

Tugas MK. IFB-201 Grafika Komputer Terapan TA 2024-2025

Tugas Besar Transformasi Objek 2D Bertemakan Pemandangan Pantai

Disusun oleh

Kelompok– 11

15-2023-003	-	Muhammad Rifqi Yusufi
15-2023-020	-	Muhamad Rizky

Tgl. Penugasan : 26-11-2024

Tgl. Penyerahan : xx-01-2025



Prodi Informatika
Fakultas Teknologi Industri
Institut Teknologi Nasional
Bandung
2025

Kata Pengantar

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulisan laporan yang berjudul "Transformasi Objek 2D Bertemakan Pemandangan Pantai" ini dapat diselesaikan dengan baik.

Laporan ini disusun sebagai bentuk kontribusi dalam mengembangkan pemahaman mengenai teknik transformasi objek dua dimensi (2D) dalam seni dan desain grafis, khususnya dengan menggunakan tema pemandangan pantai yang kaya akan elemen visual dan estetika. Melalui laporan ini, diharapkan pembaca dapat memperoleh wawasan tentang penerapan teknik transformasi 2D serta inspirasi untuk menciptakan karya-karya kreatif yang inovatif.

Penulisan ini tidak lepas dari bantuan, dukungan, dan saran dari berbagai pihak. Oleh karena itu, kami ingin menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam proses penyelesaian laporan ini. Kami menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan demi penyempurnaan karya ini di masa mendatang. Semoga laporan ini bermanfaat bagi pembaca dan dapat memberikan kontribusi positif dalam bidang seni, desain grafis, dan pendidikan.

Daftar Isi

Kata Pengantar	ii
BAB I.....	1
Pendahuluan	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	1
1.3. Tujuan	1
BAB II	3
Landasan Teori	3
2.1. Software yang Digunakan	3
2.2. Bahasa Pemrograman yang digunakan	4
BAB III.....	6
Pembahasan.....	6
3.1. Sebelum Transformasi diimplementasikan	6
3.2. Setelah Transformasi diimplementasikan	12
BAB IV	15
Penutup	15
4.1. Kesimpulan	15
Daftar Pustaka	16

BAB I

Pendahuluan

1.1. Latar Belakang

Dalam dunia seni dan desain grafis, transformasi objek dua dimensi (2D) menjadi salah satu teknik yang sangat penting untuk menciptakan karya visual yang menarik dan dinamis. Transformasi melibatkan manipulasi objek melalui translasi, rotasi, refleksi, skala, dan shearing. Teknik ini tidak hanya membantu dalam mengubah tampilan visual, tetapi juga memberikan dimensi baru pada sebuah karya yang memungkinkan kreativitas berkembang lebih luas.

Pemandangan pantai sebagai tema visual menawarkan keindahan alam yang memikat, seperti hamparan pasir, laut yang luas, pohon kelapa, serta matahari terbenam yang memancarkan gradasi warna indah. Dengan menggunakan transformasi objek 2D, elemen-elemen ini dapat diolah menjadi karya yang lebih menarik, baik dalam bentuk ilustrasi statis maupun animasi.

Selain itu, penggunaan transformasi objek 2D bertemakan pemandangan pantai dapat dimanfaatkan dalam berbagai aplikasi, seperti desain poster, animasi video, media pembelajaran, hingga aplikasi interaktif. Kombinasi antara teknik transformasi dan tema pemandangan pantai juga memberikan kesempatan untuk mengeksplorasi aspek estetika dan teknologi secara bersamaan.

Penelitian dan eksplorasi ini menjadi penting untuk memberikan wawasan lebih mendalam tentang bagaimana transformasi objek 2D dapat digunakan untuk menciptakan karya seni yang tidak hanya estetis, tetapi juga komunikatif dan aplikatif. Oleh karena itu, laporan mengenai transformasi objek 2D dengan tema pemandangan pantai diharapkan dapat membuka peluang inovasi dalam bidang seni, desain, dan pendidikan visual.

1.2. Rumusan Masalah

- Bagaimana penerapan teknik transformasi objek 2D (translasi, rotasi, refleksi, skala, dan shearing) pada elemen-elemen pemandangan pantai?
- Bagaimana transformasi objek 2D dapat digunakan untuk meningkatkan nilai estetika dan daya tarik visual dari karya bertemakan pemandangan pantai?
- Apa saja tantangan yang dihadapi dalam proses transformasi objek 2D pada tema pemandangan pantai, dan bagaimana solusi yang dapat diimplementasikan?
- Bagaimana hasil transformasi objek 2D bertemakan pemandangan pantai dapat diaplikasikan dalam berbagai media kreatif, seperti desain grafis, animasi, atau media pembelajaran?

1.3. Tujuan

- Mengidentifikasi dan memahami penerapan teknik transformasi objek 2D (translasi, rotasi, refleksi, skala, dan shearing) pada elemen-elemen visual bertema pemandangan pantai.

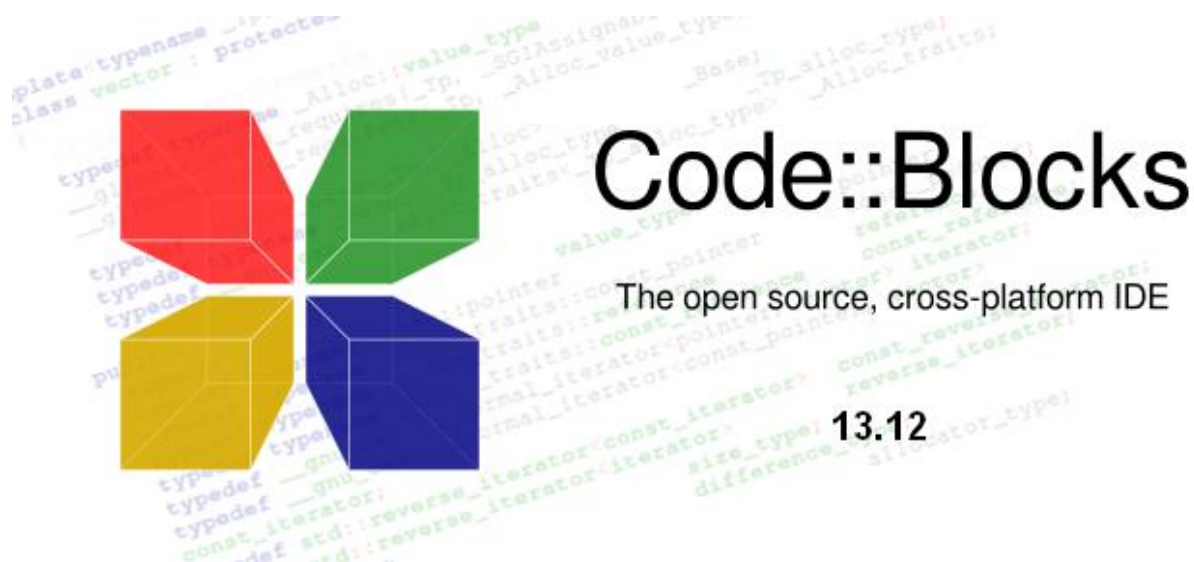
- Mengeksplorasi cara penggunaan transformasi objek 2D untuk meningkatkan nilai estetika dan daya tarik visual karya bertemakan pemandangan pantai.
- Mengatasi tantangan yang muncul dalam proses transformasi objek 2D pada tema pemandangan pantai dengan merumuskan solusi yang efektif.
- Menghasilkan karya berbasis transformasi objek 2D bertema pemandangan pantai yang dapat diaplikasikan dalam berbagai media kreatif, seperti desain grafis, animasi, atau media pembelajaran.

BAB II

Landasan Teori

2.1. Software yang Digunakan

- Code Blocks



Gambar 2.1 Logo aplikasi Code Blocks

Code Blocks adalah sebuah Integrated Development Environment (IDE) yang dirancang untuk mendukung pengembangan perangkat lunak dengan bahasa pemrograman seperti C, C++, dan Fortran. IDE ini bersifat open-source, sehingga bebas digunakan dan dimodifikasi. Dengan antarmuka yang sederhana namun lengkap, Code::Blocks sering menjadi pilihan untuk pengembang perangkat lunak, baik pemula maupun profesional.

- Fungsi Utama

Code Blocks berfungsi sebagai alat bantu dalam menulis, menguji, dan menjalankan kode program. IDE ini menyediakan berbagai fitur yang mendukung proses pengembangan perangkat lunak, seperti:

- ❖ **Editor teks canggih:** Membantu menulis kode dengan lebih cepat dan terstruktur.
- ❖ **Kompilasi dan debugging:** Mempermudah proses penerjemahan kode ke dalam format yang dapat dijalankan oleh komputer serta melacak kesalahan dalam program.
- ❖ **Pengelolaan proyek:** Membantu pengaturan file dan direktori dalam pengembangan perangkat lunak skala kecil hingga menengah.

- **Fitur-Fitur Utama**
 - ❖ **Editor yang Intuitif:** Mendukung fitur seperti syntax highlighting (penyorotan sintaks), auto-completion (penyelesaian otomatis), dan template kode.
 - ❖ **Dukungan untuk Banyak Compiler:** Code Blocks kompatibel dengan compiler seperti GCC (MinGW), Clang, dan Microsoft Visual C++.
 - ❖ **Debugger Terintegrasi:** Memiliki debugger visual yang mendukung fitur seperti breakpoint, watch variable, dan analisis call stack.
 - ❖ **Sistem Plugin:** Mendukung penambahan fungsionalitas baru melalui plugin, misalnya untuk integrasi alat analisis atau manajemen proyek.

2.2. Bahasa Pemrograman yang digunakan

- C++



Gambar 2.2 Logo C++

C++ adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Bjarne Stroustrup pada awal tahun 1980-an. Bahasa ini merupakan pengembangan dari bahasa C dengan menambahkan fitur pemrograman berorientasi objek (object-oriented programming). C++ dirancang untuk memberikan fleksibilitas, efisiensi, dan kekuatan yang tinggi dalam pengembangan perangkat lunak.

- **Karakteristik Utama C++**
 - ❖ **Multi-paradigma:**
C++ mendukung berbagai paradigma pemrograman, seperti:
 - Pemrograman Prosedural (berbasis fungsi).
 - Pemrograman Berorientasi Objek (OOP).
 - Pemrograman Generik (menggunakan template).
 - ❖ **Portabel:** Program yang ditulis dalam C++ dapat dijalankan di berbagai platform dengan sedikit atau tanpa modifikasi.
 - ❖ **Efisiensi Tinggi:** C++ memberikan kontrol langsung atas sumber daya perangkat keras, seperti memori, sehingga cocok untuk aplikasi yang memerlukan kinerja tinggi.

- ❖ Ekstensi Bahasa C: Karena merupakan pengembangan dari C, hampir semua kode C dapat dijalankan dalam lingkungan C++.
- Kelebihan C++
 - ❖ Kinerja Tinggi: Cocok untuk aplikasi dengan kebutuhan tinggi seperti sistem operasi, game, dan perangkat lunak sistem.
 - ❖ Fleksibilitas: Mendukung berbagai gaya pemrograman, sehingga dapat digunakan untuk berbagai jenis proyek.
 - ❖ Komunitas Besar: Memiliki dokumentasi yang luas, banyak pustaka (library), dan komunitas global yang aktif.
 - ❖ Pemrograman Berorientasi Objek: Mendukung konsep-konsep seperti class, inheritance, polymorphism, dan encapsulation, yang membantu dalam pengembangan perangkat lunak yang kompleks.
- Kekurangan C++
 - ❖ Kompleksitas: Fitur yang sangat luas dapat membuat bahasa ini sulit dipelajari untuk pemula.
 - ❖ Manajemen Memori Manual: Walaupun memberi kontrol lebih besar, pengguna harus berhati-hati dalam mengelola memori untuk menghindari kebocoran (memory leaks).
 - ❖ Kesalahan Sulit Dideteksi: Kesalahan seperti pointer yang salah atau kondisi undefined behavior dapat menyebabkan program gagal dengan cara yang sulit dipahami.

BAB III

Pembahasan

3.1. Sebelum Transformasi diimplementasikan

Source code program sebelum transformasi diimplementasikan dijelaskan pada **Tabel 4.1**.

Tabel 4.1 Penjelasan Source Code Program Simulasi Sistem Smart Home

Program	Keterangan
<pre> void cloud1(){ glPushMatrix(); //circle-1 int jeight; GLfloat p8=.34f; GLfloat q8=.8f; GLfloat radius8 =.08f; int triangleAmount8 = 20; // # of triangles used to draw circle //GLfloat radius = 0.8f; //radius GLfloat twicePi8 = 2.0f * PI; glBegin(GL_TRIANGLE_FAN); glColor3ub(253, 254, 254); glVertex2f(p8, q8); // center of circle for(jeight = 0; jeight <= triangleAmount8; jeight++) { glVertex2f(p8 + (radius8 * cos(jeight * twicePi8 / triangleAmount8)), q8 + (radius8 * sin(jeight * twicePi8 / triangleAmount8))); } glEnd(); //circle-2 int jnine; GLfloat p9=.41f; GLfloat q9=.87f; GLfloat radius9 =.07f; int triangleAmount9 = 20; // # of triangles used to draw circle //GLfloat radius = 0.8f; //radius GLfloat twicePi9 = 2.0f * PI; glBegin(GL_TRIANGLE_FAN); glColor3ub(253, 254, 254); glVertex2f(p9, q9); // center of circle for(jnine = 0; jnine <= triangleAmount9; jnine++) { </pre>	<p>Pada void cloud 1 (void untuk menggambar awan) tidak ada transformasi yang dilakukan</p>

Tabel 4.1 Penjelasan Source Code Program Simulasi Sistem Smart Home

Program	Keterangan
<pre> glVertex2f(p9 + (radius9 * cos(jnine * twicePi9 / triangleAmount9)), q9 + (radius9 * sin(jnine * twicePi9 / triangleAmount9))); } glEnd(); //circle-3 int jten; GLfloat p10=.30f; GLfloat q10=.87f; GLfloat radius10 =.08f; int triangleAmount10 = 20; // # of triangles used to draw circle //GLfloat radius = 0.8f; //radius GLfloat twicePi10 = 2.0f * PI; glBegin(GL_TRIANGLE_FAN); glColor3ub(253, 254, 254); glVertex2f(p10, q10); // center of circle for(jten = 0; jten <= triangleAmount10; jten++) { glVertex2f(p10 + (radius10 * cos(jten * twicePi10 / triangleAmount10)), q10 + (radius10 * sin(jten * twicePi10 / triangleAmount10))); } glEnd(); //circle-4 int jeleven; GLfloat p11=.45f; GLfloat q11=.8f; GLfloat radius11 =.08f; int triangleAmount11 = 20; // # of triangles used to draw circle //GLfloat radius = 0.8f; //radius GLfloat twicePi11 = 2.0f * PI; glBegin(GL_TRIANGLE_FAN); glColor3ub(253, 254, 254); glVertex2f(p11, q11); // center of circle for(jeleven = 0; jeleven <= triangleAmount11; jeleven++) { glVertex2f(p11 + (radius11 * cos(jeleven * twicePi11 / triangleAmount11)), q11 + (radius11 * sin(jeleven * twicePi11 / triangleAmount11))); } glEnd(); </pre>	

Tabel 4.1 Penjelasan Source Code Program Simulasi Sistem Smart Home

Program	Keterangan
<pre>); } glEnd(); glPopMatrix(); glFlush(); } </pre>	
<pre> void Bigtree(){ glPushMatrix(); glBegin(GL_QUADS); glColor3ub(139, 71, 38); glVertex2f(-0.8f, -0.8f); glVertex2f(-1.0f, -0.8f); glVertex2f(-0.95f, 0.3f); glVertex2f(-0.9f, 0.3f); glEnd(); // Apply rotation transformation to the leaves float windAngle = sin(timeValue) * 3.0f; // Oscillate angle between -3 to 3 degrees glPushMatrix(); // Draw leaves glBegin(GL_TRIANGLES); // Leaf 1 glColor3ub(46, 139, 87); glVertex2f(-0.6f, -0.4f); glVertex2f(-0.925f, 0.1f); glVertex2f(-0.925f, 0.3f); // Leaf 2 glVertex2f(-0.4f, -0.3f); glVertex2f(-0.925f, 0.1f); glVertex2f(-0.925f, 0.3f); // Leaf 3 glVertex2f(-0.5f, -0.1f); glVertex2f(-0.925f, 0.1f); glVertex2f(-0.925f, 0.3f); // Leaf 4 glVertex2f(-0.4f, 0.1f); </pre>	<p>Pada void Bigtree (void untuk menggambar pohon) tidak ada transformasi yang dilakukan</p>

Tabel 4.1 Penjelasan Source Code Program Simulasi Sistem Smart Home

Program	Keterangan
<pre> glVertex2f(-0.925f, 0.1f); glVertex2f(-0.925f, 0.3f); // Leaf 5 glVertex2f(-0.4f, 0.5f); glVertex2f(-0.925f, 0.0f); glVertex2f(-0.925f, 0.3f); // Leaf 6 glVertex2f(-0.6f, 0.8f); glVertex2f(-0.925f, 0.0f); glVertex2f(-0.925f, 0.3f); // Leaf 7 glVertex2f(-0.8f, 0.9f); glVertex2f(-0.95f, 0.3f); glVertex2f(-0.925f, 0.1f); // Leaf 8 glVertex2f(-1.0f, 0.9f); glVertex2f(-0.9f, 0.3f); glVertex2f(-0.925f, 0.1f); glEnd(); glPopMatrix(); glBegin(GL_QUADS); glColor3ub(46, 139, 87); glVertex2f(-1.0f, 0.0f); glVertex2f(-1.0f, 0.4f); glVertex2f(-0.925f, 0.3f); glVertex2f(-0.925f, 0.0f); glEnd(); glPopMatrix(); } </pre>	
<pre> void chair(){ //Chair Umbrella glBegin(GL_TRIANGLES); glColor3ub(188,143,143); glVertex2f(1.0f, -0.2f); glVertex2f(0.4f, -0.2); glVertex2f(0.7f, 0.1f); </pre>	<p>Pada void chair (void untuk menggambar kursi) tidak ada transformasi yang dilakukan</p>

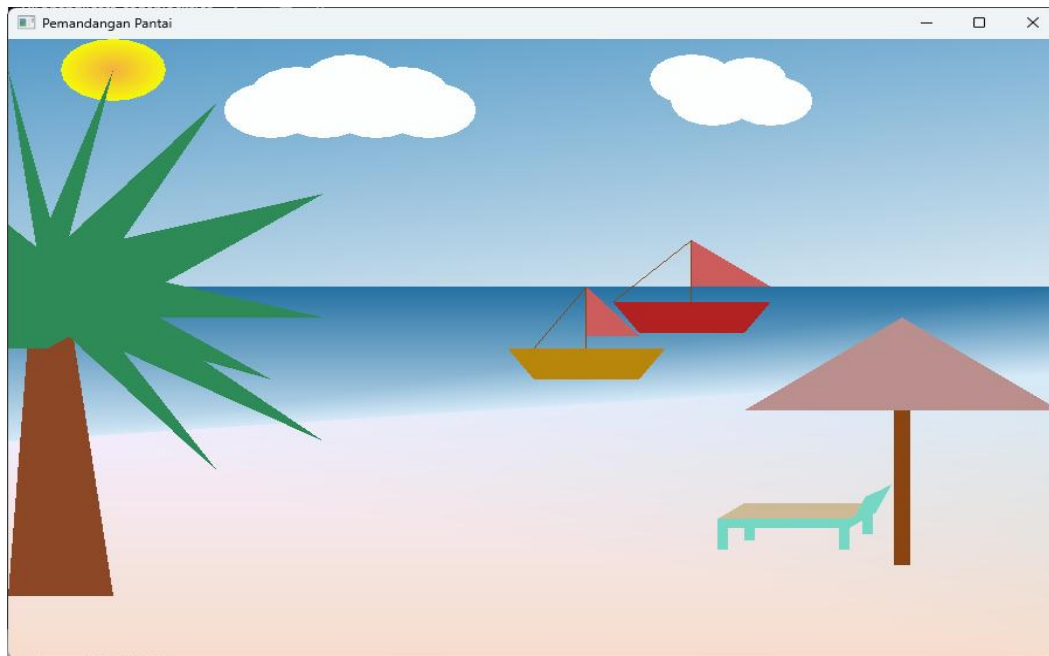
Tabel 4.1 Penjelasan Source Code Program Simulasi Sistem Smart Home

Program	Keterangan
<pre> glEnd(); //Chair Umbrella Stand glBegin(GL_QUADS); glColor3ub(139, 69, 19); glVertex2f(0.715f, -0.7f); glVertex2f(0.685f, -0.7f); glVertex2f(0.685f, -0.2f); glVertex2f(0.715f, -0.2f); glEnd(); //-----chair----- //sit glBegin(GL_QUADS); glColor3ub(205, 186, 150); glVertex2f(0.65f, -0.5f); glVertex2f(0.4f, -0.5f); glVertex2f(0.35f, -0.55f); glVertex2f(0.6f, -0.55f); glEnd(); //sit left label glBegin(GL_QUADS); glColor3ub(118, 215, 196); glVertex2f(0.6f, -0.58f); glVertex2f(0.6f, -0.55f); glVertex2f(0.35f, -0.55f); glVertex2f(0.35f, -0.58); glEnd(); //sit back label glBegin(GL_QUADS); glColor3ub(118, 215, 196); glVertex2f(0.65f, -0.50f); glVertex2f(0.65f, -0.53f); glVertex2f(0.6f, -0.58f); glVertex2f(0.6f, -0.55f); glEnd(); //chair relaxer glBegin(GL_QUADS); glColor3ub(118, 215, 196); glVertex2f(0.65f, -0.53f); glVertex2f(0.6f, -0.58f); glVertex2f(0.63f, -0.48); glVertex2f(0.68f, -0.44); glEnd(); //chair legs - left front glBegin(GL_QUADS); </pre>	

Tabel 4.1 Penjelasan Source Code Program Simulasi Sistem Smart Home

Program	Keterangan
<pre> glColor3ub(118, 215, 196); glVertex2f(0.35f, -0.58f); glVertex2f(0.37f, -0.58f); glVertex2f(0.37f, -0.65f); glVertex2f(0.35f, -0.65f); glEnd(); //chair legs - right front glBegin(GL_QUADS); glColor3ub(118, 215, 196); glVertex2f(0.4f, -0.58f); glVertex2f(0.42f, -0.58f); glVertex2f(0.42f, -0.62f); glVertex2f(0.4f, -0.62f); glEnd(); //chair legs - right back glBegin(GL_QUADS); glColor3ub(118, 215, 196); glVertex2f(0.645f, -0.52f); glVertex2f(0.645f, -0.6f); glVertex2f(0.625f, -0.6f); glVertex2f(0.625f, -0.52f); glEnd(); //chair legs - left back glBegin(GL_QUADS); glColor3ub(118, 215, 196); glVertex2f(0.6f, -0.58f); glVertex2f(0.6f, -0.65f); glVertex2f(0.58f, -0.65f); glVertex2f(0.58f, -0.58f); glEnd(); } </pre>	

Output hasil kode program sebelum transformasi diimplementasikan dapat dilihat pada **Gambar 3.1.**



3.2. Setelah Transformasi diimplementasikan

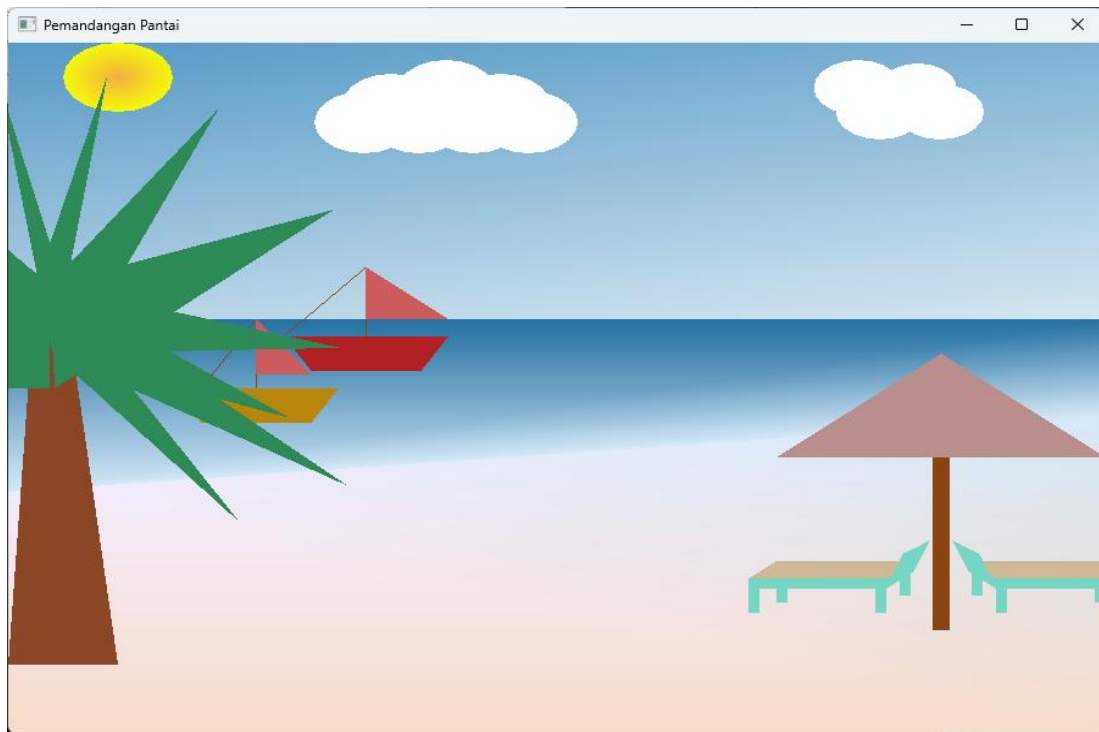
Tabel 4.1 Penjelasan Source Code Program Simulasi Sistem Smart Home

Program	Keterangan
<pre> GLfloat positionCloud1 = 0.4f; GLfloat speed = 0.009f; //declared globally for every movable object... ..SPEED... ..!!! !!! !!! void updateCloud1(int value) { if(positionCloud1 <-1.0) positionCloud1 = 1.0f; positionCloud1 -= speed; glutPostRedisplay(); glutTimerFunc(100, updateCloud1, 0); } void cloud1(){ ... </pre>	<p>Translasi diimplementasikan dengan menambah void updateCloud1 (untuk mengatur kecepatan gerak) dan variabel GLfloat position dan GLfloat speed.</p> <p>Pada void cloud1 menambahkan glTranslatef untuk mengaktifkan translasi pada objek</p>

Tabel 4.1 Penjelasan Source Code Program Simulasi Sistem Smart Home

Program	Keterangan
<pre>glTranslatef(-positionCloud1,0.0f,0.0f); ... }</pre>	
<pre>void BigTree(){ ... glTranslatef(-0.925f, 0.3f, 0.0f);// set pivot point glRotatef(windAngle, 0.0f, 0.0f, 1.0f); glTranslatef(0.925f, -0.3f, 0.0f); ... }</pre>	<p>Rotasi diimplementasikan dengan menambah kode glTranslatef dan glRotate.</p> <p>glTranslate untuk menentukan pivot point dan glRotate untuk mengaktifkan rotasi pada objek</p>
<pre>void chairReflection() { glPushMatrix(); // Simpan matriks transformasi saat ini glScalef(-1.0f, 1.0f, 1.0f); // Lakukan refleksi pada sumbu Y glTranslatef(-1.4f, 0.0f, 0.0f); // Geser kursi refleksi agar tidak menumpuk dengan kursi asli chair(); // Gambar kursi asli dalam matriks transformasi terbalik glPopMatrix(); // Pulihkan matriks transformasi } void day(){ ... chairReflection(); ... }</pre>	<p>Refleksi diimplementasikan dengan membuat refleksi dengan void chairReflection. Kode glScalef (untuk merefleksi pada sumbu Y) dan glTranslatef (untuk menggeser refleksi agar tidak menumpuk)</p> <p>Pada void main (atau day pada kode ini) void chairReflection dipanggil untuk mengaktifkan transformasi refleksi objek.</p>

Output hasil kode program setelah transformasi diimplementasikan dapat dilihat pada **Gambar 3.2.**



BAB IV

Penutup

4.1. Kesimpulan

Berdasarkan laporan yang telah dilakukan, dapat disimpulkan bahwa transformasi objek 2D merupakan teknik yang sangat bermanfaat dalam menciptakan karya visual yang estetis dan menarik, terutama dengan tema pemandangan pantai. Teknik-teknik seperti translasi, rotasi, refleksi, skala, dan shearing memungkinkan elemen-elemen visual, seperti laut, pasir, matahari, dan pohon kelapa, untuk dimanipulasi menjadi bentuk yang lebih dinamis dan ekspresif.

Penerapan transformasi objek 2D bertemakan pemandangan pantai tidak hanya meningkatkan nilai estetika, tetapi juga memberikan potensi besar dalam pengembangan media kreatif, seperti desain grafis, animasi, dan media pembelajaran. Meski menghadapi beberapa tantangan, seperti kesesuaian proporsi dan harmoni visual, tantangan tersebut dapat diatasi melalui perencanaan dan eksplorasi yang matang.

Dengan demikian, transformasi objek 2D bertemakan pemandangan pantai memberikan ruang luas bagi inovasi dalam seni dan desain grafis. Hasil laporan ini diharapkan dapat menjadi referensi bagi para pelaku seni, desainer, dan pendidik untuk terus mengeksplorasi teknik-teknik kreatif dalam menciptakan karya yang bermanfaat dan inspiratif.

Daftar Pustaka

<https://github.com/mdsowmikaiub/ComputerGraphics-SeaBeach>
<https://www.codeblocks.org/>
<https://id.wikipedia.org/wiki/C%2B%2B>