

Implementasi Running Text LED P10 Berbasis Arduino–ESP32

Nur Shita Harata¹, Jajat Sudrajat²
Teknik Komputer, Politeknik Piksi Ganesha
piksi.nurshitaharata.23304014@gmail.com

Abstract. The development of Internet of Things (IoT) technology enables more flexible and efficient remote control of electronic devices. The aim of this research project is the development and implementation of an ESP32-based system for displaying scrolling text, which can be controlled via the Telegram Bot application. Rapid Application Development (RAD) was used as the development methodology, as this approach accelerates the system development process in the phases of requirements planning, design, development, implementation, and testing. The developed system allows users to send text messages via the Telegram application, which are then processed by the ESP32 and displayed in real time on the scrolling text display. To ensure that all functions meet user requirements, system tests were conducted using black-box testing. The test results showed that the system could receive commands, process data, and display text correctly without any significant problems. Thus, the ESP32-based scrolling text system, controlled via the Telegram Bot, was successfully implemented and represents a practical, user-friendly, and efficient alternative for information transmission.

Keywords - information system, goods delivery, web-based system, SDLC, waterfall model.

Abstrak. Perkembangan teknologi Internet of Things (IoT) memungkinkan pengendalian perangkat elektronik secara jarak jauh dengan tingkat fleksibilitas dan efisiensi yang lebih tinggi. Penelitian ini bertujuan untuk merancang dan merealisasikan sistem Running Text berbasis ESP32 yang dapat dikendalikan melalui aplikasi Telegram Bot. Metode pengembangan yang diterapkan adalah Rapid Application Development (RAD), karena mampu mempercepat proses pembangunan sistem melalui tahapan perencanaan kebutuhan, perancangan, konstruksi, serta implementasi dan pengujian. Sistem yang dikembangkan memungkinkan pengguna mengirimkan pesan teks melalui aplikasi Telegram, yang kemudian diproses oleh ESP32 dan ditampilkan pada media running text secara real-time. Pengujian sistem dilakukan menggunakan metode black box testing untuk memastikan setiap fungsi berjalan sesuai dengan kebutuhan pengguna. Hasil pengujian menunjukkan bahwa sistem mampu menerima perintah, memproses data, dan menampilkan teks dengan baik tanpa kendala yang signifikan. Dengan demikian, sistem Running Text berbasis ESP32 yang dikendalikan melalui Telegram Bot berhasil diimplementasikan dan dapat menjadi solusi alternatif dalam penyampaian informasi yang praktis, mudah dioperasikan, serta efisien.

Kata Kunci - Sistem Informasi, SPBU, UML, Waterfall, Karawang.

Diterima Redaksi : 25-06-2024 | **Selesai Revisi :** 30-06-2024 | **Terbit :** 31-06-2025

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang berlangsung pesat telah membawa perubahan signifikan dalam pola penyampaian informasi [1]. Media informasi konvensional yang bersifat statis dinilai kurang efektif karena keterbatasannya dalam menyampaikan informasi secara cepat dan fleksibel. Kondisi tersebut mendorong perlunya penerapan media digital yang mampu menampilkan informasi secara dinamis serta mudah diperbarui sesuai dengan kebutuhan [2].

Running text merupakan salah satu bentuk media informasi digital yang banyak dimanfaatkan di berbagai lingkungan, seperti instansi pendidikan, perkantoran, tempat ibadah, dan fasilitas umum [10]. Media ini mampu menampilkan pesan secara berulang dan menarik perhatian audiens. Namun demikian, sebagian besar sistem running text masih dikelola secara manual atau menggunakan perangkat

khusus yang kurang praktis, sehingga menyulitkan pengguna dalam melakukan pembaruan informasi secara cepat.

Penerapan konsep Internet of Things (IoT) menjadi salah satu solusi dalam meningkatkan efektivitas pengelolaan perangkat elektronik [1]. Teknologi IoT memungkinkan perangkat saling terhubung melalui jaringan internet dan dapat dikendalikan dari jarak jauh. Dengan memanfaatkan konsep ini, sistem running text dapat dikelola secara real-time tanpa dibatasi oleh ruang dan waktu.

ESP32 merupakan salah satu mikrokontroler yang banyak digunakan dalam pengembangan sistem berbasis IoT karena telah dilengkapi dengan modul Wifi dan Bluetooth terintegrasi [7]. Selain itu, ESP32 memiliki kemampuan pemrosesan yang baik, konsumsi daya yang relatif rendah, serta biaya yang terjangkau. Karakteristik tersebut menjadikan ESP32 sebagai alternatif yang sesuai untuk digunakan sebagai pengendali utama pada sistem running text berbasis jaringan.

Pada sistem yang dikembangkan, Telegram Bot digunakan sebagai media antarmuka antara pengguna dan perangkat. Telegram dipilih karena merupakan aplikasi pesan instan yang populer serta menyediakan Application Programming Interface (API) yang mendukung pengembangan bot [4]. Melalui Telegram Bot, pengguna dapat mengirimkan perintah atau pesan teks dengan mudah tanpa memerlukan aplikasi tambahan, sehingga meningkatkan kemudahan dan kenyamanan dalam pengoperasian sistem.

Pengembangan sistem pada penelitian ini menerapkan metode Rapid Application Development (RAD). Metode RAD menekankan pada proses pengembangan yang cepat melalui tahapan perencanaan kebutuhan, perancangan, konstruksi, serta implementasi dan pengujian [9]. Pendekatan ini memungkinkan adanya umpan balik dari pengguna secara berkelanjutan, sehingga sistem yang dikembangkan dapat disesuaikan secara optimal dengan kebutuhan pengguna.

Beberapa penelitian terdahulu telah membahas pemanfaatan mikrokontroler dan media komunikasi berbasis internet untuk pengendalian perangkat elektronik. Namun, integrasi antara ESP32, Telegram Bot, serta penerapan metode RAD pada sistem running text masih perlu dikaji lebih lanjut untuk mengetahui efektivitas dan kinerja sistem secara menyeluruh. Oleh karena itu, penelitian ini dilakukan sebagai upaya pengembangan dan penyempurnaan terhadap sistem yang telah ada.

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk merancang dan membangun sistem running text berbasis ESP32 yang dikendalikan melalui Telegram Bot dengan menerapkan metode RAD. Diharapkan sistem yang dihasilkan mampu memberikan solusi penyampaian informasi yang lebih praktis, fleksibel, dan efisien, serta dapat diimplementasikan pada berbagai kebutuhan media informasi berbasis IoT.

TINJAUAN PUSTAKA

Perkembangan teknologi Internet of Things (IoT) dalam beberapa tahun terakhir menjadi fokus utama dalam berbagai penelitian, khususnya pada pengembangan sistem kendali dan pemantauan jarak jauh [1]. Berbagai studi menunjukkan bahwa penerapan IoT mampu meningkatkan efisiensi, fleksibilitas, serta kemudahan dalam pengelolaan perangkat elektronik dibandingkan dengan sistem konvensional. Penelitian-penelitian tersebut menegaskan bahwa keberhasilan implementasi sistem berbasis IoT sangat dipengaruhi oleh integrasi yang baik antara perangkat

keras, jaringan internet, serta antarmuka pengguna yang sederhana dan mudah dioperasikan.

Dalam bidang media informasi digital, running text telah banyak dikembangkan sebagai sarana penyampaian informasi yang bersifat dinamis. Sejumlah penelitian terdahulu menunjukkan bahwa penggunaan running text berbasis mikrokontroler memberikan keunggulan dalam kecepatan pembaruan informasi serta meningkatkan daya tarik visual [10]. Namun demikian, sebagian penelitian masih menerapkan pengelolaan secara lokal, seperti melalui pengaturan langsung menggunakan komputer atau media penyimpanan eksternal, yang dinilai kurang efisien ketika sistem dioperasikan pada lokasi yang berbeda dengan penggunanya. Seiring dengan meningkatnya kebutuhan akan pengendalian jarak jauh, beberapa penelitian mulai mengintegrasikan sistem running text dengan teknologi jaringan internet. Pemanfaatan mikrokontroler yang telah mendukung konektivitas internet memungkinkan sistem diakses dan dikendalikan secara real-time dari berbagai lokasi [7]. Hasil penelitian tersebut menyimpulkan bahwa sistem berbasis jaringan memiliki tingkat efisiensi dan kemudahan operasional yang lebih baik, terutama pada lingkungan publik yang memerlukan pembaruan informasi secara cepat dan berkelanjutan.

ESP32 menjadi salah satu mikrokontroler yang banyak digunakan dalam pengembangan dan penelitian IoT karena kemampuannya dalam menyediakan konektivitas Wi-Fi secara terintegrasi. Berbagai studi membuktikan bahwa ESP32 memiliki kinerja yang stabil dalam pengolahan data serta komunikasi jaringan. Selain itu, penggunaan ESP32 dinilai mampu menekan biaya pengembangan tanpa mengurangi kualitas kinerja sistem, sehingga sesuai untuk diterapkan pada sistem informasi digital seperti running text [7].

Dari sisi antarmuka pengguna, beberapa penelitian sebelumnya memanfaatkan aplikasi berbasis web atau aplikasi khusus sebagai media pengendali sistem. Namun, pendekatan tersebut seringkali memerlukan proses pengembangan yang kompleks serta mengharuskan pengguna melakukan instalasi aplikasi tambahan. Penelitian-penelitian terbaru cenderung mengarah pada pemanfaatan aplikasi pesan instan sebagai media kendali karena lebih ringan, mudah diakses, dan telah familiar bagi pengguna. Telegram Bot, khususnya, banyak digunakan karena menyediakan Application Programming Interface (API) yang stabil serta mendukung komunikasi dua arah secara real-time [3].

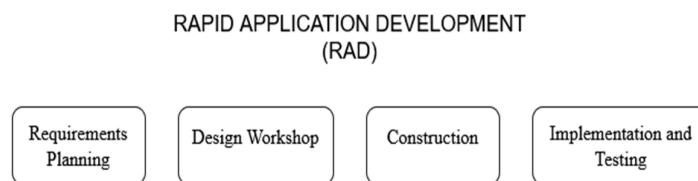
Dari aspek metodologi pengembangan, metode Rapid Application Development (RAD) telah diterapkan dalam berbagai penelitian sistem informasi maupun sistem berbasis IoT. Hasil penelitian menunjukkan bahwa metode RAD mampu mempercepat proses pengembangan sistem melalui tahapan yang bersifat iteratif serta melibatkan pengguna secara aktif [9]. Pendekatan ini dinilai efektif untuk penelitian yang menuntut implementasi sistem dalam waktu relatif singkat, tanpa mengesampingkan kesesuaian sistem dengan kebutuhan pengguna.

Berdasarkan kajian terhadap penelitian-penelitian terdahulu, dapat disimpulkan bahwa integrasi teknologi IoT, mikrokontroler ESP32, serta media kendali berbasis aplikasi pesan instan memiliki potensi besar dalam pengembangan sistem running text yang lebih fleksibel dan efisien. Namun demikian, masih diperlukan penelitian yang mengkaji secara terintegrasi pemanfaatan ESP32 dan Telegram Bot dengan pendekatan pengembangan sistem yang cepat dan adaptif, seperti metode RAD.

Oleh karena itu, penelitian ini diarahkan pada pengembangan sistem running text berbasis ESP32 yang dikendalikan melalui Telegram Bot dengan menerapkan metode Rapid Application Development. Secara implisit, penelitian ini didasarkan pada asumsi bahwa penerapan ESP32 dan Telegram Bot dengan metode RAD mampu menghasilkan sistem running text yang fungsional, mudah digunakan, serta efektif sebagai media penyampaian informasi berbasis IoT.

METODE

Penelitian ini menerapkan metode Rapid Application Development (RAD), yaitu pendekatan pengembangan sistem yang berfokus pada kecepatan proses, pengembangan secara berulang (iteratif), serta keterlibatan pengguna secara langsung dalam setiap tahapannya [9]. Metode RAD dipilih karena dinilai efektif dalam mempercepat proses perancangan dan penerapan sistem running text dua lantai berbasis Arduino yang dapat dikendalikan melalui aplikasi Telegram [9]. Penerapan metode RAD dalam penelitian ini terdiri atas beberapa tahapan utama, yaitu perencanaan kebutuhan, perancangan sistem, konstruksi sistem, serta tahap implementasi dan pengujian [9].



Gambar 1. Metode RAD

Penggunaan metode Rapid Application Development (RAD) ditujukan untuk mempercepat proses pengembangan aplikasi melalui pendekatan iteratif serta keterlibatan pengguna secara aktif. Metode ini memberikan fleksibilitas bagi pengembang dalam menyesuaikan sistem dengan kebutuhan pengguna, sehingga setiap perubahan dapat

segera direspons dan diimplementasikan. Dengan penerapan RAD, sistem yang dihasilkan menjadi lebih selaras dengan kebutuhan nyata pengguna serta dapat diselesaikan dalam waktu yang relatif singkat tanpa mengurangi kualitas fungsionalnya.

Tahapan RAD (Rapid Application Development)

1. Perencanaan Kebutuhan (Requirements Planning)
Tahap awal untuk mengidentifikasi tujuan sistem, kebutuhan pengguna, dan batasan sistem melalui diskusi antara pengguna dan pengembang.
2. Perancangan Sistem (Design Workshop)
Tahap perancangan dilakukan secara kolaboratif dengan pengguna melalui pembuatan prototipe, desain antarmuka, dan alur sistem secara iteratif.
3. Konstruksi Sistem (Construction)
Tahap pengembangan sistem berdasarkan desain yang telah disepakati, meliputi proses pengkodean dan pengujian awal.
4. Implementasi dan Pengujian (Implementation and Testing)

Tahap penerapan sistem ke lingkungan pengguna, disertai pengujian akhir untuk memastikan sistem berjalan sesuai kebutuhan.

HASIL DAN PEMBAHASAN

Hasil Perencanaan Kebutuhan (Requirements Planning)

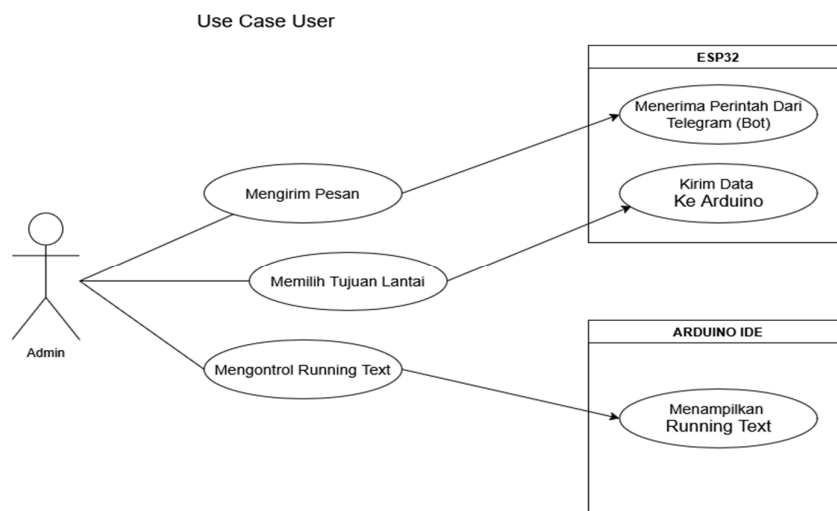
Tahap perencanaan kebutuhan bertujuan untuk mengidentifikasi kebutuhan sistem Running Text berbasis ESP32 yang dikendalikan melalui Telegram Bot. Pada tahap ini, analisis difokuskan pada permasalahan utama, yaitu perlunya sistem penampil teks yang dapat dioperasikan secara jarak jauh dan real-time tanpa memerlukan pengaturan langsung pada perangkat [1].

Hasil analisis kebutuhan menunjukkan bahwa sistem harus mampu menerima perintah dari pengguna melalui aplikasi Telegram, memproses pesan teks yang dikirimkan, serta menampilkannya pada modul running text. Selain itu, sistem diharapkan memiliki tingkat kemudahan penggunaan yang baik, waktu respons yang cepat, serta koneksi yang stabil antara ESP32 dan Telegram Bot [7].

Kebutuhan fungsional dan nonfungsional yang diperoleh pada tahap ini selanjutnya direpresentasikan dalam bentuk use case diagram guna menggambarkan interaksi umum antara pengguna dan sistem.

a) Use Case Diagram

Use case diagram pada sistem running text dua lantai ini digunakan untuk menggambarkan fungsi-fungsi utama yang dapat dilakukan oleh pengguna (admin) dalam mengoperasikan dan mengendalikan sistem melalui aplikasi Telegram. Fungsi-fungsi tersebut mencakup pengiriman pesan teks, penentuan tujuan tampilan pada lantai tertentu, serta pengendalian tampilan running text pada modul LED P10. Diagram ini dirancang untuk memperlihatkan bentuk interaksi antara pengguna dengan sistem yang dikendalikan oleh ESP32 dan Arduino sebagai unit pengendali utama.



Gambar 2. Use Case Diagram

Diagram use case menampilkan satu aktor utama, yaitu Admin, yang berperan sebagai pengguna sistem. Admin berinteraksi dengan sistem melalui Telegram Bot untuk mengirim pesan, memilih tujuan lantai, dan mengontrol running text.

Use case Mengirim Pesan menggambarkan proses admin mengirimkan teks informasi yang diterima sistem sebagai input untuk ditampilkan pada modul LED P10. Pada use case Memilih Tujuan Lantai, admin menentukan lokasi tampilan pesan, yaitu lantai 1, lantai 2, atau keduanya, sehingga pengelolaan informasi menjadi lebih fleksibel. Use case Mengontrol Running Text menunjukkan proses pemrosesan perintah oleh sistem, di mana ESP32 menerima perintah dari Telegram dan meneruskannya ke Arduino melalui komunikasi serial. Arduino kemudian mengendalikan modul LED P10 untuk menampilkan pesan dalam bentuk teks berjalan.

Secara keseluruhan, diagram ini menggambarkan alur sistem yang sederhana dan efektif, memungkinkan admin mengelola informasi pada dua lantai secara terpusat dan real-time, dengan ESP32 sebagai pengelola komunikasi dan Arduino sebagai pengendali tampilan LED P10.

B. Hasil Perancangan Sistem (Design Workshop)

Tahap perancangan sistem dilaksanakan berdasarkan hasil analisis kebutuhan

yang telah ditetapkan pada tahap sebelumnya. Pada fase ini, proses perancangan dilakukan secara kolaboratif dengan memperhatikan alur komunikasi antara pengguna, Telegram Bot, ESP32, dan modul running text [7].

Hasil perancangan direpresentasikan dalam bentuk activity diagram yang menggambarkan urutan aktivitas sistem, mulai dari pengguna mengirimkan perintah melalui Telegram hingga teks ditampilkan pada modul running text. Selain itu, dilakukan perancangan struktur sistem serta alur pemrosesan data untuk memastikan pesan yang diterima dapat diproses secara tepat oleh mikrokontroler [10].

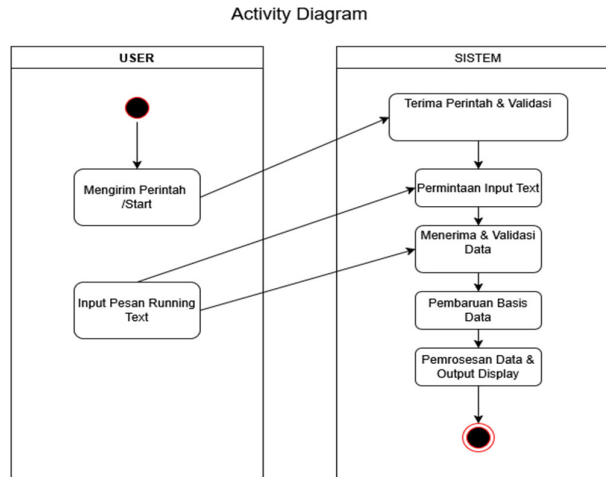
Pada tahap ini juga dirancang tampilan antarmuka Telegram Bot sebagai sarana interaksi antara pengguna dan sistem. Prototipe yang dihasilkan berfungsi sebagai acuan pada tahap pengembangan serta memungkinkan dilakukannya evaluasi desain secara lebih awal sebelum sistem diimplementasikan.

b) Activity Diagram

Activity diagram pada sistem running text berbasis ESP32 yang dikendalikan melalui Telegram Bot menggambarkan alur interaksi antara user dan sistem dalam proses pengiriman serta penampilan teks berjalan. Diagram ini dibagi menjadi dua swimlane, yaitu User dan Sistem, untuk memperjelas peran dan aktivitas masing-masing. Proses diawali saat user mengirimkan perintah /start melalui Telegram sebagai tahap inisialisasi. Sistem kemudian menerima dan memvalidasi perintah tersebut, lalu meminta input teks sebagai tanda kesiapan sistem. Selanjutnya, user mengirimkan pesan running text yang akan ditampilkan.

Pesan yang diterima sistem divalidasi kembali, kemudian disimpan atau diperbarui pada media penyimpanan. Pada tahap akhir, sistem mengolah data dan

menampilkan teks berjalan pada modul running text. Proses berakhir setelah pesan berhasil ditampilkan, yang ditunjukkan dengan final node pada activity diagram. Secara keseluruhan, activity diagram ini menunjukkan alur kerja sistem yang terstruktur dari awal hingga akhir serta kesesuaian antara interaksi pengguna dan proses internal sistem dalam pengendalian running text.



Gambar 2. Activity Diagram

C. Hasil Konstruksi Sistem (Construction)

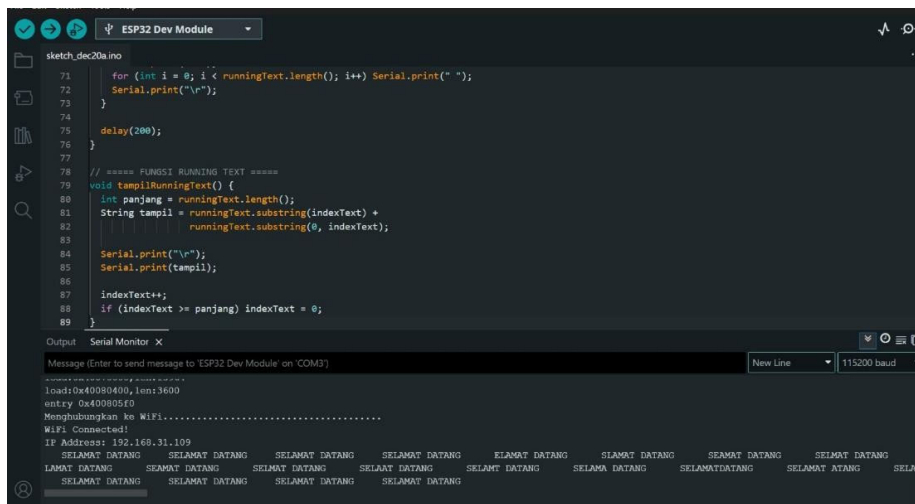
Tahap konstruksi merupakan fase pengembangan sistem yang dilaksanakan berdasarkan desain yang telah dirancang sebelumnya. Pada tahap ini dilakukan pemrograman pada ESP32 untuk mengintegrasikan sistem dengan Telegram Bot serta mengendalikan tampilan running text [7].

Proses konstruksi meliputi penerapan logika pemrograman untuk menerima pesan, memproses perintah, dan menampilkan teks sesuai dengan masukan pengguna [10]. Selain itu, pengujian awal dilakukan untuk memastikan setiap fungsi sistem berjalan sesuai dengan rancangan yang telah ditetapkan.

Hasil pada tahap ini menunjukkan bahwa sistem telah berhasil dikembangkan dan mampu menampilkan teks berjalan berdasarkan perintah yang dikirimkan melalui Telegram Bot [3]. Tampilan sistem pada fase ini telah berfungsi dengan baik dan siap untuk dilakukan pengujian pada tahap selanjutnya.



Gambar 3. Tampilan Bot Telegram



```

71 for (int i = 0; i < runningText.length(); i++) Serial.print(" ");
72 Serial.print("\n");
73 }
74
75 delay(200);
76 }
77
78 // ===== RUNNING TEXT =====
79 void tampilRunningText() {
80   int panjang = runningText.length();
81   String tampil = runningText.substring(indexText) +
82     runningText.substring(0, indexText);
83   Serial.print("\n");
84   Serial.print(tampil);
85   indexText++;
86   if (indexText >= panjang) indexText = 0;
87 }
88
89

```

Output Serial Monitor X

Message (Enter to send message to ESP32 Dev Module on COM3)

load:0x40000000, len:3600
entry 0x40000000

Menghubungkan ke Wifi.....

Wifi Connected!

IP Address: 192.168.31.109

SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG
SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG
SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG SELAMAT DATANG

Gambar 5. Hasil Pengujian Sistem Melalui Serial Monitor Arduino IDE

D. Hasil Implementasi dan Pengujian (Implementation and Testing)

Tahapan implementasi dilaksanakan dengan menerapkan sistem Running Text berbasis ESP32. Sistem dijalankan dengan memanfaatkan koneksi internet guna

memastikan proses komunikasi antara perangkat ESP32 dan Telegram Bot dapat berlangsung secara stabil dan berkelanjutan [7].

Pengujian sistem dilakukan menggunakan metode black box testing yang berfokus pada evaluasi fungsi-fungsi utama sistem, meliputi pengiriman perintah, penerimaan pesan, serta proses penampilan teks pada media running text [5]. Setiap skenario pengujian dirancang untuk memverifikasi kesesuaian antara masukan yang diberikan dengan keluaran yang dihasilkan oleh sistem.

Berdasarkan hasil pengujian yang telah dilakukan, seluruh fungsi sistem dapat beroperasi dengan baik dan memenuhi kebutuhan yang telah ditentukan. Sistem mampu menampilkan teks secara real-time serta memberikan respons yang tepat terhadap perintah pengguna, sehingga dapat disimpulkan bahwa sistem layak dan siap untuk digunakan [10].

Black Box

Pengujian black box dilaksanakan untuk mengevaluasi fungsionalitas sistem berdasarkan kesesuaian antara masukan yang diberikan oleh pengguna melalui Telegram Bot dan keluaran yang dihasilkan berupa tampilan teks pada media running text, tanpa mempertimbangkan struktur internal perangkat lunak.

Pada tahap pengujian ini, pengguna mengirimkan berbagai perintah serta pesan teks melalui aplikasi Telegram, yang selanjutnya diproses oleh modul ESP32 dan ditampilkan pada perangkat running text. Penyusunan skenario pengujian dilakukan dengan mengacu pada kebutuhan fungsional yang telah ditetapkan pada tahap perencanaan sistem.

Hasil pengujian menunjukkan bahwa sistem mampu menerima, memproses, dan menampilkan pesan dengan baik, mulai dari proses inisialisasi hingga pembaruan

teks secara real-time. Selain itu, sistem tetap beroperasi secara stabil meskipun menerima pesan dengan variasi panjang karakter yang berbeda.

Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa seluruh fitur utama sistem telah berfungsi sesuai dengan yang diharapkan. Dengan demikian, sistem Running Text berbasis ESP32 yang dikendalikan melalui Telegram Bot dinyatakan memenuhi kebutuhan fungsional dan layak untuk digunakan.

Tabel 1. Pengujian Blackbox

Fitur yang Diuji	Skenario Pengujian	Input	Output yang Diharapkan	Hasil
Koneksi Sistem	Menyalakan ESP32 dan menghubungkannya ke Wi-Fi	ESP32 aktif	ESP32 terhubung ke jaringan dan Telegram Bot aktif	Berhasil
Perintah Start	Mengirim perintah awal ke sistem	/start	Sistem merespons bahwa bot siap digunakan	Berhasil
Pengiriman Teks	Mengirim teks melalui Telegram Bot	“Selamat Datang”	Teks ditampilkan pada running text	Berhasil
Teks Panjang	Mengirim teks dengan jumlah karakter lebih banyak	“Sistem Running Text ESP32”	Teks berjalan dengan lancar tanpa terputus, dan akan terputus jika ada noise	Berhasil
Perubahan Teks	Mengirim teks baru setelah teks sebelumnya	“Uji Sistem”	Teks lama digantikan dengan teks baru	Berhasil
Respon Sistem	Respon Sistem	Teks Telegram	Sistem merespons dengan cepat	Berhasil

E. Pembahasan

Berdasarkan hasil pengembangan dan pengujian yang telah dilakukan, sistem Running Text berbasis ESP32 yang dikendalikan melalui Telegram Bot berhasil direalisasikan dan dapat beroperasi sesuai dengan tujuan penelitian [7]. Penerapan metode Rapid Application Development (RAD) memungkinkan proses pengembangan sistem berlangsung secara lebih cepat dan sistematis melalui tahapan perencanaan kebutuhan, perancangan, konstruksi, serta implementasi dan pengujian [12].

Hasil pada tahap perencanaan kebutuhan menunjukkan bahwa sistem yang dikembangkan mampu menjawab permasalahan utama, yaitu kebutuhan akan

media penampil informasi yang dapat diakses dan dikendalikan secara jarak jauh. Seluruh kebutuhan fungsional yang telah ditetapkan, seperti pengiriman pesan teks melalui Telegram dan penampilan teks pada media running text, berhasil diimplementasikan dengan baik.

Pada tahap perancangan sistem, pemanfaatan activity diagram serta perancangan antarmuka Telegram Bot berperan dalam memvisualisasikan alur kerja sistem secara jelas dan terstruktur. Kondisi ini memberikan kontribusi positif pada tahap konstruksi, karena proses pengembangan dapat dilaksanakan sesuai dengan rancangan yang telah ditetapkan, sehingga dapat meminimalkan terjadinya kesalahan fungsi maupun perubahan desain yang signifikan.

Hasil konstruksi sistem menunjukkan bahwa integrasi antara perangkat ESP32, Telegram Bot, dan modul running text dapat berjalan secara optimal [15]. Sistem mampu menerima perintah dari pengguna, memproses pesan yang diterima, serta menampilkan teks secara real-time. Pengujian black box yang dilakukan pada tahap implementasi menunjukkan bahwa seluruh fungsi utama sistem beroperasi sesuai dengan yang diharapkan, baik pada pengujian teks dengan jumlah karakter yang relatif pendek maupun panjang.

Secara keseluruhan, hasil penelitian menunjukkan bahwa penerapan metode RAD efektif dalam menghasilkan sistem yang fungsional dan sesuai dengan kebutuhan pengguna. Keterlibatan pengguna pada tahap perancangan mendukung fleksibilitas sistem serta memudahkan proses penyesuaian. Dengan demikian, sistem Running Text berbasis ESP32 yang dikendalikan melalui Telegram Bot dinyatakan layak digunakan sebagai media penyampaian informasi yang efisien dan praktis.

KESIMPULAN

Berdasarkan hasil perancangan, pengembangan, serta pengujian yang telah dilaksanakan, dapat disimpulkan bahwa sistem Running Text berbasis ESP32 yang dikendalikan melalui Telegram Bot berhasil direalisasikan dan beroperasi sesuai dengan tujuan penelitian. Sistem ini mampu menerima perintah dan pesan teks dari pengguna melalui aplikasi Telegram, memproses data menggunakan ESP32, serta menampilkan teks berjalan secara real-time pada modul running text.

Penerapan metode Rapid Application Development (RAD) terbukti efektif dalam mempercepat proses pengembangan sistem. Pendekatan iteratif yang disertai dengan keterlibatan pengguna pada setiap tahapan pengembangan memungkinkan sistem yang dihasilkan lebih selaras dengan kebutuhan pengguna serta memudahkan penyesuaian terhadap perubahan selama proses pengembangan berlangsung.

Hasil pengujian menggunakan metode black box testing menunjukkan bahwa seluruh fungsi utama sistem berjalan dengan baik dan memenuhi kebutuhan fungsional yang telah ditetapkan. Oleh karena itu, sistem yang dikembangkan dinyatakan layak digunakan sebagai media penyampaian informasi yang praktis, fleksibel, serta dapat dikendalikan secara jarak jauh.

DAFTAR PUSTAKA

- [1] [1] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2021). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 23(1).

- [2] Buyya, R., Dastjerdi, A. V., & Gill, S. S. (2022). *Internet of Things: Principles and paradigms* (2nd ed.). Morgan Kaufmann.
- [3] Fauzi, A., & Nugroho, S. (2021). Sistem kendali jarak jauh berbasis Telegram Bot dan mikrokontroler ESP32. *Jurnal Teknologi dan Sistem Komputer*, 9(2).
- [4] Hidayat, R., & Suryani, E. (2022). Pemanfaatan Telegram Bot sebagai media antarmuka sistem IoT berbasis mikrokontroler. *Jurnal Teknik Elektro*, 14(1).
- [5] IEEE. (2022). *IEEE standard for software and system test documentation* (IEEE Std 29119-3-2022). IEEE.
- [6] ISO/IEC. (2023). *ISO/IEC 25010:2023 systems and software quality requirements and evaluation (SQuARE)*. ISO.
- [7] Kurniawan, D., & Wibowo, A. (2021). Implementasi sistem Internet of Things berbasis ESP32 untuk monitoring dan kontrol perangkat elektronik. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(3).
- [8] McGrath, M., & O'Flynn, B. (2021). *Embedded systems and IoT design*. Springer.
- [9] Pressman, R. S., & Maxim, B. R. (2022). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- [10] Ramadhan, M., & Setiawan, B. (2023). Perancangan running text LED P10 berbasis ESP32 sebagai media informasi digital. *Jurnal Elektronika dan Otomasi Industri*, 15(1).
- [11] Saputra, R., & Pratama, A. (2022). Pengembangan sistem IoT berbasis ESP32 menggunakan metode Rapid Application Development. *Jurnal Informatika dan Sistem Informasi*, 10(2).
- [12] Sommerville, I. (2021). *Software engineering* (10th ed.). Pearson Education.
- [13] Susanto, A., & Wijaya, R. (2024). Implementasi black box testing pada sistem informasi berbasis Internet of Things. *Jurnal Rekayasa Perangkat Lunak*, 6(1).
- [14] Telegram. (2023). *Telegram bot API documentation*. Telegram Developer Platform.
- [15] Yuliana, N., & Prabowo, D. (2025). Integrasi ESP32 dan aplikasi pesan instan untuk pengendalian perangkat IoT secara real-time. *Journal of Internet of Things and Smart Systems*, 3(1).