

IMPLEMENTASI POSENET DALAM GAME SEMAPHORE UNTUK MENGENALI GERAKAN TUBUH PENGGUNA

Baby Aisha Maritza Virginia^{1*}, Anis Cherid²

^{1,2}Fakultas Ilmu Komputer, Universitas Mercu Buana
Email: ^{1*}41520010072@student.mercubuana.ac.id, ²anis.cherid@mercubuana.ac.id

(*: *Corresponding Author*)

(Naskah masuk: 2 Desember 2023, diterima untuk diterbitkan: 27 Desember 2023)

Abstrak

Saat ini, pembelajaran semaphore masih dilakukan secara manual dengan bimbingan instruktur, yang memiliki keterbatasan dalam ketersediaan instruktur dan efisiensi waktu. Penelitian ini bertujuan untuk membuat *game* semaphore yang mengimplementasikan PoseNet untuk meningkatkan efisiensi dan interaktivitas pembelajaran gestur semaphore. Melalui penerapan teknologi PoseNet, *game* ini bertujuan untuk mengenali gerakan tubuh pengguna secara *real-time*. Dengan mengidentifikasi 17 titik kunci pada tubuh manusia, PoseNet memungkinkan deteksi gestur semaphore secara otomatis dan secara *real-time*. Metode dan model pengembangan sistem yang digunakan adalah model prototipe & MDLC, dan penerapan library p5.js dan ml5.js memberikan basis untuk integrasi PoseNet ke dalam *game* Semaphore. Hasil penelitian ini menyajikan *game* semaphore yang mengimplementasikan PoseNet sebagai solusi inovatif untuk meningkatkan pembelajaran gestur semaphore.

Kata kunci: *semaphore, posenet, body recognition, game interaktif.*

IMPLEMENTATION OF POSENET IN SEMAPHORE GAME FOR USER BODY MOVEMENT RECOGNITION

Abstract

Currently, semaphore learning is still done manually with instructor guidance, which has limitations in instructor availability and time efficiency. This research aims to create a semaphore game that implements PoseNet to increase the efficiency and interactivity of learning semaphore gestures. Through the application of PoseNet technology, this game aims to recognize the user's body movements in real-time. By identifying 17 key points on the human body, PoseNet enables automatic, real-time detection of semaphore gestures. The system development method and model used is the prototype & MDLC model, and the implementation of the p5.js and ml5.js libraries provides the basis for the integration of PoseNet into the Semaphore game. The results of this research present a semaphore game that implements PoseNet as an innovative solution to improve semaphore gesture learning.

Keywords: *semaphore, posenet, body recognition, interactive game.*

1. PENDAHULUAN

Semaphore adalah suatu cara berkomunikasi jarak jauh yang biasa dipraktikkan dalam kegiatan kepramukaan. Informasi disampaikan melalui gestur atau gerakan-gerakan khusus menggunakan alat bantu berupa bendera atau dayung, batang. Pembelajaran gestur semaphore saat ini masih dilakukan secara manual dengan bimbingan instruktur, yang memiliki keterbatasan dalam ketersediaan instruktur dan efisiensi waktu. Untuk mengatasi masalah ini, penelitian ini mengusulkan implementasi PoseNet dalam *Game* Semaphore. PoseNet adalah teknologi yang telah digunakan dalam berbagai aplikasi, seperti deteksi dan koreksi

pose yoga[13], estimasi pose manusia, dan aplikasi fisioterapi dan rehabilitasi. Teknologi ini juga telah digunakan dalam pengembangan *game* berbasis gestur dan aplikasi deteksi gerakan tubuh lainnya.

PoseNet bekerja dengan mengidentifikasi 17 titik kunci pada tubuh manusia, memungkinkan deteksi gestur semaphore secara otomatis[3]. Dalam konteks *game*, ini dapat meningkatkan interaktivitas dan efisiensi. Model pengembangan sistem yang digunakan dalam penelitian ini adalah model prototipe dan MDLC, dan *library* p5.js dan ml5.js digunakan sebagai basis untuk integrasi PoseNet ke dalam *game* Semaphore.

Tujuan dari penelitian ini adalah untuk mengenali gerakan tubuh pengguna secara *real-time* melalui penerapan teknologi PoseNet. Diharapkan, penelitian ini dapat memberikan solusi inovatif untuk meningkatkan pembelajaran gestur semaphore.

Ada penelitian terdahulu yang telah mencoba mengatasi masalah ini. Penelitian oleh A. M. Syahrul dan M. I. Rahayu yang membuat aplikasi *game* semaphore berbasis android[16]. Penelitian ini tidak menerapkan *body recognition* untuk mendeteksi dan menginterpretasikan gerakan semaphore pengguna. Oleh karena itu, penelitian ini mencoba untuk mengatasi keterbatasan tersebut dengan menggunakan teknologi PoseNet yang lebih mudah diakses.

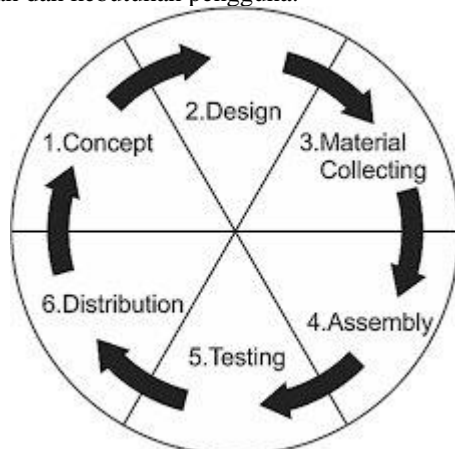
2. METODE

2.1 Metode Pengembangan Sistem

Metode penelitian yang digunakan dalam pengembangan aplikasi *Game* Semaphore dengan implementasi PoseNet menggabungkan pendekatan Prototipe dan *Multimedia Development Life Cycle* (MDLC). Pendekatan ini dirancang untuk memastikan pengembangan multimedia yang terstruktur dan efisien, sehingga produk akhir dapat memenuhi tujuan dan kebutuhan pengguna.

Model prototipe digunakan dalam pengembangan aplikasi ini. Pendekatan ini melibatkan beberapa tahap yang dilalui dalam proses pembuatannya. Tahap-tahap tersebut melibatkan konsepsi ide, desain awal, implementasi, dan evaluasi. Jika pada tahap evaluasi ditemukan bahwa sistem yang dibangun belum sempurna, dilakukan evaluasi ulang dan perbaikan.

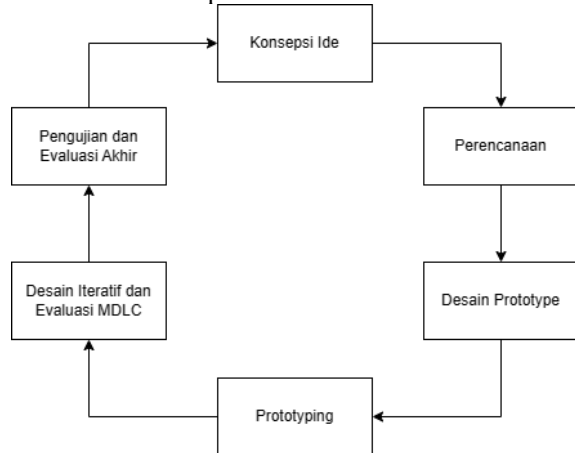
MDLC digunakan sebagai kerangka kerja sistematis dalam pengembangan multimedia. Tahapan MDLC yang diterapkan mencakup konsepsi ide, perencanaan, desain, produksi, pengujian, dan implementasi. Setiap tahapannya dirancang untuk memastikan pengembangan yang terstruktur dan efisien, sehingga produk multimedia dapat memenuhi tujuan dan kebutuhan pengguna.



Gambar 1. Multimedia Development Life Cycle

Implementasi teknologi PoseNet bertujuan untuk mendeteksi dan melacak pose tubuh pengguna dalam

waktu nyata. Dengan memanfaatkan *model machine learning* PoseNet, 17 titik kunci pada tubuh manusia dapat dipantau, memungkinkan pengenalan gestur Semaphore secara akurat[1]. Proses implementasi melibatkan integrasi PoseNet ke dalam perangkat lunak *Game* Semaphore.



Gambar 2. Tahapan Penelitian

Tahapan dalam penelitian ini disesuaikan dengan model yang diimplementasikan dan meliputi langkah-langkah berikut:

1. Konsepsi Ide

Pada tahap ini, identifikasi kebutuhan dan tujuan pengembangan aplikasi *Game* Semaphore dengan PoseNet. Lakukan studi literatur untuk memahami konsep Semaphore, PoseNet, dan metode pengembangan multimedia. Sosialisasikan ide dengan para ahli dan calon pengguna untuk mendapatkan masukan awal.

2. Perencanaan

a. Identifikasi Pengguna dan Kebutuhan

Aplikasi *Game* Semaphore dengan PoseNet ditujukan untuk dua kelompok utama pengguna. Pertama, target utama adalah para pelajar, khususnya murid sekolah dasar hingga sekolah menengah. Aplikasi ini dapat digunakan sebagai alat pembelajaran yang interaktif dan menyenangkan, membantu mereka memahami dan menguji keterampilan mereka dalam mengenali gerakan Semaphore.

Kedua, aplikasi ini juga dapat bermanfaat bagi instruktur atau guru yang ingin memberikan pengajaran Semaphore secara inovatif dan efisien.

Dari segi kebutuhan fungsional, aplikasi ini harus dapat secara akurat mendeteksi gerakan Semaphore yang dilakukan oleh pengguna menggunakan teknologi PoseNet. Integrasi PoseNet juga diharapkan mampu melacak pose tubuh pengguna secara *real-time*, memungkinkan evaluasi gerakan dengan akurasi tinggi[14]. Selain itu, antarmuka aplikasi dirancang untuk menjadi interaktif dan edukatif, bertujuan untuk meningkatkan pemahaman pengguna terhadap gerakan Semaphore. Adanya umpan balik *real-time* menjadi fitur penting untuk memberikan informasi segera tentang kebenaran atau kesalahan dalam

eksekusi gerakan Semaphore, memungkinkan perbaikan secara instan.

Dari segi kebutuhan non-fungsional, aplikasi diharapkan responsif terhadap gerakan tubuh pengguna, menciptakan pengalaman pengguna yang mulus. Antarmuka pengguna dirancang agar sederhana dan intuitif, memastikan pengguna dari berbagai kelompok usia dapat dengan mudah mengakses dan menggunakan aplikasi ini.

Ketersediaan platform menjadi aspek penting, sehingga aplikasi dapat diakses baik melalui perangkat *mobile* maupun desktop. Dalam hal keamanan data pengguna, aplikasi ini harus memastikan perlindungan data, terutama karena melibatkan penggunaan kamera untuk deteksi gerakan tubuh. Terakhir, performa tinggi menjadi fokus untuk menjaga kualitas pelacakan pose tubuh dan deteksi gerakan Semaphore agar berjalan secara optimal.[2]

b. Perancangan Prototipe Awal

Dalam prototipe pertama, beberapa fitur utama yang akan diimplementasikan dalam aplikasi *Game Semaphore* dengan PoseNet melibatkan aspek deteksi gerakan dan pengalaman pengguna yang interaktif. Berikut adalah fitur-fitur utama yang akan menjadi fokus implementasi pada tahap awal:

1) Deteksi Gerakan Semaphore:

Integrasi PoseNet untuk mendeteksi gerakan Semaphore yang dilakukan oleh pengguna. Identifikasi dan pelacakan pose tubuh dengan akurasi tinggi untuk memastikan interpretasi gerakan yang tepat[3].

2) *Interface* Interaktif:

Desain antarmuka yang bersifat interaktif dan intuitif agar pengguna dapat dengan mudah berinteraksi dengan aplikasi. Visualisasi yang jelas dari gerakan Semaphore untuk memfasilitasi pemahaman pengguna[6].

3) Umpan Balik *Real-time*:

Sistem umpan balik langsung kepada pengguna setelah melakukan gerakan Semaphore, memberikan informasi segera jika gerakan benar.

4) Integrasi p5.js dan ml5.js:

Pemanfaatan p5.js untuk membuat elemen visual yang menarik dan interaktif pada aplikasi. Integrasi ml5.js untuk mendukung fitur *machine learning*, seperti evaluasi gerakan dan perbaikan *real-time*.

Implementasi fitur-fitur ini di dalam prototipe pertama bertujuan untuk memberikan pengalaman pengguna yang memadai dan mendemonstrasikan kemampuan deteksi gerakan serta interaktivitas aplikasi *Game Semaphore* dengan PoseNet[5]. Selain itu, fitur-fitur ini juga mendukung tujuan pembelajaran dan memberikan pengguna umpan balik yang berguna dalam mengasah keterampilan mereka dalam mengenali gerakan Semaphore.

3. Desain Prototipe

a. Implementasi PoseNet

```
57 function setup() {
58   createCanvas(windowHeight * 1.4, windowHeight * 0.8);
59   video = createCapture(VIDEO);
60   videoWidth = width;
61   videoHeight = width * 0.75;
62   video.size(videoWidth, videoHeight);
63   video.hide();
64   const poseNet = ml5.poseNet(video, modelLoaded);
65   poseNet.on("pose", gotPoses);
```

Gambar 3. Potongan code bagian Implementasi Algoritma PoseNet

Implementasi PoseNet dalam kode tersebut dapat dijelaskan sebagai berikut:

- 1) `createCanvas(windowHeight * 1.4, windowHeight * 0.8);`: Kode ini membuat elemen *canvas* dengan lebar sekitar 1.4 kali tinggi jendela browser dan tinggi sekitar 0.8 kali tinggi jendela browser. Ini akan digunakan untuk menampilkan konten visual, termasuk gambar dari kamera.
- 2) `video = createCapture(VIDEO);`: Baris ini menciptakan elemen video dengan mengakses kamera *webcam* pengguna. Video ini akan digunakan sebagai sumber masukan untuk deteksi pose oleh PoseNet.
- 3) `videoWidth = width; dan videoHeight = width * 0.75;`: Variabel `videoWidth` dan `videoHeight` digunakan untuk menentukan lebar dan tinggi video sesuai dengan lebar *canvas* yang telah dibuat sebelumnya, dengan aspek rasio 4:3 (0.75).
- 4) `video.size(videoWidth, videoHeight);`: Mengatur dimensi video yang diambil dari kamera sesuai dengan `videoWidth` dan `videoHeight` yang telah dihitung sebelumnya.
- 5) `video.hide();`: Kode ini menyembunyikan elemen video, sehingga video tidak akan ditampilkan secara langsung di halaman web.
- 6) `const poseNet = ml5.poseNet(video, modelLoaded);`: Kode ini membuat objek PoseNet menggunakan pustaka ml5.js. Ini mengambil elemen video (*webcam*) sebagai sumber data dan akan memicu fungsi `modelLoaded` ketika model PoseNet telah dimuat sepenuhnya.
- 7) `poseNet.on("pose", gotPoses);`: Kode ini menghubungkan fungsi `gotPoses` ke peristiwa "pose". Ini berarti ketika PoseNet mendeteksi pose manusia dalam video, ia akan memanggil fungsi `gotPoses` dan memberikan data pose tersebut untuk diolah lebih lanjut.

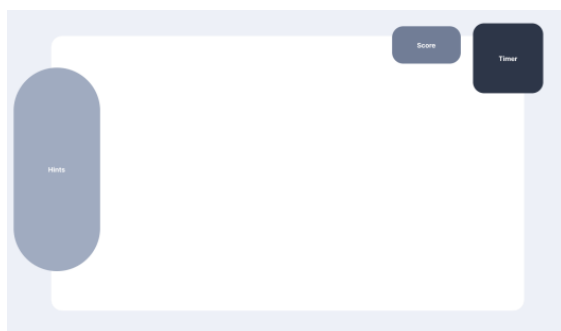
Dengan implementasi ini, PoseNet akan terus memantau video dari kamera pengguna dan mengirimkan data pose manusia ke fungsi `gotPoses` ketika ada deteksi pose yang terjadi. Fungsi `gotPoses` kemudian dapat digunakan untuk mengakses dan memproses data pose tersebut, misalnya untuk

mengenali gerakan atau tindakan berdasarkan pose yang dikenali

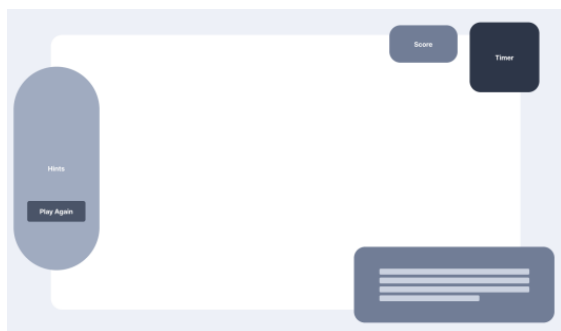
b. Desain Multimedia



Gambar 4. Wireframing tampilan utama



Gambar 5. Wireframing tampilan in-game



Gambar 6. Wireframing tampilan game selesai

4. Prototyping

a. Pembuatan Prototipe Awal

Untuk inialisasi proyek, struktur folder proyek disusun dengan terpisah antara file HTML, CSS, dan JavaScript. File-file ini disusun agar memudahkan pengembangan dan pemeliharaan. Selain itu, pustaka p5.js dan ml5.js diperlukan untuk menghadirkan elemen visual dan fitur *machine learning*[17]. Kedua pustaka ini dapat disertakan melalui tautan atau dapat diunduh secara lokal untuk integrasi dalam proyek.

Dalam tahap implementasi deteksi gerakan Semaphore menggunakan PoseNet, PoseNet diintegrasikan dengan memuat model dan menyiapkan objek deteksi. Penggunaan kamera web atau video sebagai *input* memungkinkan pengambilan data gerakan tubuh pengguna. Selanjutnya, titik-titik kunci yang dihasilkan oleh PoseNet digunakan untuk mengidentifikasi gerakan Semaphore. Proses ini memastikan aplikasi mampu

akurat menangkap dan mengenali gerakan yang dilakukan oleh pengguna.

Pada tahap antarmuka interaktif dan umpan balik real-time, p5.js dimanfaatkan untuk membuat elemen antarmuka yang interaktif[10], seperti tombol dan petunjuk. Selain itu, implementasi umpan balik real-time ditambahkan untuk memberikan informasi langsung kepada pengguna mengenai kebenaran atau kesalahan dalam eksekusi gerakan Semaphore. Hal ini meningkatkan interaktivitas aplikasi dan memungkinkan pengguna untuk memperbaiki gerakan secara instan.

Dengan langkah-langkah tersebut, proyek ini mulai membentuk prototipe awal yang menggabungkan elemen visual menarik, deteksi gerakan real-time menggunakan PoseNet, serta interaktivitas antarmuka yang mendukung pembelajaran gerakan Semaphore dengan efektif.

b. Evaluasi Prototipe Awal

1) Pengalaman Pengguna (UX):

Positif: Antarmuka interaktif mudah digunakan, menyajikan instruksi yang jelas untuk gerakan Semaphore.

Perbaikan: Menambahkan animasi atau panduan visual lebih lanjut untuk meningkatkan pemahaman pengguna.

2) Deteksi Gerakan:

Positif: PoseNet mampu mendeteksi gerakan tubuh dengan akurasi tinggi.

Perbaikan: Memastikan deteksi gerakan tetap konsisten pada berbagai posisi dan kondisi pencahayaan.

3) Umpan Balik *Real-time*:

Positif: Umpan balik langsung memberikan informasi yang berguna setelah setiap gerakan.

Perbaikan: Menyesuaikan umpan balik untuk lebih spesifik, memberikan petunjuk perbaikan yang lebih rinci.

4) Mode Pembelajaran:

Positif: Mode pembelajaran memberikan panduan langkah demi langkah yang efektif.

Perbaikan: Menambahkan opsi untuk mengulangi instruksi atau mendapatkan bantuan tambahan saat diperlukan.

5) Rekam dan Evaluasi Gerakan:

Positif: Fitur rekam dan evaluasi memberikan kemungkinan untuk mereview dan memperbaiki gerakan.

Perbaikan: Menambahkan fitur *rewind* atau *slow-motion* untuk memudahkan analisis gerakan.

6) Responsif dan Ketersediaan Platform:

Positif: Aplikasi responsif pada berbagai perangkat, termasuk perangkat *mobile* dan desktop.

Perbaikan: Menyesuaikan tata letak antarmuka untuk memastikan konsistensi pada ukuran layar yang lebih kecil.

7) Integrasi p5.js dan ml5.js:

Positif: Integrasi p5.js dan ml5.js memberikan elemen visual menarik dan mendukung fitur machine learning.

Perbaikan: Optimalisasi performa dan memastikan konsistensi integrasi pada berbagai skenario penggunaan.

5. Desain Iteratif dan Evaluasi MDLC

a. Evaluasi MDLC

1) Konsepsi Ide:

Positif: Ide pengembangan *Game* Semaphore dengan PoseNet telah jelas dan relevan.

Perbaikan: Melakukan survei lanjutan untuk memastikan ide tersebut tetap relevan dengan kebutuhan pengguna.

2) Perencanaan:

Positif: Rencana pengembangan dan penjadwalan tahapan MDLC telah disusun dengan baik.

Perbaikan: Meninjau jadwal dan sumber daya untuk memastikan keberlanjutan pengembangan.

3) Desain Sistem:

Positif: Desain sistem telah mempertimbangkan integrasi PoseNet, antarmuka interaktif, dan umpan balik *real-time*.

Perbaikan: Menyempurnakan desain sistem untuk meningkatkan keterkaitan antara elemen-elemen aplikasi.

4) Pengembangan Prototipe:

Positif: Prototipe awal berhasil diimplementasikan dengan PoseNet dan fitur utama terintegrasi.

Perbaikan: Melakukan perbaikan kecil pada deteksi gerakan dan memperbarui antarmuka berdasarkan umpan balik pengguna.

5) Evaluasi Prototipe:

Positif: Evaluasi awal menunjukkan keberhasilan dalam deteksi gerakan dan pengalaman pengguna yang baik.

Perbaikan: Menyesuaikan beberapa elemen antarmuka dan meningkatkan umpan balik *real-time*.

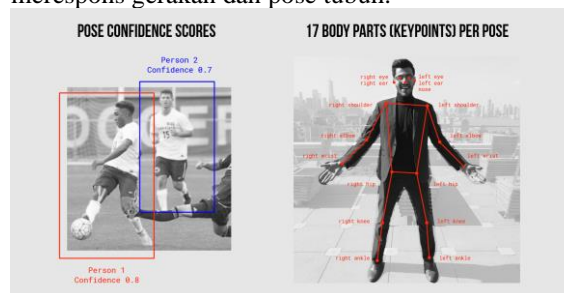
2.2 Library dan Komponen Perangkat Lunak yang Digunakan

Dalam implementasi PoseNet dalam *Game* Semaphore, elemen HTML digunakan untuk membangun struktur dasar halaman web. Kontennya dapat mencakup *tag* judul, paragraf untuk menjelaskan permainan, tautan untuk sumber daya eksternal, dan elemen-elemen lain yang mendukung tampilan dan fungsionalitas *game*.

CSS digunakan untuk mendefinisikan presentasi elemen-elemen HTML dalam *Game* Semaphore. Ini mencakup penyesuaian tata letak, warna, dan gaya visual agar sesuai dengan konsep permainan. CSS memungkinkan pengaturan yang konsisten dan estetika yang menarik, menciptakan pengalaman pengguna yang lebih menyenangkan dan profesional.

JavaScript memainkan peran kunci dalam implementasi PoseNet. Dengan menggunakan PoseNet, JavaScript dapat memanipulasi Model Objek Dokumen (DOM)[9] untuk melacak dan menganalisis gerakan tubuh pengguna dalam *real-time*. JavaScript juga dapat digunakan untuk mengintegrasikan PoseNet dengan logika permainan, memastikan responsivitas dan interaktivitas permainan Semaphore[12].

PoseNet adalah model machine learning yang dikembangkan oleh Google yang dapat memperkirakan pose manusia secara *real-time* menggunakan kamera web atau gambar input[7]. Ini mengidentifikasi titik-titik kunci pada tubuh manusia, seperti sendi dan keypoints, memungkinkan pengembangan untuk melacak dan menganalisis gerakan tubuh. PoseNet umumnya digunakan dalam aplikasi terkait kebugaran, realitas terkini, dan interaksi manusia-komputer untuk menangkap dan merespons gerakan dan pose tubuh.



Gambar 7. Implementasi PoseNet

PoseNet diintegrasikan dalam implementasi *Game* Semaphore untuk mengidentifikasi dan melacak gerakan tubuh pengguna. Dengan memanfaatkan model *machine learning* ini, permainan dapat memahami pose manusia dalam gambar atau video secara *real-time*[11]. PoseNet menetapkan titik-titik kunci pada tubuh, yang akan digunakan untuk memonitor dan mengoreksi gerakan tubuh pengguna saat melakukan gestur semaphore[8].

p5.js adalah perpustakaan JavaScript yang dirancang untuk *coding* kreatif dan membuat visualisasi. Ini menyederhanakan proses membuat grafis dan konten interaktif di web. p5.js menyediakan serangkaian fungsi menggambar dan utilitas, membuatnya mudah diakses bagi seniman, desainer, dan pemula untuk bereksperimen dengan pemrograman dan mengekspresikan kreativitas mereka melalui elemen visual interaktif pada halaman web.

Dalam konteks implementasi *Game* Semaphore, p5.js dapat digunakan untuk membuat elemen visual yang menarik dan interaktif pada halaman web. Misalnya, p5.js dapat digunakan untuk menampilkan grafis atau animasi yang memperkaya pengalaman bermain permainan Semaphore, menciptakan atmosfer yang lebih menarik dan menyenangkan.

ml5.js adalah perpustakaan *machine learning* yang dibangun di atas TensorFlow.js, dirancang untuk membuat *machine learning* lebih mudah

diakses bagi pengembang kreatif. Ini menyediakan model yang telah dilatih sebelumnya dan alat untuk mengimplementasikan fungsionalitas *machine learning* dalam aplikasi web dengan mudah. *ml5.js* memungkinkan pengembang mengintegrasikan fitur *machine learning*, termasuk klasifikasi gambar, deteksi objek, dan pemrosesan bahasa alami, ke dalam proyek mereka tanpa memerlukan pemahaman mendalam tentang algoritma *machine learning*.

ml5.js dapat dimanfaatkan untuk memperkaya fungsionalitas *Game Semaphore* dengan mengintegrasikan model *machine learning* tambahan. Contohnya, *ml5.js* dapat digunakan untuk memberikan umpan balik atau skor berdasarkan gerakan tubuh pengguna, menciptakan pengalaman permainan yang lebih adaptif dan dinamis[15].

Dengan demikian, integrasi HTML, CSS, JavaScript, PoseNet, *p5.js*, dan *ml5.js* dalam *Game Semaphore* memberikan dasar yang kokoh untuk menciptakan pengalaman permainan yang responsif, interaktif, dan menarik berdasarkan gerakan tubuh pengguna.

3. HASIL DAN PEMBAHASAN

Metode yang digunakan pada sistem game dapat dijelaskan pada gambar berikut:



Gambar 8. Diagram pemetaan file pada sistem game semaphore

Pemetaan file pada sistem game Semaphore bergantung pada arsitektur dan struktur direktori yang digunakan dalam pengembangan perangkat lunak tersebut.

Struktur dasar direktori terdiri atas folder images yang berisi svg huruf-huruf yang digunakan untuk pembuatan kamus semaphore dan svg icon-icon yang digunakan untuk button, logo, dll., folder scripts yang berisi file skrip atau source code game, dan folder styles yang berisi file untuk menghias file html atau stylesheet.

Terdapat 2 file inti HTML, yang kita gunakan untuk launch game adalah

3.1 State Machine Diagram

1. State: Waiting for Start

Deskripsi: Aplikasi berada dalam keadaan ini saat pertama kali dibuka.

Transisi: Jika pengguna memulai permainan, beralih ke "State: Waving Tutorial".

2. State: Waving Tutorial

Deskripsi: Pengguna diberikan tutorial tentang cara memulai permainan dengan gerakan tangan.

Transisi: Jika pengguna berhasil menyelesaikan tutorial, beralih ke "State: Game On".

3. State: Game On

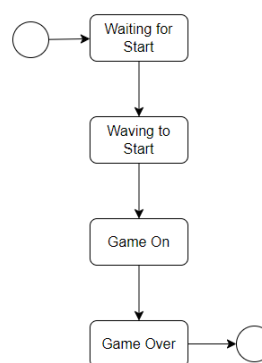
Deskripsi: Permainan aktif, dan sistem mendeteksi gerakan tangan pengguna menggunakan PoseNet.

Transisi: Jika waktu habis atau kondisi tertentu terpenuhi, beralih ke "State: Game Over".

4. State: Game Over

Deskripsi: Menampilkan skor akhir dan memberikan opsi untuk memulai kembali atau keluar dari permainan.

Transisi: Jika pengguna memilih untuk memulai kembali, kembali ke "State: Game Initialization". Jika pengguna memilih keluar, kembali ke "State: Waving".



Gambar 9. State Machine Diagram

3.2 Pemanggilan Library PoseNet

```

57 function setup() {
58   createCanvas(windowHeight * 1.4, windowHeight * 0.8);
59   video = createCapture(VIDEO);
60   videoWidth = width;
61   videoHeight = width * 0.75;
62   video.size(videoWidth, videoHeight);
63   video.hide();
64   const poseNet = ml5.poseNet(video, modelLoaded);
65   poseNet.on("pose", gotPoses);
66
67   //set count down interval to 1 second
68   setInterval(timer, 1000);
69 }

```

Gambar 10. Pemanggilan Library PoseNet di sketch.js

Pemanggilan *library* PoseNet berada di Fungsi 'setup()'. Fungsi ini dipanggil sekali saat aplikasi dimuat. Ini membuat kanvas untuk menampilkan video dari *webcam* pengguna dan menginisialisasi model PoseNet untuk mengenali gerakan tubuh.

Implementasi PoseNet dimulai dengan pembuatan layar visual berukuran tertentu. Selanjutnya, kamera diaktifkan untuk mengambil gambar, dan ukuran gambar disesuaikan agar sesuai dengan layar yang telah dibuat sebelumnya. Gambar dari kamera disembunyikan, dan PoseNet digunakan untuk mengenali gerakan atau pose tubuh dari gambar tersebut. Ketika perubahan pose terdeteksi di depan kamera, fungsi khusus dipanggil untuk menangani situasi tersebut. Dengan demikian, melalui langkah-langkah tersebut, implementasi PoseNet memungkinkan pemahaman dan respons terhadap gerakan atau pose yang terjadi di depan kamera.

Dengan implementasi ini, PoseNet akan terus memantau video dari kamera pengguna dan

mengirimkan data pose manusia ke fungsi `gotPoses` ketika ada deteksi pose yang terjadi. Fungsi `gotPoses` kemudian dapat digunakan untuk mengakses dan memproses data pose tersebut, misalnya untuk mengenali gerakan atau tindakan berdasarkan pose yang dikenali

3.3 Pembuatan Kamus

Kode ini merupakan bagian dari program yang berinteraksi dengan *webcam* dan mengenali gerakan pengguna untuk sebuah aplikasi permainan atau aktivitas lainnya.

Terdapat *array* `posX` dan `posY` yang berfungsi sebagai wadah untuk menyimpan koordinat X dan Y dari titik-titik kunci. Biasanya, titik-titik ini mewakili posisi tangan dan siku pengguna, yang penting untuk mengenali berbagai gerakan tubuh.

Selanjutnya, *array* `lerpPosX` dan `lerpPosY` digunakan untuk menciptakan efek pelunakan atau smoothing dalam pergerakan. Hal ini bertujuan agar pergerakan tangan pengguna terasa lebih halus dan tidak bersifat tiba-tiba.

Terdapat juga variabel *boolean*, seperti `gotPose`, yang menandakan apakah sistem telah berhasil mengenali gerakan tubuh pengguna. Ketika nilai variabel ini menjadi *true*, itu menunjukkan bahwa ada gerakan yang telah dikenali.

Selain itu, variabel `gameStart` menandakan apakah permainan sudah dimulai. Nilai *true* akan diberikan ketika pengguna memulai permainan dengan gerakan yang sesuai.

Beberapa variabel lainnya, seperti `timerValue`, `previousLetter`, `wave`, dan `numCorrect`, memiliki peran khusus dalam mengelola waktu permainan, melacak huruf sebelumnya, menghitung jumlah gerakan tangan, dan mencatat jawaban yang benar dari pengguna.

Akhirnya, terdapat variabel `videoWidth` dan `videoHeight` yang mengatur dimensi video yang diambil dari *webcam* pengguna. Semua elemen ini bekerja bersama-sama untuk menciptakan pengalaman interaktif berbasis gerakan dalam aplikasi tersebut.

```
22 wave_angle = {
23     up: [-180, 0],
24     down: [-135, -45],
25 };
```

Gambar 11. Potongan code bagian `wave_angle`

3.4 Proses Pengenalan Postur Semaphore

Proses pengenalan postur Semaphore melibatkan beberapa langkah utama menggunakan teknologi PoseNet. Berikut adalah tahapan umumnya:

```
28 letter_angle = {
29     A: [-90, -45],
30     B: [-90, 0],
31     C: [-90, 45],
32     D: [-90, 90],
33     E: [135, -90],
34     F: [-180, -90],
35     G: [-135, -90],
36     H: [-45, 0],
37     I: [-45, 45],
38     J: [-180, 90],
39     K: [90, -45],
40     L: [135, -45],
41     M: [-180, -45],
42     N: [-135, -45],
43     O: [45, 0],
44     P: [90, 0],
45     Q: [135, 0],
46     R: [-180, 0],
47     S: [-135, 0],
48     T: [90, 45],
49     U: [135, 45],
50     V: [-135, 90],
51     W: [-180, 135],
52     X: [-135, 135],
53     Y: [-180, 45],
54     Z: [-180, -135],
55 };
```

Gambar 12. Potongan code bagian `letter_angle`

a. Inisialisasi Model PoseNet:

Aplikasi akan memanggil *library* atau model PoseNet untuk memulai proses pengenalan postur. Inisialisasi ini mencakup memuat model PoseNet yang telah dilatih sebelumnya.

b. Perekaman Video atau Data Gambar:

Aplikasi akan mengakses kamera pengguna atau menggunakan data gambar/video yang disediakan untuk merekam gerakan tubuh.

c. Deteksi Pose Manusia:

PoseNet akan mendeteksi pose manusia dalam setiap *frame* gambar atau video yang direkam. Model ini akan mengidentifikasi lokasi 17 titik kunci pada tubuh manusia, seperti kepala, bahu, siku, pergelangan tangan, pinggul, lutut, dan pergelangan kaki.

d. Analisis Postur Semaphore:

Setelah mendeteksi pose manusia, aplikasi akan melakukan analisis terhadap posisi dan hubungan antar titik kunci tersebut. Posisi spesifik, seperti posisi tangan atau lengan, akan digunakan untuk mengidentifikasi gerakan tubuh yang sesuai dengan postur Semaphore.

e. Koreksi Gerakan Tubuh:

Jika diperlukan, aplikasi dapat memberikan umpan balik langsung kepada pengguna terkait akurasi gerakan Semaphore mereka. Misalnya, jika pose yang terdeteksi tidak sesuai dengan gestur yang benar, aplikasi dapat memberikan koreksi visual atau instruksi[4].

f. Interaksi dengan *Game* Semaphore:

PoseNet akan memproses data pose manusia dalam konteks *game* Semaphore. Gerakan tubuh pengguna yang telah diidentifikasi dan dikoreksi akan diintegrasikan dalam logika permainan. Misalnya, pengguna mungkin diharuskan melakukan gestur tertentu untuk berkomunikasi dalam *game* Semaphore.

g. Pemberian Skor atau *Feedback*:

Aplikasi dapat memberikan skor atau umpan balik kepada pengguna berdasarkan seberapa baik mereka meniru gerakan Semaphore yang benar. Skor ini dapat mencerminkan tingkat ketepatan gerakan dan memberikan pengguna insentif untuk meningkatkan keterampilan mereka.

h. Pembaruan *Real-Time*:

Proses pengenalan postur Semaphore ini berlangsung secara *real-time*, yang berarti aplikasi terus memproses pose manusia dan memberikan respons dalam waktu nyata selama pengguna berinteraksi dengan permainan.

Penerapan PoseNet ini memerlukan pemahaman mendalam tentang API dan dokumentasi resmi yang disediakan oleh penyedia teknologi tersebut.

3.5 Proses Menghitung Score

```
let numCorrect = 0;
```

Gambar 13. Inisialisasi Variabel Skor

Variabel *numCorrect* digunakan untuk menyimpan jumlah gerakan Semaphore yang benar yang telah dikenali oleh pengguna.

```
if (gotPose == true) {
  drawDots();
}
waveStart(3);
if (gameStart == true) {
  countdown();
  initDom();

  if (
    verifyAngle(getUserAngle(), letter_angle[randomLetter], 10) &&
    timerValue > 0
  ) {
    numCorrect++; //increase score count
    previousLetter = randomLetter;
    randomLetter = getRandomLetter(letter_angle);

    while (randomLetter == previousLetter) {
      randomLetter = getRandomLetter(letter_angle);
    }
    gameDom();
  } //end the game once the timer runs to 0
  else if (gameStart == false) {
    handleEnd();
  }
}
```

Gambar 14. Mengecek gerakan yang benar dan perbarui skor

Pada kondisi di atas, *verifyAngle* digunakan untuk memeriksa apakah gerakan tubuh pengguna sesuai dengan gerakan Semaphore yang ditentukan (*letter_angle[randomLetter]*). Jika gerakan benar dan waktu masih tersisa (*timerValue > 0*), maka *numCorrect* akan ditingkatkan, dan gerakan baru akan dipilih untuk ditampilkan. Fungsi *gameDom()* dipanggil untuk mengupdate tampilan dan skor pada antarmuka pengguna.

```
function handleEnd() {
  endArrow = select(".start-end_img");
  endArrow.addClass("start-end_img-move");

  uiScore = select(".score");
  uiTimer = select(".countdown");

  uiScore.addClass("score-end");
  uiTimer.addClass("countdown-end");

  uiInfo = select(".info");
  uiInfo.addClass("info-end");

  endCard = select(".hint-end");
  endCard.addClass("hint-end-mid");

  finalScore = select(".hint_final-score");
  if (numCorrect < 10) {
    finalScore.html("0" + numCorrect);
  }
  if (numCorrect >= 10) {
    finalScore.html(numCorrect);
  }
}
```

Gambar 15. Menampilkan skor akhir

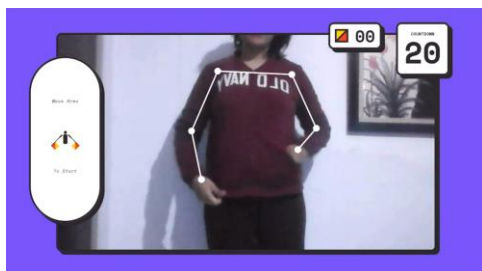
Ketika permainan berakhir, skor akhir ditampilkan pada elemen HTML dengan kelas *.hint_final-score*.

Dengan demikian, setiap kali pengguna berhasil melakukan gerakan Semaphore yang benar, skor akan ditingkatkan, dan tampilan skor akan diperbarui sesuai. Skor akhir kemudian ditampilkan ketika permainan berakhir.

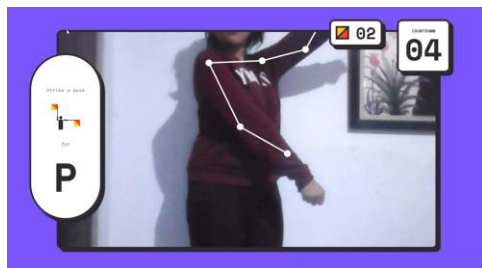
3.6 Pengujian dan Evaluasi Akhir



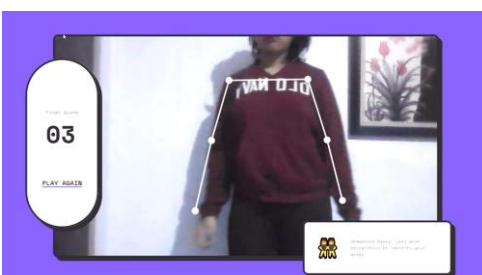
Gambar 16. Menu Awal



Gambar 17. Wave to Start



Gambar 18. Tampilan in game



Gambar 19. Tampilan Game Over

4. KESIMPULAN

Dalam penelitian ini, saya telah mengembangkan sebuah aplikasi permainan interaktif yang menggunakan teknologi deteksi gerakan tubuh melalui kamera *webcam*. Aplikasi ini bertujuan untuk memberikan pengalaman bermain yang unik dan melibatkan pengguna dalam permainan dengan cara yang interaktif. Hasil penelitian dan pengujian aplikasi kami dapat disimpulkan sebagai berikut:

1. Aplikasi telah berhasil dikembangkan dan memungkinkan pengguna untuk bermain dengan cara mengikuti petunjuk gerakan tubuh.
2. Pengujian aplikasi menunjukkan bahwa fungsionalitas permainan bekerja dengan baik, dan pengguna dapat dengan mudah memahami aturan dan panduan yang disediakan.
3. Algoritma pengenalan gerakan tubuh dalam waktu nyata telah mampu mendeteksi gerakan dengan akurasi yang memadai.
4. Aplikasi memberikan pengalaman bermain yang responsif dan menyenangkan.

DAFTAR PUSTAKA

- [1] "Teach-AI-to-Dance-with-PoseNet," GitHub, 2023. [Online]. Available: <https://github.com/znreza/Teach-AI-to-Dance-with-PoseNet>
- [2] "Awesome Dancing with AI Tutorial," GitHub, 2023. [Online]. Available:

<https://github.com/dancingwithai/dancingwithai.github.io>

- [3] "google-coral/project-posenet," GitHub, 2023. [Online]. Available: <https://github.com/google-coral/project-posenet>
- [4] "Human Pose Classification with MoveNet and TensorFlow Lite," TensorFlow, 2023. https://www.tensorflow.org/lite/tutorials/pose_classification
- [5] Bhosale, Pranjal, & Bale. "Yoga Pose Detection and Correction using Posenet and KNN," *Research Gate*. [Online]. Available: https://www.researchgate.net/publication/360950640_Yoga_Pose_Detection_and_Correction_using_Posenet_and_KNN
- [6] D. Shah, V. Rautela, C. Sharma and A. Florence A, "Yoga Pose Detection Using Posenet and k-NN," 2021 International Conference on Computing, Communication and Green Engineering (CCGE), Pune, India, 2021, pp. 1-4, doi: 10.1109/CCGE50943.2021.9776451. [Online]. Available: <https://ieeexplore.ieee.org/document/9776451>
- [7] "Yoga Pose Estimation using POSENET vision model," GitHub, 2023. [Online]. Available: https://github.com/Anjanapradeep/POSE_ESTIMATION-YOGA-POSES
- [8] "PoseNet: Revolutionizing Human Pose Estimation with Deep Learning," Medium, 2023. [Online]. Available: <https://medium.com/aimonks/posenet-revolutionizing-human-pose-estimation-with-deep-learning-1eebc873966>
- [9] "Evaluation of PoseNet for applied AI-Fitness applications," Medium, 2023. [Online]. Available: <https://medium.com/optima-ai/evaluation-of-posenet-for-applied-ai-in-fitness-applications-56de98d6c4e4>
- [10] "Selecting Your Real-Time Pose Estimation Models," Medium, 2023. [Online]. Available: <https://maurentkt.medium.com/selecting-your-2d-real-time-pose-estimation-models-7d0777bf935f>
- [11] Davids Joe (2022). Artificial Intelligence for Physiotherapy and Rehabilitation *Research Gate*. [Online]. Available: https://link.springer.com/referenceworkentry/10.1007/978-3-030-64573-1_339
- [12] Chung, J. L., Ong L. Y., & Chew L. M. (2022). Comparative Analysis of Skeleton Based Human Pose Estimation, *Research Gate*. [Online]. Available: <https://www.mdpi.com/1999-5903/14/12/380>
- [13] "7 Popular AI Projects On Gesture Gaming," Analytics India Magazine, 2023. [Online] Available: <https://analyticsindiamag.com/7-popular-ai-projects-on-gesture-gaming/>
- [14] "Body movement recognition in the 'Smart Baduanjin' App," Medium, 2023. [Online]. Available: <https://medium.com/tensorflow/body-movement-recognition-in-the-smart-baduanjin-app-2a4e2d5159c8>
- [15] "Body Detection using Computer Vision," Medium, 2023. [Online]. Available: <https://medium.com/instrument-stories/body-detection-with-computer-vision-1898cdc6b7d>
- [16] Syahrul, A. M. ., & Rahayu, M. I. (2019). Aplikasi Game "Semaphore" Berbasis Android. *Jurnal Teknologi Informasi Dan Komunikasi*, 8(1), 1–10.

[Online]. Available:
<https://doi.org/10.58761/juristikstmikbandung.v8i1.121>

SpringerLink, pp. 125-149, Sep. 2020. [Online].
Available: https://doi.org/10.1007/978-1-4842-6168-7_6

- [17] J. D. Santos Rivera, "Making a game with PoseNet, a pose estimator model," in Practical TensorFlow.js,