

RANCANG BANGUN *MOTION ROBO MAKER APPS* BERBASIS WEBSITE SEBAGAI MEDIA PEMBELAJARAN ROBOTIKA PADA MAHASISWA PESERTA KONTES UNTUK MENGUKUR *CRITICAL THINKING* DAN *COLLABORATIVE SKILLS* (STUDI KASUS : KONTES ROBOT SENI TARI INDONESIA)

Zaskhia Artina Isnalifah

Pendidikan Teknologi Informasi, Fakultas Teknik, Universitas Negeri Surabaya

Email : zaskhia.20015@mhs.unesa.ac.id

Muhamad Syariffuddin Zuhrie

Pendidikan Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya

Email : zuhrie@unesa.ac.id

Abstrak

Penelitian ini mengembangkan aplikasi website bernama "*Motion Robo Maker Apps*" untuk memudahkan pembuatan gerakan robot tari *humanoid* dan menjadi media pembelajaran robotika untuk mengukur kemampuan *critical thinking* dan *collaborative skills* mahasiswa dalam kompetisi pada studi kasus Kontes Seni Tari Indonesia. Metode yang digunakan adalah *Research and Development* (R&D) dengan model *eXtreme Programming* (XP). Pengujian dilakukan dengan *Blackbox Testing* dan menunjukkan tingkat keberhasilan 100%. Pengujian pengiriman nilai *servo* dari aplikasi ke mikrokontroler menunjukkan *error* 3%, sedangkan pembacaan nilai *servo* dari aplikasi menunjukkan *error* 0%. Media ini divalidasi oleh 3 ahli dan memperoleh *rating* validitas 92% (Sangat Valid), komunikasi visual 87% (Valid), dan isi media 90% (Sangat Valid). Hasil respon mahasiswa terhadap media mendapat *rating* Sangat Valid dengan persentase 93% dari mahasiswa anggota tim Azzahrly Divisi KRSTI DEWO Robotik Fakultas Teknik Universitas Negeri Surabaya. Hasil tes menunjukkan nilai rata-rata 88 untuk kemampuan *critical thinking* mahasiswa dan skor rata-rata 79,11 untuk kemampuan *collaborative skills*. Uji normalitas menunjukkan data berdistribusi normal dengan nilai signifikan untuk *critical thinking* sebesar 0,089 dan *collaborative skills* sebesar 0,847. Kesimpulannya, aplikasi ini efektif dalam meningkatkan kemampuan berpikir kritis dan kolaboratif mahasiswa, serta mendukung mereka dalam kompetisi robotika. Aplikasi ini diharapkan dapat mempercepat dan memudahkan proses pembuatan gerakan robot tari *humanoid* serta mendukung inovasi dalam robotika dan seni.

Kata Kunci : *Motion Robo Maker Apps*, *Critical Thinking*, *Collaborative Skills*, Robotika, *eXtreme Programming* (XP)

Abstract

This research developed a website application called "*Motion Robo Maker Apps*" to make it easier to create *humanoid* dance robot movements and become a robotics learning medium to measure students' *critical thinking* and *collaborative skills* in competitions in the case study of the Indonesian Dance Arts Contest. The method used is *Research and Development* (R&D) with the *eXtreme Programming* (XP) model. Testing was carried out using *Blackbox Testing* and showed a success rate of 100%. Testing of sending *servo* values from the application to the microcontroller showed an *error* of 3%, while reading *servo* values from the application showed an *error* of 0%. This media was validated by 3 experts and obtained a validity *rating* of 92% (Very Valid), visual communication 87% (Valid), and media content 90% (Very Valid). The results of student responses to the media received a Very Valid *rating* with a percentage of 93% from students belonging to the Azzahrly team, KRSTI DEWO Robotics Division, Faculty of Engineering, Surabaya State University. The test results showed an average score of 88 for students' *critical thinking* abilities and an average score of 79.11 for *collaborative skills*. The normality test shows that the data is normally distributed with a significant value for *critical thinking* of 0.089 and *collaborative skills* of 0.847. In conclusion, this application is effective in improving students' *critical* and *collaborative thinking* skills, as well as supporting them in robotics competitions. This application is expected to speed up and simplify the process of creating *humanoid* dance robot movements and support innovation in robotics and art.

Keywords: *Motion Robo Maker Apps*, *Critical Thinking*, *Collaborative Skills*, Robotics, *eXtreme Programming* (XP).

PENDAHULUAN

Seiring dengan kebutuhan manusia yang semakin kompleks, teknologi informasi, khususnya robotika, berkembang dengan cepat. Robot telah menjadi alat penunjang untuk menyelesaikan pekerjaan di bidang industri, kesehatan, pertahanan, pendidikan, hiburan ataupun pekerjaan lainnya (Alasiry dkk., 2018). Banyak negara telah mengembangkan berbagai macam jenis robot, salah satunya adalah robot *humanoid* (Satria dkk., 2023). Robot *humanoid* merupakan robot yang menyerupai manusia dengan menampilkan bentuk manusia karakteristik manusia, atau meniru perilaku manusia (Blut dkk., 2023). Pada umumnya, robot *humanoid* dibangun dari serangkaian aktuator yang berupa *servo*, sensor, *mikrokontroller*, dan *bracket* atau penyangga sebagai penghubung antar *servo* (Somiseti dkk., 2020). Robot ini memiliki banyak manfaat dan dapat melakukan berbagai macam hal seperti olahraga, manajemen bencana dan kesenian (Blut dkk., 2023; Fahd dkk., 2018).

Para peneliti telah menggunakan robot *humanoid* untuk mengembangkan bidang kesenian. Dalam beberapa jurnal, topik ini secara mendalam, seperti penelitian yang dilakukan oleh S. Xu dan L. Zhou yang membahas terkait pembuatan robot tari *humanoid* sebagai upaya melestarikan budaya Tarian khas lokal Tiongkok, China yang terancam punah akibat dari perkembangan zaman (Xu & Zhou, 2021). Dalam penelitian mereka, robot *humanoid* dikombinasikan dengan teknologi internet untuk menciptakan inovasi berupa robot tari tiruan manusia yang dapat melakukan gerakan tari seperti manusia (Xu & Zhou, 2021). Seperti halnya negara lainnya, Indonesia juga dikenal sebagai negara yang kaya akan keanekaragaman budaya dan keseniannya salah satunya seni tari. Hampir semua wilayah di Indonesia memiliki tari mereka sendiri. Ini dapat menjadi peluang besar untuk melestarikan dan mempromosikan seni tari tradisional dengan memanfaatkan robot *humanoid* melalui kemajuan teknologi.

Pada penelitian yang dilakukan oleh Pratama dkk., (2016) yang berjudul “Penentuan Sudut Lengan Robot *Humanoid* Berdasarkan Koordinat Yang Dikirim Dari PC Menggunakan Pengguna *Interface* Yang Dibuat Dari Qt” yang mengembangkan aplikasi yang dapat menggerakkan motor DC *servo* pada lengan robot sesuai koordinat yang telah ditentukan. Aplikasi ini menggunakan metode Persamaan Invers Kinematik untuk menghitung perubahan sudut tiap sendi pada lengan robot. Perbedaan dengan referensi yang ada adalah posisi basis pada aplikasi ini berada di tanah, sehingga beberapa variabel perlu dimodifikasi. Masukan koordinat berasal dari PC melalui Pengguna *Interface* yang dibuat dengan Qt. Pengujian menunjukkan bahwa terdapat kesalahan pada gerakan

robot karena adanya *error* pada nilai karakterisasi masing-masing *servo*.

Pada penelitian yang dilakukan oleh (Satria dkk., 2023) yang berjudul “*The Development of Dance Movement in Humanoid Robot Dancing ERISA*” yang menggunakan menggunakan *software* Blender dalam menentukan dan membuat gerak tari dengan *output* berupa derajat tiap sendi yang kemudian diimplementasikan dan dibuktikan dengan memasukkan data setelah diubah menjadi data PWM untuk menggerakkan motor *servo* terdapat perbedaan yang mempunyai perbedaan sedikit atau besar dengan rata-rata *error* sebesar 3,25%. Dengan menggunakan *software* Blender dapat mempersingkat waktu dalam pembuatan gerak tari serta dapat langsung melihat hasil dan bentuk gerakan yang dibuat.

Di Indonesia penelitian dan pengembangan terkait robot *humanoid* masih terus berkembang. Terdapat banyak kegiatan yang dilakukan untuk mendukung perkembangan salah satunya adalah melalui Kontes Robot Indonesia (KRI) yang diselenggarakan oleh Balai Pengembangan Talenta Indonesia (BPTI) di Pusat Prestasi Nasional (Pusprenas) Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi Republik Indonesia (Kusumoputro dkk., 2023). Kegiatan ini terdiri dari beberapa divisi, salah satunya adalah Kontes Robot Seni Tari Indonesia (KRSTI). Kontes Robot Seni Tari Indonesia (KRSTI) merupakan ajang kompetensi serta pengembangan ide dalam perancangan, implementasi, strategi dan pemrograman robot *humanoid* yang disertai dengan unsur-unsur seni dan budaya bangsa Indonesia khususnya seni tari yang telah terkenal di bumi pertiwi (Kusumoputro dkk., 2023).

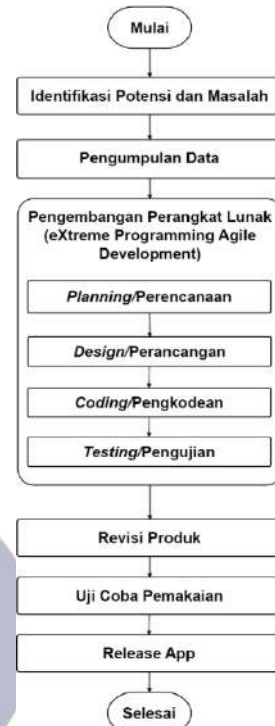
Setiap tahun, Kontes Robot Seni Tari Indonesia (KRSTI) memilih tema dan arena perlombaan yang berbeda (Satria dkk., 2023). Pada tahun 2023, divisi KRSTI memilih tema “Robot Penari Denok Semarang/Robot Penari Gambang Semarang” dari Jawa Tengah yang bertujuan untuk membangkitkan kembali kecintaan dan pelestarian budaya-budaya Nasional Indonesia (Kusumoputro dkk., 2023). Pada tahun kali ini, robot harus mampu melakukan gerakan tari Gambang Semarang mulai zona *Start* hingga zona *Finish* dengan mengikuti alunan musik (Kusumoputro dkk., 2023). Robot di program khusus agar dapat menyelesaikan misi tersebut untuk mendapatkan poin tertinggi. Namun untuk membuat sebuah gerakan tari pada robot membutuhkan waktu yang cukup lama untuk memperkirakan sudut *servo* yang digunakan. Dan kemudian *mikrokontroller* memproses nilai tersebut untuk mengendalikan motor DC *servo* di setiap sendi robot. Jika menggunakan metode ini, akan membutuhkan proses *trial* dan *error* berulang kali hingga menemukan gerakan tari yang diinginkan, dan ini dapat mempengaruhi umur motor *servo*.

Permasalahan tersebut sudah teratasi dengan penggunaan perangkat lunak RoboPlus 1.0. RoboPlus 1.0 merupakan perangkat lunak keluaran perusahaan pengembangan robotika yaitu ROBOTIS(Robotis, 1999a). Perangkat lunak didesain untuk digunakan pada desktop dan memungkinkan untuk dapat berinteraksi dan mengontrol semua perangkat keras ROBOTIS, *Dynamixel*, sensor, dan komponen perangkat keras lainnya(Robotis, 1999b). Namun, hal ini tidak akan bertahan lama karena ketersediaan komponen yang digunakan, seperti *mikrokontroller*, sensor, dan *servo* terbatas dan mahal. Akibatnya, kami berhenti menggunakan RoboPlus 1.0 dan kembali lagi ke metode yang lama. Pada proses peralihan ke metode yang lama sangat tidak efisien karena membutuhkan waktu yang lama dan perlu menyesuaikan nilai derajat *servo* satu-satu.

Oleh karena itu, dibutuhkan aplikasi yang dapat mengontrol gerakan motor DC *servo* pada robot tari *humanoid* sesuai dengan titik koordinat yang telah ditentukan. Aplikasi diharapkan dapat mengelola nilai sudut *servo*, seperti membaca, dan menyimpan nilai *servo* berdasarkan titik koordinatnya, serta pengguna dapat memasukkan dan menghapus nilai derajat *servo* sesuai keinginannya. Serta dalam proses implementasinya aplikasi dapat meningkatkan efektivitas proses pembuatan gerakan robot tari *humanoid* dan menjadi media untuk pembelajaran mahasiswa dalam merancang dan membangun robot seni tari *humanoid*. Pada penelitian ini, akan difokuskan pada "Rancang Bangun Motion Robo Maker Apps Berbasis Website Sebagai Media Pembelajaran Robotika pada Mahasiswa Peserta Kontes (Studi Kasus : Kontes Robot Seni Tari Indonesia)".

METODE

Penelitian ini menggunakan metode penelitian pengembangan atau *Research and Development (R&D)* untuk menghasilkan dan menguji keefektifan produk tersebut (Sugiyono, 2011). Sedangkan pengembangan aplikasi ini menggunakan model *eXtreme Programming (XP)* yang merupakan sebuah model dalam *Agile Software Development*, salah satu metodologi dalam siklus hidup pengembangan perangkat lunak *Software Development Life Cycle (SDLC)*(Suryantara, 2017). Model ini dinilai cepat, efisien, beresiko rendah, fleksibel, terprediksi, *scientific*, dan menyenangkan.



Gambar 1. Flowchart Alur Penelitian

Pengumpulan Data

- 1) Studi Lapangan: Melakukan pengamatan langsung atau observasi di Laboratorium Robotika Universitas Negeri Surabaya untuk mengumpulkan data atau fakta dan menggali kebutuhan *software* serta *user requirement*.
- 2) Kuisisioner (Angket): Menggunakan kuesioner untuk mengetahui penilaian ahli media dan respon mahasiswa mengenai Rancang Bangun Motion Robo Maker Apps Berbasis Website Sebagai Media Pembelajaran Robotika Pada Mahasiswa (Studi Kasus : Kontes Robot Seni Tari Indonesia).
- 3) Tes Hasil Belajar : Mengumpulkan data dengan memberi lembar soal pertanyaan kepada responden terkait materi.
- 4) Studi Literatur: Mengumpulkan data dan informasi melalui dokumen seperti buku, jurnal, *prosiding*, laporan, atau bentuk lain dalam bentuk tercetak maupun digital.

Metode yang digunakan untuk pengembangan perangkat lunak yaitu dengan menggunakan model *eXtreme Programming (XP)*, yang meliputi : **Planning/Perencanaan**, **Design/Perancangan**, **Coding/ Pengkodean**, dan **Testing/Pengujian**.

Planning/Perencanaan

Tahap pertama, yaitu perencanaan, dimulai dengan mengidentifikasi masalah pembuatan gerakan tari pada robot yang memakan waktu lama karena belum ada aplikasi pengelolaan data *servo*. Selanjutnya, disusun beberapa fitur penting yang perlu dimasukkan ke dalam sistem.

- 1) Analisis Kebutuhan Fungsional: Analisa kebutuhan fungsional terkait dengan proses yang akan berjalan pada sistem aplikasi *Motion Robo Maker Apps* antara lain :

Tabel 1. Analisa Kebutuhan Fungsional

Kode	Kategori	Rincian Kebutuhan
F-01	<i>Dashboard</i>	Sistem menampilkan halaman utama yang berisi daftar proyek yang dibuat.
F-02	Membuat <i>Project</i>	Sistem memungkinkan <i>user</i> untuk membuat proyek baru dengan mengisi <i>form</i> yang telah disediakan.
F-03	Mengelola <i>Project</i>	Sistem memungkinkan <i>user</i> untuk menyimpan, melihat, mengubah, mengunduh dan menghapus proyek.
F-04	Membuat <i>Motion</i>	Sistem memungkinkan <i>user</i> untuk membuat <i>motion</i> baru dengan mengisi <i>form</i> yang telah disediakan.
F-05	Mengelola <i>Motion</i>	Sistem memungkinkan <i>user</i> untuk menyimpan, mengubah, dan menghapus <i>motion</i> .
F-06	Mengelola <i>Step Motion</i>	Sistem memungkinkan <i>user</i> untuk menambah, mengubah, dan menghapus <i>step motion</i> .
F-07	Mengelola Nilai ID <i>Servo</i>	Sistem memungkinkan <i>user</i> untuk menambah, mengubah dan menghapus Nilai ID <i>Servo</i> .
F-08	Mengelola <i>Servos ID Group</i>	Sistem memungkinkan <i>user</i> untuk menambah dan menghapus <i>Servos ID Group</i> .
F-09	<i>Play Motion</i>	Sistem memungkinkan <i>user</i> untuk menjalankan <i>motion</i> yang dibuat
F-10	<i>Stop Motion</i>	Sistem memungkinkan <i>user</i> untuk menghentikan <i>motion</i> yang dijalankan
F-11	Koneksi dengan Robot	Sistem memungkinkan <i>user</i> untuk mengkoneksikan aplikasi dengan robot.
F-12	<i>Import Project</i>	Sistem memungkinkan <i>user</i> untuk mengelola <i>file</i> dari luar aplikasi

- 2) Analisis Kebutuhan Non-Fungsional: Analisis kebutuhan non-fungsional mencakup perangkat keras (*hardware*), perangkat lunak (*software*), dan batasan dan karakteristik sistem. Spesifikasi komputer yang diperlukan berikut:

Tabel 2. Analisis Kebutuhan Perangkat Keras

Perangkat Keras (Hardware)	
<i>Processor</i>	Intel Core i5-113G7 Gen 11
<i>Memory</i>	16GB DDR4
Penyimpanan	512GB SSD
Sistem Operasi	Windows 11 Home 64-bit

Tabel 3. Analisis Kebutuhan Perangkat Lunak

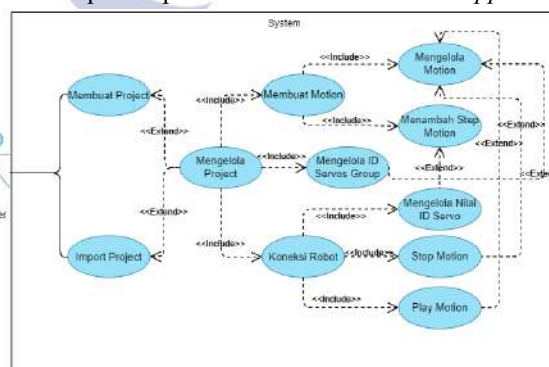
Perangkat Lunak (Software)	
Text Editor	Visual Studio Code
Web Browser	Chrome, Microsoft Edge
Bahasa	Javascript, library React
Pemograman	Js, Framework Tailwind CSS, Daisy UI
Penyimpanan Data	Cloud Storage Firebase

- 3) Perencanaan Fitur: Pada tahap ini, peneliti menentukan beberapa fitur minimum yang diperlukan untuk sistem agar pengembangan aplikasi lebih cepat. Beberapa fitur yang diperlukan termasuk: *Dashboard*, *New Project*, *Import Project*, *New Motion*, *Add Motion*, *Delete Motion*, *Save Project*, *Play Motion*, *Stop Motion*, *Connect Robot* dan *Download Project*.

Design/Perancangan

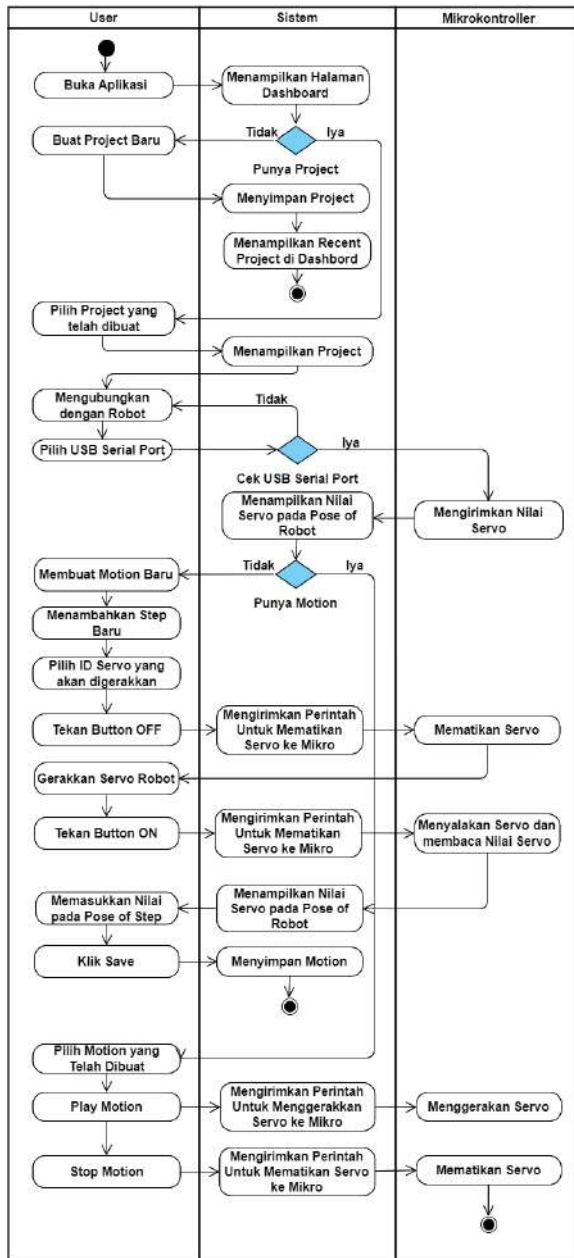
Tahap kedua, yaitu perancangan, melibatkan desain sistem agar berfungsi dengan baik dan memenuhi fitur yang direncanakan. Ini mencakup *Use Case*, *Activity Diagram*, *Flowchart*, Skema Pengembangan Aplikasi, Skema *Storage*, dan *Wireframe* untuk mempermudah pembuatan sistem.

- 1) *Use Case*: Diagram ini menggambarkan interaksi yang terjadi antara sistem dan pengguna. Berikut merupakan *Use Case* pada Aplikasi *Motion Robo Maker Apps*.



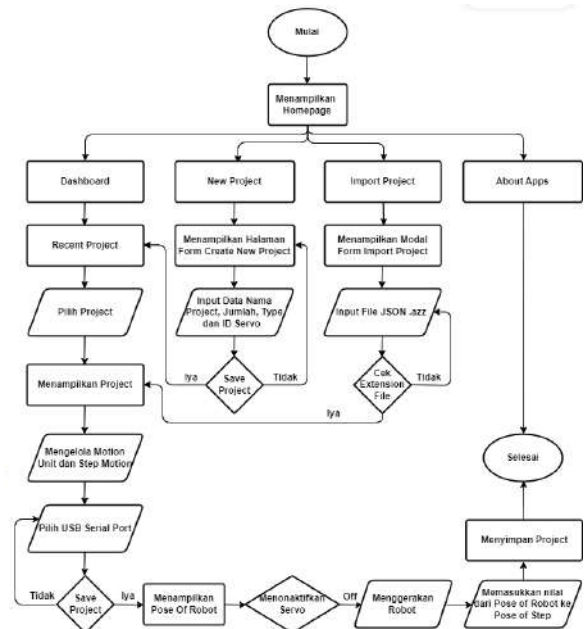
Gambar 2. Use Case Sistem

- 2) *Activity Diagram Sistem*: Menggambarkan alur kerja sistem masing-masing aktor, tugas yang dilakukan, dan kapan tugas dilakukan. Pada sistem ini terdapat 3 aktor yaitu *User* (Pengguna), *Sistem* dan *Mikrokontroler*. Setiap aktor mempunyai tugasnya masing-masing.



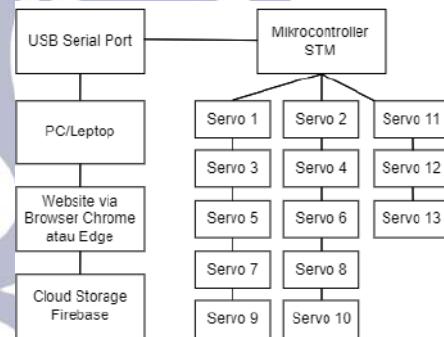
Gambar 3. Activity Diagram Sistem

- 3) *Flowchart Diagram*: Untuk memudahkan pengguna memahami alur aplikasi maka alur dan proses diantaranya. Tampilan dari sistem menampilkan beberapa menu, pada pilihan menu ada 4 bagian penting yaitu menu *Dashboard*, *Create Project*, *Import Project* *About Apps* dan *User's Guide*.



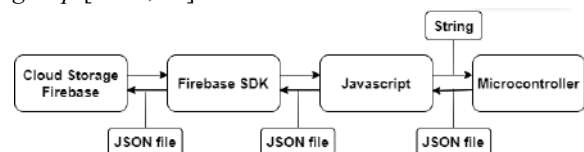
Gambar 4. Flowchart Sistem

- 4) *Skema Pengembangan Aplikasi*: Menggambarkan interaksi *Software* dan *Hardware* yang digunakan dalam aplikasi *Motion Robo Maker Apps* berbasis *website*.



Gambar 5. Skema Pengembangan Aplikasi

- 5) *Skema Storage* : Menggambarkan struktur alur data dari mikrokontroller yang disimpan di layanan penyimpanan *Cloud Storage Firebase*. Data yang disimpan pada *Cloud Storage Firebase* adalah berupa data json yang didalamnya terdapat data *Array* meliputi data *servo* [ID, Type Servo], data *motion* [name, step[], next], data *step* [time, pause, value] dan data *servo ID group* [name, ids].



Gambar 6. Skema Storage

- 6) *Wireframe* : penyusunan desain antarmuka pengguna (UI) menggunakan *wireframe*. Proses ini bertujuan menetapkan tampilan visual yang *user-friendly* dan memadai untuk memenuhi kebutuhan pengguna akhir aplikasi.

Coding/ Pengkodean

Pada tahap ketiga, yaitu tahap pengkodean, sistem yang telah dirancang diubah menjadi kode program. Pengembangan dilakukan menggunakan Visual Studio Code, dengan bahasa pemrograman JavaScript, *library* ReactJS, dan *framework* Tailwind CSS. Untuk penyimpanan data, digunakan layanan Cloud Storage Firebase. Penggunaan perangkat dan lingkungan ini membuat proses pengembangan aplikasi lebih efisien dan terorganisir.

Testing/Pengujian

Pada tahap keempat, sistem diuji menggunakan metode *blackbox testing* untuk menghindari kesalahan atau kegagalan tanpa memerlukan pemahaman tentang struktur kode. Fokusnya adalah pada hasil masukan, antarmuka pengguna, dan keluaran aplikasi. Pengujian ini dilakukan untuk memastikan sistem berfungsi sesuai tujuan, termasuk pengujian pembacaan dan pengiriman nilai *servo* oleh aplikasi ke mikrokontroler untuk menghitung nilai *error* menggunakan rumus tertentu. Untuk menghitung nilai *error* maka digunakan rumus dibawah ini :

$$\varepsilon = \left| \frac{Nm - Nk}{Nm} \right| \times 100\% \quad (1)$$

Keterangan :

Nm : Nilai Masukan

Nk : Nilai Keluaran

ε : Nilai Error

Setelah media selesai dalam bentuk produk jadi, dilakukan peninjauan oleh dosen pembimbing sebelum dilakukan validasi oleh ahli media. Proses validasi dilakukan bertujuan untuk menilai kelayakan dan menerima saran peningkatan kualitas produk sebelum diuji coba pada mahasiswa.

Release Apps

Tahap kelima dalam siklus pengembangan aplikasi melibatkan *hosting* aplikasi pada *server*, sehingga dapat diakses oleh pengguna melalui *browser web* seperti Google Chrome dan Microsoft Edge pada Windows. Tahap ini penting untuk memungkinkan penggunaan aplikasi dan mencakup pemantauan serta pemeliharaan untuk memastikan aplikasi berfungsi dengan baik. Selanjutnya, aplikasi diuji coba oleh mahasiswa tim Azzaharaly Divisi KRSTI DEWO Robotik di Fakultas Teknik Universitas Negeri Surabaya untuk mengevaluasi respon pengguna dan menguji kelayakan produk.

Populasi dan Sampel

Sugiyono (2019:126) menjelaskan bahwa konsep populasi dalam konteks penelitian merujuk pada jumlah dan karakteristik tertentu yang telah ditetapkan oleh

peneliti untuk dianalisis. Dalam penelitian ini, sampel penelitian terdiri dari 20 mahasiswa anggota tim Azzaharaly Divisi KRSTI DEWO Robotik Fakultas Teknik Universitas Negeri Surabaya.

Teknik Analisis Data

1) Analisis Validasi dan Respon Mahasiswa

Penilaian dari instrumen validasi pada penelitian ini menggunakan skala Likert dalam penilaiannya. Skala Likert memiliki rentang antara 1 sampai 5.

Tabel 4. Penilaian Lembar Validasi dan Respon Mahasiswa

Kriteria	Nilai
Sangat valid	5
Valid	4
Cukup Valid	3
Tidak valid	2
Sangat tidak valid	1

Berikut merupakan gambaran dari perhitungan jumlah jawaban validator. (Sumber: Sugiyono (2015:95).

$$\begin{aligned} \text{Jumlah skor "Sangat Valid"} &: n \times 5 \\ \text{Jumlah skor "Valid"} &: n \times 4 \\ \text{Jumlah skor "Cukup Valid"} &: n \times 3 \\ \text{Jumlah Skor "Tidak Valid"} &: n \times 2 \\ \text{Jumlah Skor "Sangat Tidak Valid"} &: n \times 1 + \\ \text{Jumlah total jawaban} &: \dots \end{aligned} \quad (2)$$

Dimana, nilai n merupakan jumlah banyaknya validator yang menjawab. Jika nilai jumlah validator sudah diketahui, maka selanjutnya adalah menentukan skor rating dengan rumus sebagai berikut.

$$\text{Hasil Rating} = \frac{\text{Jumlah total jawaban}}{\text{Jumlah jawaban maksimum}} \times 100\% \quad (3)$$

Selanjutnya dilakukan konversi hasil dalam bentuk menentukan interval terbawahnya yaitu sebagai berikut:

$$\text{Rating} = \frac{1}{5} \times 100\% = 20\% \quad (4)$$

Selanjutnya, menentukan panjang interval dari kriteria penilaian yang ada seperti berikut Sumber : (Widoyoko, 2017):

$$\frac{\text{Rating tertinggi} - \text{Rating terendah}}{\text{Jumlah kriteria}} = \frac{100 - 20}{5} = 16 \quad (5)$$

Dari hasil perhitungan diatas didapatkan kriteria penilaian sebagai berikut Sumber : (Widoyoko, 2017):

Tabel 5. Penilaian Kriteria Validasi dan Respon Mahasiswa

Kriteria	Interpretasi (%)
Sangat valid	88 – 100
Valid	71 – 87
Cukup Valid	54 - 70
Tidak valid	37 – 53
Sangat tidak valid	20 - 36

2) Analisis Hasil Belajar Mahasiswa Kemampuan *Critical Thinking* dan *Collaborative Skills*

Desain eksperimen *One Shot Case Study* ini digunakan untuk mengukur dampak pengembangan media pada mahasiswa anggota tim Azzaharly Divisi KRSTI DEWO Robotik Fakultas Teknik Universitas Negeri Surabaya. Uji normalitas digunakan untuk menentukan apakah populasi data berdistribusi normal atau tidak. Jika sampel lebih dari 50, digunakan rumus *Kolmogorov-Smirnov*, sedangkan jika sampel kurang dari 50, digunakan uji *Shapiro-Wilk*. Interpretasi dari uji normalitas adalah untuk mengetahui apakah data memiliki distribusi normal atau tidak.:

- Jika nilai signifikansi $\geq 0,05$ maka dapat disimpulkan bahwa data normal.
- Jika nilai signifikansi $< 0,05$ maka dapat disimpulkan bahwa data tidak normal.

Uji homogenitas adalah pengujian data yang dilakukan peneliti untuk menentukan apakah kedua data yang diteliti berasal dari populasi yang homogen (sama) atau berbeda.

- Jika nilai signifikansi $> 0,05$ maka data homogen (sama)
- Jika nilai signifikansi $< 0,05$ maka data tidak homogen (tidak sama)

Setelah pengujian normalitas yang menghasilkan data distribusi normal, langkah selanjutnya adalah uji hipotesis menggunakan *One Sample T-Test* untuk mengetahui hasil signifikan pada hipotesis. Jika hasilnya tidak signifikan, maka dilakukan uji nonparametrik menggunakan uji *Wilcoxon signed ranks test* (Miraza dkk., 2018).

HASIL DAN PEMBAHASAN

Implementasi GUI (Graphic User Interface)

Penelitian ini menghasilkan sistem aplikasi *Motion Robo Maker Apps*. Aplikasi yang dikembangkan pada penelitian ini berbasis *website*. Berikut merupakan tampilan GUI (*Graphical User Interface*) pada sistem :



Gambar 7. Tampilan Halaman Utama Website

Pada Gambar 7. merupakan halaman utama *Motion Robo Maker Apps* menampilkan logo atau brand aplikasi dan sidebar dengan menu *Dashboard*, *New Project*, *Import*

Project, dan *About Apps*. Dan menampilkan daftar proyek yang dibuat oleh pengguna.



Gambar 8. Tampilan Halaman *New Project*

Pada Gambar 8. merupakan halaman *New Project* dari *Motion Robo Maker Apps* menampilkan *form* untuk memasukkan nama proyek, jumlah *servo* yang akan digunakan, serta memilih *type* dan *ID* untuk setiap *servo*.



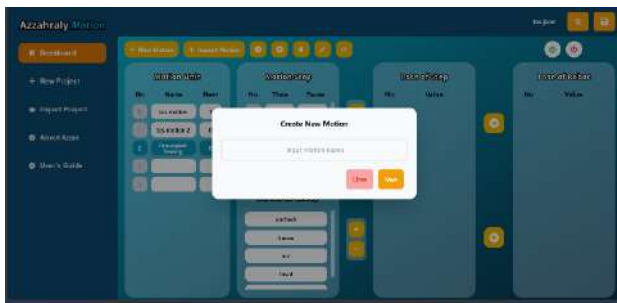
Gambar 9. Tampilan Halaman *Project*

Pada Gambar 9. merupakan halaman *Project* menampilkan beberapa komponen, termasuk keterangan nama proyek, tombol *Connect USB Serial Port*, dan *Save Project* di *header*. Di *body content*, terdapat tombol seperti *New Motion*, *Import Motion*, *Stop*, *Play*, *Delete*, *Edit*, dan *Generate Code*. Selain itu, ada *form* *Motion Unit*, *Motion Step*, *ID Servo Group*, *Pose of Step*, dan *Pose of Robot*.



Gambar 10. Tampilan *Motion Unit*

Gambar 10. menunjukkan tampilan *Motion Unit*, yang menampilkan daftar *motion* yang telah dibuat. Pengguna dapat mengelola setiap *motion*, termasuk mengubah nama, menghapus, mengimpor *motion*, dan menghasilkan kode.



Gambar 11. Tampilan Form Create New Motion

Pada Gambar 11. merupakan tampilan Form Create new Motion. Fitur ini digunakan untuk membuat motion baru.



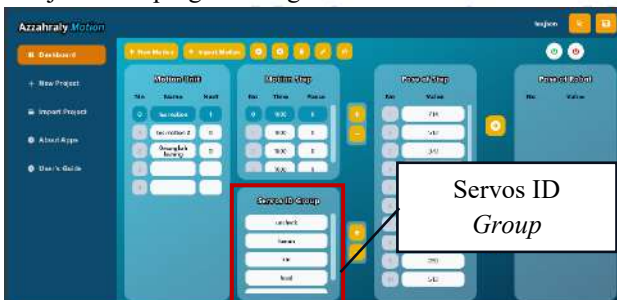
Gambar 12. Tampilan Form Change Motion Name

Pada Gambar 12. merupakan tampilan Form Change Motion Name. Fitur ini digunakan untuk mengubah nama motion yang sudah ada.



Gambar 13. Tampilan Menu Import Motion

Gambar 13. merupakan tampilan menu Import Motion, yang memungkinkan pengguna mengimpor motion berupa kode dengan tombol Import Motion dan mengubah motion menjadi kode program dengan tombol Generate Code.



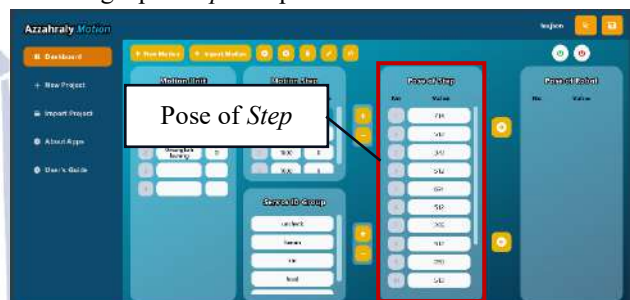
Gambar 14. Tampilan Servo ID Group

Pada Gambar 14. merupakan hasil tampilan Servos ID Group. Memungkinkan pengguna menambah dan menghapus grup yang sudah ada.



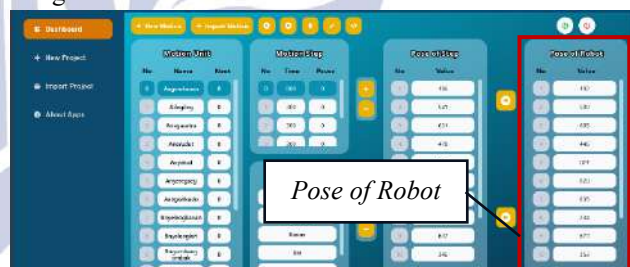
Gambar 15. Tampilan Menu Motion Step

Gambar 15. merupakan tampilan Motion Step. Pengguna dapat mengatur time dan pause, serta menambah dan menghapus step setiap motion.



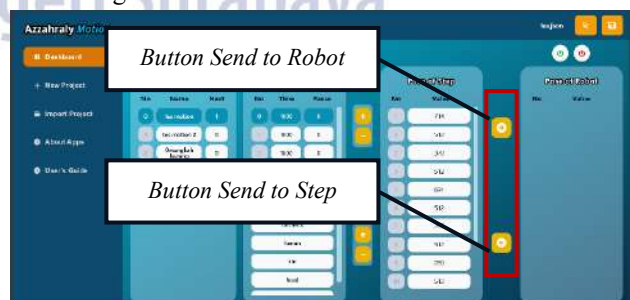
Gambar 16. Tampilan Menu Pose of Step

Gambar 16. menunjukkan tampilan Pose of Step, di mana nilai setiap ID Servo disimpan di setiap step motion dan dapat diubah untuk menyesuaikan gerakan robot yang diinginkan.



Gambar 17. Tampilan Menu Pose of Robot

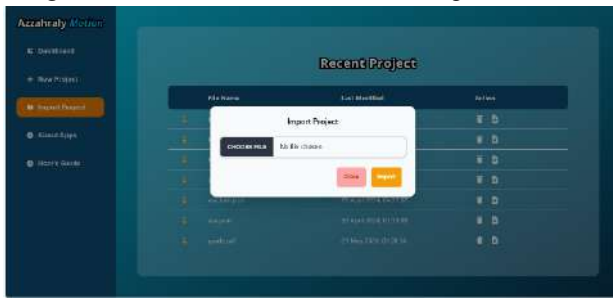
Gambar 17. merupakan tampilan Pose of Robot, yang akan aktif jika aplikasi terhubung dengan robot. Mirip dengan fitur Pose of Step, nilai setiap servo ditampilkan sesuai dengan ID-nya. Pengguna juga dapat menggerakkan servo dengan menonaktifkan ID servo.



Gambar 18. Tampilan Button Send to Robot dan Send to Step

Gambar 18. menunjukkan tombol Send to Robot dan Send to Step. Tombol Send to Robot mengirimkan data nilai

servo ke robot, sedangkan tombol *Send to Step* mengirimkan data nilai *servo* ke *motion step*.



Gambar 19. Tampilan Form Import Project

Pada Gambar 19. merupakan Halaman Import Project menampilkan sebuah *form* yang digunakan untuk mengunggah file JSON dengan ekstensi *.json* dari penyimpanan lokal pada laptop atau PC.



Gambar 20. Tampilan Halaman Menu About Apps

Pada Gambar 20. merupakan tampilan halaman *About Apps* yang menampilkan informasi umum terkait aplikasi meliputi definisi, fitur, versi, dan pembuat aplikasi.



Gambar 21. Tampilan Halaman Menu User's Guide

Pada Gambar 21. merupakan tampilan halaman *User's Guide* yang menampilkan paduan penggunaan aplikasi.

Hasil Pengujian Sistem dengan *Blackbox Testing*

Setelah pengembangan sistem berhasil, langkah berikutnya adalah verifikasi melalui pemeriksaan dan pengujian menyeluruh untuk mengidentifikasi kesalahan atau kegagalan. Peneliti menggunakan metode *blackbox testing* untuk menguji aplikasi.

Kategori	Yang Diharapkan	Hasil Pengamatan	Keterangan
<i>Dashboard</i>	Menampilkan halaman aplikasi dan daftar <i>project</i>	Tampil halaman <i>dashboard</i> dan daftar <i>project</i> yang dibuat	Sukses

Membuat <i>Project</i>	Menampilkan halaman form <i>create new project</i>	Tampil halaman form <i>create new project</i> , input name dan jumlah <i>servo</i> .	Sukses
Mengelola <i>Project</i>	Sistem dapat melakukan CRUD pada <i>project</i>	Terdapat fitur untuk menampilkan, menyimpan, menghapus dan mengunduh <i>project</i> yang dibuat	Sukses
Membuat <i>Motion</i>	Menampilkan halaman form <i>create new motion</i>	Tampil modal form <i>create new motion</i> , input name <i>motion</i> .	Sukses
Mengelola <i>Motion</i>	Sistem dapat melakukan CRUD pada fitur <i>motion</i>	Terdapat fitur untuk menampilkan, mengimpor, mengubah, menghapus dan <i>generate motion</i> yang dibuat ke kode	Sukses
Mengelola <i>Step Motion</i>	Sistem dapat melakukan CRUD pada fitur <i>step motion</i>	Terdapat fitur untuk menampilkan, mengubah, menambah, dan menghapus <i>step motion</i>	Sukses
Mengelola Nilai ID <i>Servo</i>	Sistem dapat melakukan CRUD pada fitur <i>Pose of step</i> dan <i>Pose of robot</i>	Terdapat fitur untuk menampilkan, mengubah, menyimpan, menambah, dan menghapus nilai ID <i>servo</i>	Sukses
Mengelola <i>Servos ID Group</i>	Sistem dapat melakukan CRUD pada fitur <i>Servos ID Group</i>	Terdapat fitur untuk menampilkan, mengubah, menyimpan, menambah, dan menghapus nilai ID <i>servo</i>	Sukses
<i>Play Motion</i>	Sistem dapat menjalankan <i>motion</i>	Robot bergerak sesuai dengan <i>motion</i> yang dijalankan	Sukses
<i>Stop Motion</i>	Sistem dapat menghentikan <i>motion</i> yang sedang	Robot berhenti bergerak.	Sukses

	dijalankan <i>motion</i>		
Koneksi dengan Robot	Sistem menampilkan USB <i>Serial port</i> yang terhubung dengan PC/Laptop	Tampil USB <i>Serial port</i> yang terhubung dengan PC/Laptop	Sukses
<i>Import Project</i>	Menampilkan halaman form <i>import project</i>	Tampil modal form <i>import project</i> .	Sukses
<i>About Apps</i>	Menampilkan halaman menu <i>About Apps</i>	Tampil halaman menu <i>About Apps</i>	Sukses
<i>User's Guide</i>	Menampilkan halaman menu <i>User's Guide</i>	Tampil modal buku penggunaan aplikasi dalam format .pdf	Sukses

Pengujian sistem yang dilakukan pada aplikasi "*Motion Robo Maker Apps*" berbasis *website* dapat disimpulkan bahwa semua pengujian tersebut memperoleh hasil yang memuaskan dengan total keberhasilannya sebesar 100%.

Hasil Pengujian Pembacaan Nilai *Servo* dari Aplikasi

Pengujian ini digunakan untuk mengetahui tingkat keberhasilan aplikasi dalam membaca nilai *servo* pada robot. Setelah beberapa percobaan, ditemukan perbedaan antara nilai *servo* yang dikirimkan ke robot dan hasil implementasi gerakan *servo* pada robot. Oleh karena itu, dilakukan pengujian untuk mengetahui nilai *error* dari proses ini. Berikut hasil pembacaan nilai *servo* oleh aplikasi melalui USB *Serial Port*.

ID	Step 0			Step 1			Step 2		
	Nm	Nk	ϵ	Nm	Nk	ϵ	Nm	Nk	ϵ
1	631	631	0,0%	744	744	0,0%	838	838	0,0%
2	445	445	0,0%	306	306	0,0%	183	183	0,0%
3	435	435	0,0%	343	343	0,0%	270	270	0,0%
4	557	557	0,0%	661	661	0,0%	753	753	0,0%
5	301	301	0,0%	509	509	0,0%	516	516	0,0%
6	632	632	0,0%	454	454	0,0%	473	473	0,0%
7	560	560	0,0%	276	276	0,0%	304	304	0,0%
8	531	531	0,0%	800	800	0,0%	750	750	0,0%
9	601	601	0,0%	688	688	0,0%	713	713	0,0%
10	425	425	0,0%	343	343	0,0%	348	348	0,0%
11	508	508	0,0%	496	496	0,0%	490	490	0,0%
12	538	538	0,0%	605	605	0,0%	648	648	0,0%
13	503	503	0,0%	511	511	0,0%	511	501	0,0%
$\Sigma \epsilon$ Step 0			0,0%	$\Sigma \epsilon$ Step 1			$\Sigma \epsilon$ Step 2		
$\Sigma \epsilon$ Total			0,0%				0,0%		

Gambar 22. Hasil Nilai *Error* Pembacaan Nilai *Servo*

Terlihat nilai eror masing-masing *servo* pada robot hasil pengujian dari 3 variasi gerakan rata-rata *error* sebesar 0% dihasilkan. Dari data tersebut dapat disimpulkan bahwa *Robo Motion Maker Apps* dapat membaca nilai *servo* pada robot secara akurat.

Pengujian Pengiriman Nilai *Servo* dari Aplikasi ke Mikrokontroller

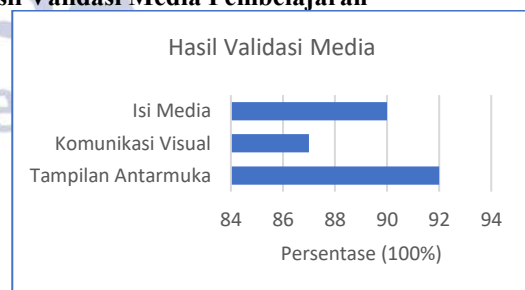
Pengujian ini bertujuan untuk mengetahui tingkat keberhasilan aplikasi dalam mengirimkan nilai *servo* ke robot. Setelah beberapa percobaan, ditemukan perbedaan antara nilai *servo* yang dikirimkan ke robot dan hasil gerakan *servo*. Oleh karena itu, dilakukan pengujian untuk menentukan nilai *error* dari proses ini. Berikut hasil pengiriman nilai *servo* dari aplikasi ke mikrokontroller melalui USB *Serial Port*.

ID	Step 0			Step 1			Step 2		
	Nm	Nk	ϵ	Nm	Nk	ϵ	Nm	Nk	ϵ
1	631	631	0,0%	744	700	5,9%	863	825	4,4%
2	445	442	0,7%	306	362	18,3%	184	206	12,0%
3	435	402	7,6%	343	343	0,0%	239	245	2,5%
4	557	589	5,7%	661	642	2,9%	760	752	1,1%
5	301	292	3,0%	509	492	3,3%	536	520	3,0%
6	632	639	1,1%	454	477	5,1%	475	482	1,5%
7	560	561	0,2%	276	291	5,4%	297	312	5,1%
8	531	531	0,0%	800	765	4,4%	741	729	1,6%
9	601	601	0,0%	688	667	3,1%	753	720	4,4%
10	425	422	0,7%	343	362	5,5%	311	346	11,3%
11	508	515	1,4%	496	503	1,4%	491	498	1,4%
12	538	543	0,9%	605	603	0,3%	648	649	0,2%
13	503	502	0,2%	511	509	0,4%	511	509	0,4%
$\Sigma \epsilon$ Step 0			2%	$\Sigma \epsilon$ Step 1			$\Sigma \epsilon$ Step 2		
$\Sigma \epsilon$ Total			3%						

Gambar 23. Hasil Nilai *Error* Pengiriman Nilai *Servo*

Terlihat nilai *error* masing-masing *servo* pada robot sebagian besar tidak sesuai dengan nilai masukan, dan hasil pengujian dari 3 variasi gerakan rata-rata *error* sebesar 3% dihasilkan. Beberapa faktor penyebab terjadinya ketidaksesuaian nilai keluaran pada robot, salah satunya adalah umur *servo* yang cukup lama. Faktor lainnya bisa dari segi mekanik dan bisa juga dari segi *hardware*, misalnya terlalu kencang atau terlalu longgarnya baut yang terpasang pada robot dapat mempengaruhi pergerakan *servo*.

Hasil Validasi Media Pembelajaran

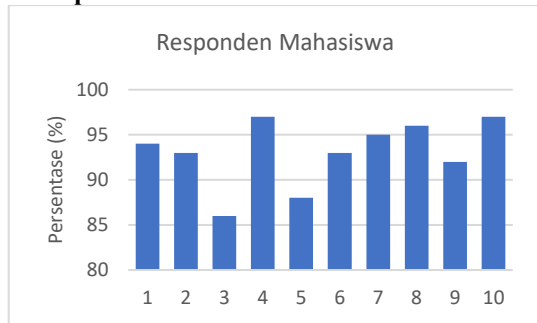


Gambar 24. Grafik Hasil Validasi Media

Berdasarkan Gambar 24. Merupakan grafik hasil validasi ahli media menunjukkan bahwa aspek tampilan antarmuka aplikasi memperoleh rating 92% (Sangat Valid), aspek komunikasi visual 87% (Valid), dan aspek isi media 90% (Sangat Valid). Kesimpulannya, validasi Media *Robo Motion Maker Apps* berbasis *website* sangat valid untuk

digunakan sebagai media pembelajaran robotika bagi mahasiswa peserta kontes.

Hasil Respon Mahasiswa



Gambar 25. Hasil Responden Mahasiswa Terhadap Aplikasi

Berdasarkan Gambar 25. Merupakan grafik hasil respon mahasiswa terhadap aplikasi menunjukkan persentase 93%, menandakan bahwa Media Robo Motion Maker Apps berbasis website sangat valid sebagai media pembelajaran robotika. Responden terdiri dari 20 mahasiswa anggota tim Azzahrly Divisi KRSTI DEWO Robotik Fakultas Teknik Universitas Negeri Surabaya..

Hasil Belajar Mahasiswa

Hasil belajar mahasiswa mempengaruhi efektivitas media pembelajaran: semakin tinggi hasil belajar, semakin efektif media pembelajaran. Untuk mengukur kenaikan hasil belajar, dilakukan penilaian hasil belajar/*Posttest* menggunakan metode *One Shot Case Study* setelah mahasiswa tim Azzahrly Divisi KRSTI DEWO Robotik menggunakan *Media Robo Motion Maker Apps* berbasis website. Keefektifan media pembelajaran dinilai melalui dua teknik pengumpulan data: aspek *critical thinking* dan *collaborative skills*, sebagai berikut :

- 1) *Aspek Chritical Thingking*: Aspek *critical thinking* diukur dengan mengumpulkan data melalui pertanyaan seputar robotika yang diberikan kepada responden untuk dijawab. Teknik pengumpulan data hasil belajar/*posttest* dilakukan menggunakan Google Form.

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Hasil Belajar tentang KRSTI	.208	20	.023	.917	20	.069

a. Lilliefors Significance Correction

Gambar 26. Hasil Uji Normalitas Aspek *Critical Thinking*

Hasil Uji Shapiro-Wilk menunjukkan nilai signifikan 0,088 pada soal *posttest* aspek *critical thinking*. Karena nilai sig. $0,088 \geq 0,05$, H_0 diterima, sehingga data berdistribusi normal.

One-Sample Test						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Hasil Belajar tentang KRSTI	6.532	19	.000	9.000	5.44	10.56

Gambar 27. Hasil Uji Hipotesis Aspek *Critical Thinking*

Hasil *One-Sample T Test* menunjukkan nilai signifikan 0,000 pada soal *posttest*. Karena nilai sig. $0,000 \leq 0,05$, H_0 ditolak dan H_1 diterima, sehingga data menunjukkan bahwa media sangat efektif.

- 2) *Aspek Collaborative Skills*: Aspek *collaborative skill* diukur dengan mengumpulkan data berupa nilai/skor hasil penampilan pada Kontes Robot Seni Tari Indonesia (KRSTI) tingkat Wilayah dan Nasional.

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Hasil wilayah dan nasional	.171	8	.200 [*]	.904	8	.847

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Gambar 28. Hasil Uji Normalitas Aspek *Collaborative Skills*

Hasil Uji *Shapiro-Wilk* menunjukkan nilai signifikan 0,847 pada soal *posttest* aspek *collaborative skills*. Karena nilai sig. $0,847 \geq 0,05$, H_0 diterima, sehingga data berdistribusi normal.

Test of Homogeneity of Variances						
	Levene Statistic		df1	df2	Sig.	
	Based on Mean	.605	1	4	.480	
	Based on Median	.155	1	4	.714	
	Based on Median and with adjusted df	.155	1	3.295	.718	
	Based on trimmed mean	.562	1	4	.495	

Gambar 29. Hasil Uji Homogenitas Aspek *Collaborative Skills*

Hasil perhitungan menunjukkan bahwa semua data memiliki nilai signifikansi (sig.) $> 0,05$, sehingga dapat disimpulkan bahwa kelompok data yang diteliti memiliki kesamaan varians atau homogen.

One-Sample Test						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Hasil wilayah dan nasional	58.545	5	.000	29.11167	27.7892	30.4351

Gambar 30. Hasil Uji Hipotesis Aspek *Collaborative Skills*

Hasil *One-Sample T Test* menunjukkan nilai signifikan 0,000 pada soal *posttest* aspek *collaborative skills*. Karena nilai sig. $0,000 \leq 0,05$, H_0 ditolak dan H_1 diterima, sehingga data menunjukkan bahwa media sangat efektif.

PENUTUP

Simpulan

1. Aplikasi *Motion Robo Maker Apps* dikembangkan dengan model *eXtreme Programming* (XP) menggunakan bahasa pemrograman JavaScript, library React Js untuk *back-end*, dan *Framework*

Tailwind CSS serta *library* Daisy UI untuk *front-end*. Setelah melalui berbagai pengujian, termasuk *blackbox testing* dengan tingkat keberhasilan 100%, aplikasi ini mendapat validasi tinggi dari para ahli sebesar 90% dan respon positif dari mahasiswa sebesar 93%. Hasil tersebut menunjukkan bahwa aplikasi ini dapat digunakan sebagai media pembelajaran robotika yang valid dan efektif.

2. Penggunaan aplikasi *Motion Robo Maker Apps* terbukti efektif dalam meningkatkan kemampuan *critical thinking* dan *collaborative skills* mahasiswa, dengan nilai rata-rata tes hasil belajar sebesar 88 dan skor rata-rata penampilan pada Kontes Robot Seni Tari Indonesia sebesar 79,11. Uji normalitas menunjukkan distribusi normal dengan nilai signifikan untuk aspek *critical thinking* sebesar 0,089 dan aspek *collaborative skills* sebesar 0,847, yang keduanya lebih besar dari 0,05. Dari data hasil belajar mahasiswa, memperkuat kesimpulan bahwa aplikasi ini sukses dalam mendukung kemampuan mahasiswa dalam kompetisi robotika.

Saran

1. Peneliti dapat memperbaiki dan mengoptimalkan algoritma aplikasi untuk mengurangi error pengiriman nilai servo, menambah fitur inovatif seperti tutorial dan simulasi robot.
2. Pendidik dapat mengintegrasikan aplikasi ini dalam pembelajaran robotika, memberikan pelatihan tentang penggunaannya, serta melakukan monitoring dan evaluasi pembelajaran.
3. Mahasiswa dapat memanfaatkan semua fitur aplikasi untuk belajar robotika, menggunakan aplikasi ini secara mandiri atau kolaboratif untuk meningkatkan keterampilan berpikir kritis dan kerja sama.
4. Perguruan tinggi dapat memberikan arahan dan bimbingan dalam pengembangan ide penelitian yang berkelanjutan terkait aplikasi ini.

Ucapan Terima Kasih

Terimakasih yang tulus diucapkan kepada:

1. Tuhan Yang Maha Kuasa serta dukungan kedua orang tua saya.
2. Dosen pembimbing skripsi Bapak Aditya Prapanca, S.T., M.Kom. yang telah memberi tuntunan hingga penelitian selesai serta seluruh dosen dan guru yang telah membimbing saya.
3. Teman – teman yang telah memberi arahan kepada peneliti dalam proses penyusunan artikel ilmiah ini.

DAFTAR PUSTAKA

Alasiry, A. H., Satria, N. F., & Sugiarto, A. (2018). Balance Control of Humanoid Dancing Robot

ERISA while Walking on Sloped Surface using PID. 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 577–581.

<https://doi.org/10.1109/ISRITI.2018.8864447>

Blut, M., Wang, C., Wunderlich, N. V., Brock, C., & Wang, C. (2023). *Understanding anthropomorphism in service provision : a meta-analysis of physical robots , chatbots , and other AI*. 2021, 632–658.

Fahd, M. A., Purwanto, D., & Fatoni, M. H. (2018). *Rancang Bangun Robot Penari Humanoid dengan*. 7(2).

Kusumoputro, B., Purnomo, M. H., Rochardjo, H. S. B., Prabowo, G., Purwanto, D., Pitowarno, E., Indrawanto, Mutijasari, K., & Muis, A. (2023). *Buku Pedoman Kontes Robot Indonesia Tahun 2023* (S. Witaradya, I. Ibrahim, K. Mumpuni, D. Fitriani, A. D. Indrianto, T. Yunita, & G. Sinthya (ed.)). Balai Pengembangan Talenta Indonesia Pusat Prestasi Nasional Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi.

Pratama, A. G., Nurussa'adah, & Rif'an, M. (2016). *Penentuan Sudut Lengan Robot Humanoid Berdasarkan Koordinat Yang Dikirim Dari Pc Menggunakan User Interface Yang Dibuat Dari Qt*. 1–6.

Robotis. (1999a). *Information of Company*. <https://en.robotis.com/>

Robotis. (1999b). *RoboPlus 1.0*. <https://robotis.co.uk/software/roboplus-1-0.html>

Satria, N. F., Binugroho, E. H., Chairussy, R. H., Basuki, D. K., & Nobelia, B. S. (2023). *The Development of Dance Movement in Humanoid Robot Dancing*. 626–632.

<https://doi.org/10.5220/0010950100003260>

Somiseti, K., Tripathi, K., & Verma, J. K. (2020). *Design , Implementation , and Controlling of a Humanoid Robot*. 831–836.

<https://doi.org/10.1109/ComPE49325.2020.9200020>

Sugiyono. (2011). *Metode Penelitian Kuantitatif, Kualitatif, dan R & D* (Alfabeta (ed.); 13 ed.). Alfabeta.

Suryantara, I. G. N. (2017). *Merancang Apikasi dengan Metodologi eXtreme Programmings* (PT Elex Media Komputindo (ed.); Pertama). PT Elex Media Komputindo.

Widoyoko, S. E. P. (2017). Analisis Kualitas Butir Soal Ujian Akhir. *Jurnal Pendidikan Surya Edukasi (JPSE)*, 3(November), 67–82.

Xu, S., & Zhou, L. (2021). *Design and implementation of the imitation human shape dance robot*. 2nd International Conference on Intelligent Design (ICID).

<https://doi.org/10.1109/ICID54526.2021.00082>