitu pula sebaliknya.

```
static void display(void)
{
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPointSize(1.0);
    PersegiPanjang();
    Sumbu();
    Kurva();
    Pembatas();
    glutSwapBuffers();
}
```

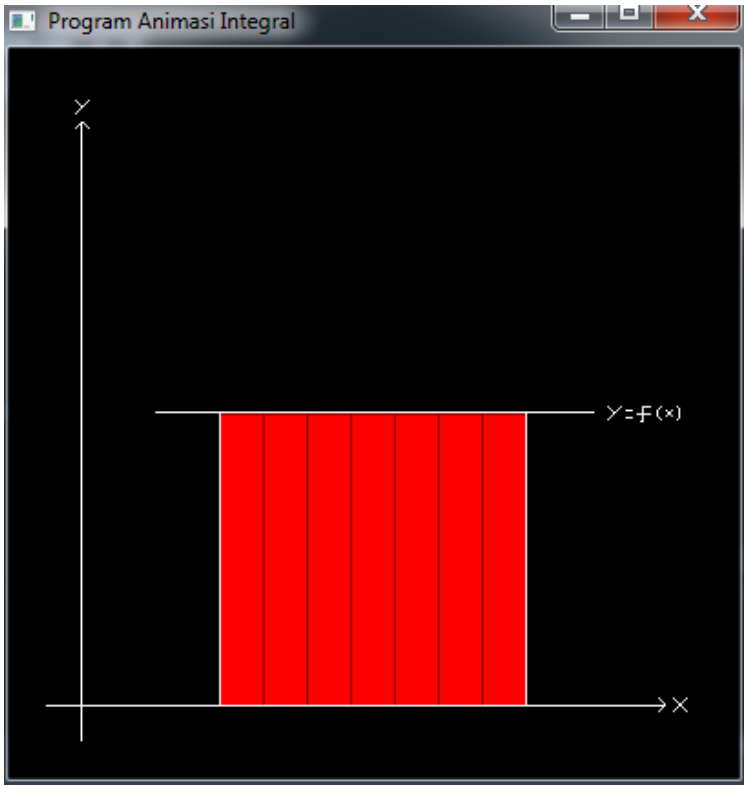
Gambar 7. Fungsi Display

Gambar 7 di atas adalah potongan kode untuk memanggil seluruh fungsi yang diperlukan agar objek-objek tampil di layar. Fungsi pertukaran *buffer* dipanggil karena menggunakan mode *display* GLUT_DOUBLE.

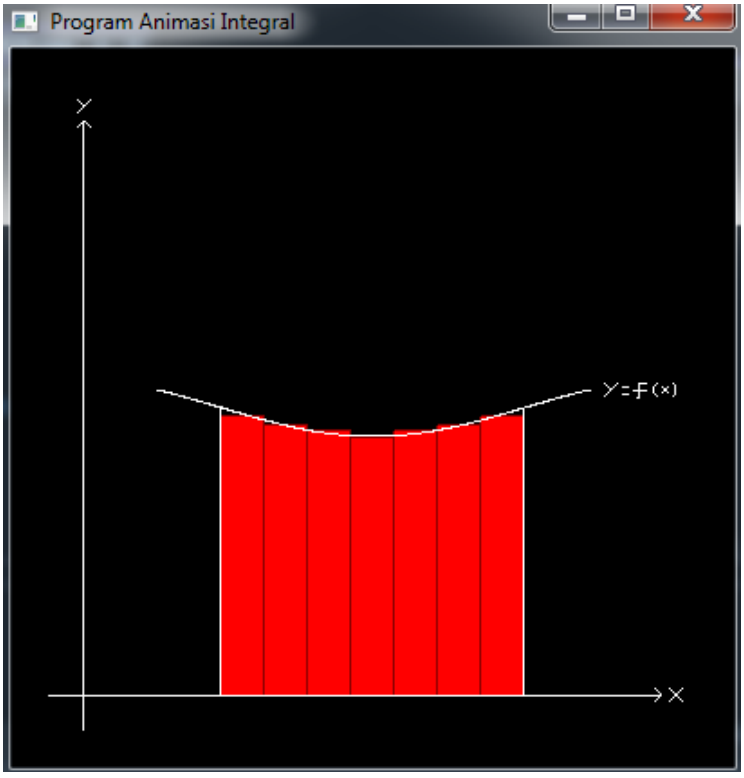
3.2 Output

Berikut ini adalah hasil program saat dijalankan :

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

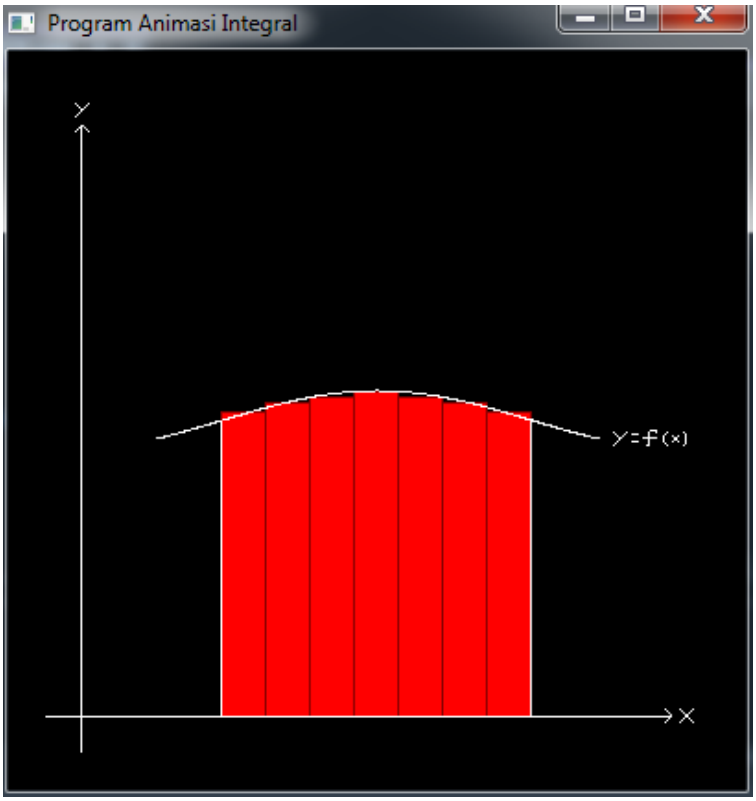


Gambar 8. Bentuk Grafik Mula-Mula (t = 0,5)

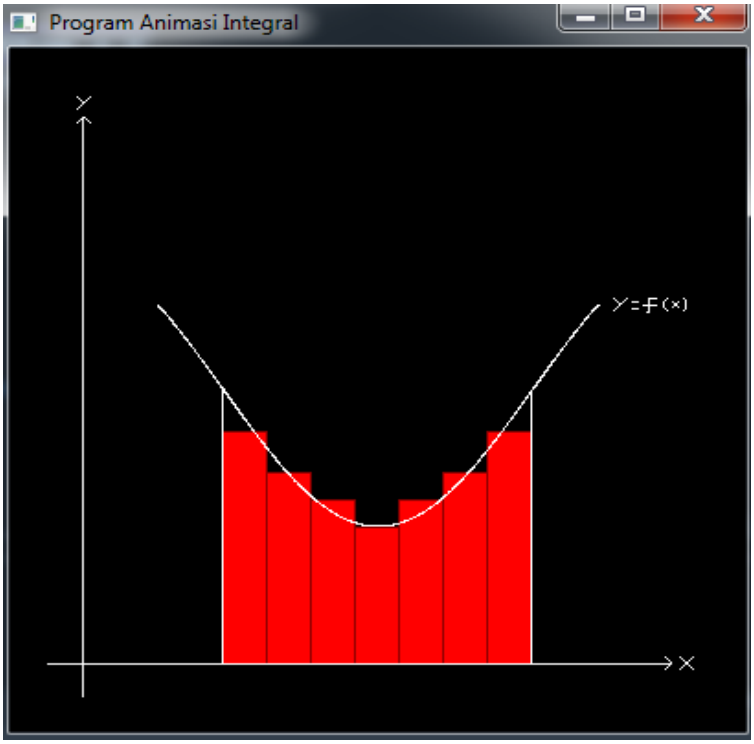


Gambar 9. Bentuk Grafik Cekung Ke Atas (t = 0.4)

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

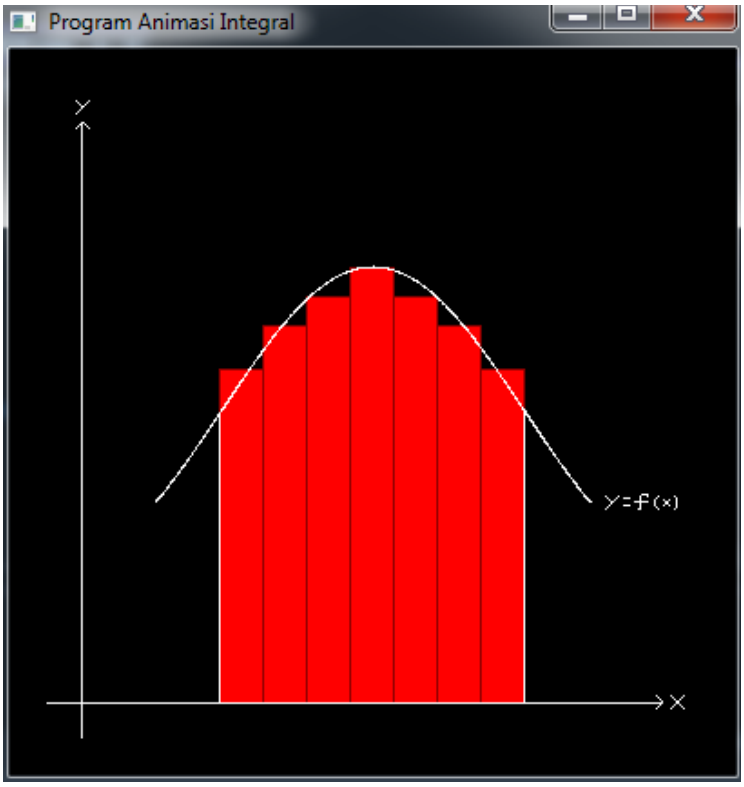


Gambar 10. Bentuk Grafik Cekung Ke Bawah (t = 0,6)



Gambar 11. Bentuk Grafik Pada Fase Minimum (t = 0,0)

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index



Gambar 12. Bentuk Grafik Pada Fase Maksimum (t = 1,0)

Gambar 8 hingga 12 di atas adalah hasil jalannya program. Pada gambar 8, bentuk kurva hanyalah sebuah garis lurus pada saat t bernilai 0,5. Gambar 9 dan 10 adalah pada saat pengguna menekan tombol panah atas atau panah bawah pada papan ketik, sehingga bentuk kurva akan menjadi cekung ke atas atau ke bawah tergantung dari tombol apa yang ditekan. Gambar 11 dan 12 adalah pada saat fase perubahan berada pada nilai ekstrim, yaitu 0,0 dan 1,0 untuk masing-masing gambar.

3.3 Perhitungan manual

Pada program, terdapat 8 objek yang berubah bentuk, yaitu 7 buah persegi panjang dan 1 buah kurva. Untuk perhitungan manual ini, diambil satu persegi panjang dengan titik-titiknya adalah A(470, 100), B(470, 300), C(530, 300), dan D(530, 100). Persegi panjang tersebut akan berubah bentuk sehingga titik-titik akhirnya adalah A'(470, 100), B'(470, 700), C'(530, 700), dan D'(530, 100). Tampak bahwa titik-titik yang berubah posisi adalah titik B dan C. Oleh karena itu, perhitungan interpolasi linear secara manual untuk kedua titik itu dijabarkan sebagai berikut :

1. Tentukan posisi titik di awal dan di akhir
   Posisi titik-titik di awal adalah B(470, 300) dan C(530, 300)
   Posisi titik-titik di akhir adalah B'(470, 700) dan C(530, 700)

2. Tentukan waktu
   Waktu dimulai dari 0 hingga 1, dengan penambahan waktu adalah 0,1 setelah selesai perhitungan

3. Hitung posisi titik di antara 2 *endpoint*
   Perhitungan untuk kedua titik dijabarkan dalam bentuk tabel seperti di bawah ini :

Tabel 1. Perhitungan Manual Interpolasi Linear (Titik B Menuju B')

t	x(t)	y(t)
0,0	470	300
0,1	470	340

Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
Volume 5 ; Nomor 2 ; Juli 2022 ; Page 269-279
E-ISSN : 2615-5133 ; P-ISSN : 2621-8976
https://ojs.trigunadharma.ac.id/index.php/jsk/index

0,2	470	380
0,3	470	420
0,4	470	460
0,5	470	500
0,6	470	540
0,7	470	580
0,8	470	620
0,9	470	660
1,0	470	700

Tabel 2. Perhitungan manual interpolasi linear (titik C menuju C')

t	x(t)	y(t)
0,0	530	300
0,1	530	340
0,2	530	380
0,3	530	420
0,4	530	460
0,5	530	500
0,6	530	540
0,7	530	580
0,8	530	620
0,9	530	660
1,0	530	700

Keterangan tabel :

1. Perhitungan terjadi setiap **0,1** detik, maka banyaknya perhitungan dalam proses *morphing* ini adalah 1/0,1 = 10 kali
2. **x(t)** dan **y(t)** adalah koordinat titik terhadap variabel t selama proses *morphing* terjadi
3. Koordinat selanjutnya ditentukan oleh persamaan (2)
4. Perhitungan pada poin sebelumnya dilakukan terus-menerus hingga $\mathbf{x = x_2}$ dan $\mathbf{y = y_2}$, serta **t = 1,0**

4. KESIMPULAN

Berdasarkan pembahasan diatas dapat disimpulkan bahwa pengimplementasian salah satu algoritma pada grafika komputer yaitu algoritma *morphing* metode interpolasi linear dapat diimplementasikan dalam pembuatan animasi integral dengan menggunakan OpenGL dan bahasa pemrograman C++.

UCAPAN TERIMA KASIH

Ucapan terima kasih kami sampaikan kepada Tuhan Yang Maha Esa atas karunia-Nya kami dapat menyelesaikan penelitian Pembuatan Animasi Integral Sebagai Implementasi Grafika Komputer Dengan Menggunakan Morphing dan OpenGL dengan tepat waktu. Selain itu, ucapan terima kasih juga kami sampaikan kepada pak Herriyance, S.T., M.Kom selaku dosen mata kuliah Komputer Grafik dan Visualisasi atas bantuan dan bimbingannya selama penyusunan jurnal ini berlangsung.

DAFTAR PUSTAKA

[1] M. Lobo, C. Appert and E. Pietriga, "Animation Plans for Before-and-After Satellite Images," in IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 2, pp. 1347-1360, 1 Feb. 2019, doi: 10.1109/TVCG.2018.2796557.

[2] Jones, James L. "Efficient Morph Target Animation Using OpenGL ES 3.0" *in GPU Pro 360 Guide to Mobile Devices*, First Edition, New York : CRC Press, 2018.

[3] S. I. Vyatkin, A. N. Romanyuk, L. A. Savytska, T. I. Troianovska, and N. V. Dobrovolska, "Real-Time Deformations of Function-Based Surfaces using Perturbation Functions," Journal of Physics: Conference Series, vol. 1015, no. 3, pp. 1–6, 2022, Accessed: Jun. 24, 2022. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1015/3/032115/meta

[4] Welly Desriyati, "Video Animasi 2D Pengenalan Bangun Datar Pada Pembelajaran Matematika ", PIXEL, vol. 14, no. 2, pp. 189-195, Dec. 2021.

[5] S. Huang, Y. Wu, F. Wei, and M. Zhou, *Text Morphing*, vol. xx, no. xx, pp. 1–12, Sep. 2018, Accessed: Jun. 24, 2022. [Online]. Available: https://arxiv.org/abs/1810.00341

[6] Muhammad Adnani and Achmad Zakki Falani, "Implementasi Open Gl Untuk Pembuatan Objek 3d: Implementasi Open Gl Untuk Pembuatan Objek 3d", ZTR, vol. 3, no. 1, pp. 1-6, Mar. 2021.

[7] T. A. Wibowo, R. R. Isnanto, and A. Hidayatno, "TEKNIK MORPHING UNTUK OBJEK CITRA TIGA DIMENSI MENGGUNAKAN METODE INTERPOLASI LINEAR," Transient: Jurnal Ilmiah Teknik Elektro, vol. 1, no. 3, pp. 68-71, Sep. 2012. https://doi.org/10.14710/transient.1.3.68-71.

[8] J. Widadi and M. Murinto, 'Media Pembelajaran Materi Pengenalan Opengl Pada Mata Kuliah Grafika Komputer', Jurnal Sarjana Teknik Informatika, vol. 6, no. 1, pp. 47–53, 2019.

[9] J. Jiang *et al.*, "An efficient algorithm for WBAF estimation based on linear interpolation and its estimation error," *Applied Acoustics*, vol. 142, pp. 44–52, Dec. 2018, doi: 10.1016/j.apacoust.2018.08.001.

[10] K. Dashdondov, K. Jo, and M.-H. Kim, "Linear interpolation and Machine Learning Methods for Gas Leakage Prediction Base on Multi-source Data Integration," Journal of the Korea Convergence Society, vol. 13, no. 3, pp. 33–41, Mar. 2022.

[11] A. Sutanto, E. Eradaru, and J. Cahyadi, "Perancangan Animasi Tentang Teknik Dan Manfaat Meditasi Pernapasan," vol. 1, no. 16, pp. 1–10, 2020, Accessed: Jun. 24, 2022. [Online]. Available: https://publication.petra.ac.id/index.php/dkv/article/view/10220/9148.

[12] Wulandari, Elin. "*Implementasi Animasi 2 Dimensi Layanan Aspirasi Dan Pengaduan Online Rakyat (Lapor) Di Dinas Komunikasi Dan Informatika Kota Palembang," M.S. thesis,* Computer Engineering, State Polytechnic of Sriwijaya, Palembang, PLM, 2021.

[13] P. Pardimin, "Pengembangan YouTube Pembelajaran Aplikasi Kalkulus Integral pada Geometri", wacana akademika, vol. 3, no. 1, pp. 47–60, Apr. 2019.