ISSN (Online) 2747-0563
Volume 5 No. 2 Tahun 2025

Pengembangan Aplikasi Pengaturan Quickshifter Berbasis Serial dan WebSocket

Erlangga Putra Ramadhan[1*], Mohammad Idhom[2], Firza Prima Aditiawan[3]

[1,3] Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur
[3]firzaprima.if@upnjatim.ac.id

[2] Sains Data, Universitas Pembangunan Nasional "Veteran" Jawa Timur
[2]idhom@upnjatim.ac.id

**Corresponding author email*: 22081010075@student.upnjatim.ac.id

Abstrak*— Perkembangan teknologi komunikasi data dan mikrokontroler mendorong adopsi fitur tambahan pada kendaraan bermotor, termasuk QuickShifter (QS). Pada modul QS berbasis Arduino, penyetelan parameter seperti kill time dan RPM threshold umumnya masih dilakukan melalui serial monitor atau pengubahan kode, sehingga kurang ramah pengguna. Penelitian ini mengembangkan aplikasi pengaturan QuickShifter berbasis komunikasi Serial dan WebSocket dengan antarmuka grafis interaktif untuk memudahkan proses konfigurasi dan pemantauan. Aplikasi dikembangkan menggunakan Visual Studio 2022 dalam bahasa C# (.NET Core) dan diintegrasikan dengan mikrokontroler sebagai penerima data. Metode penelitian mengikuti model Waterfall yang mencakup analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Pengujian fungsional dilakukan pada beberapa skenario utama yaitu koneksi Serial, koneksi WebSocket, pembacaan RPM, pembacaan data konfigurasi, dan pengiriman perubahan setelan, dan seluruh skenario menunjukkan hasil sesuai kriteria fungsional. Aplikasi mampu menampilkan data konfigurasi dan RPM secara real-time serta mengirim perubahan ke modul dengan mekanisme umpan balik dari modul untuk konfirmasi penerimaan. Dengan demikian, sistem yang dikembangkan terbukti meningkatkan kemudahan, fleksibilitas, dan efisiensi proses konfigurasi QuickShifter pada modul berbasis Arduino tanpa memerlukan akses langsung ke kode sumber perangkat keras.

Kata Kunci— **QuickShifter, Serial Communication, WebSocket, Aplikasi Konfigurasi, Mikrokontroler, C#.**

I. PENDAHULUAN

Kemajuan zaman mendorong perkembangan teknologi dalam berbagai macam aspek kehidupan, termasuk aspek transportasi. Transportasi mengalami peningkatan pada bagian efisiensi, kenyamanan, hingga keamanan [1]. Peningkatan pada efisiensi dapat memaksimalkan penggunaan sumber daya dan menghemat alokasi waktu. Kenyamanan yang meningkat akan membawa kualitas hidup manusia pada sektor transportasi ke arah yang lebih baik.

Modernisasi otomotif digunakan untuk mencapai tujuan yang telah disebutkan sebelumnya. Contoh saja munculnya teknologi "kendaraan pintar" yang dapat menyesuaikan variabel pada mesin untuk mencapai efisiensi dan kenyamanan tertinggi menggunakan mikrokontroler pada *Engine Control Unit* (ECU). Mikrokontroler tersebut dapat disetel ulang sesuai dengan kebutuhan pengguna [2].

Fitur yang dimiliki oleh kendaraan akan disesuaikan dengan harga jual suatu kendaraan bermotor, sehingga semakin murah harga kendaraan bermotor, semakin sedikit fitur yang dimiliki, begitu juga sebaliknya [3]. Untuk menyiasati hal tersebut, banyak konsumen yang melakukan modifikasi pada kendaraan mereka agar memiliki suatu fitur tambahan dengan harga yang lebih murah, seperti menambah fitur Quickshifter pada sepeda motor kopling menggunakan modul tambahan.

Guna memaksimalkan QuickShifter, diperlukan penyesuaian yang sama seperti pada ECU [4]. Pada modul QuickShifter berbasis Arduino, proses penyetelan parameter seperti kill time dan RPM *threshold* umumnya masih dilakukan secara manual melalui serial monitor atau pengubahan kode program secara langsung. Oleh karena itu, penelitian ini mengembangkan aplikasi pengaturan QuickShifter berbasis komunikasi Serial dan WebSocket menggunakan Visual Studio 2022 dengan bahasa pemrograman C#.

II. TINJAUAN PUSTAKA

A. *Quickshifter*

Quickshifter merupakan fitur untuk melakukan pergantian gigi ke atas pada sepeda motor kopling tanpa harus menekan kopling ataupun menutup gas. Pergantian gigi dapat dilakukan karena proses pembakaran pada mesin akan terjeda melalui pemutusan pengapian ataupun pemutusan suplai bahan bakar untuk mengurangi beban pada transmisi [5], [6]. Fitur ini biasa ditemukan pada sepeda motor dengan harga yang lumayan mahal (di atas 80 juta rupiah).

B. *Serial Communication*

Serial Communication adalah sebuah cara komunikasi antar perangkat elektronik dengan mengirimkan data bit per bit secara berurutan [7]. Agar komunikasi dapat terjalin dengan lancar, kedua perangkat diwajibkan berada di *baud rate* yang sama [8]. Komunikasi ini sering digunakan pada perangkat IoT.

C. *WebSocket*

WebSocket adalah salah satu dari beberapa protokol komunikasi yang dapat digunakan pada jaringan Wi-Fi dan sering disebut bagian dari HTML5 [9]. Protokol ini memungkinkan untuk melakukan pertukaran data secara *real-time* tanpa perlu melakukan *request* dan *response* sehingga

ISSN (Online) 2747-0563
Volume 5 Nomor 2 Tahun 2025

memiliki latensi yang lebih rendah dibandingkan HTTP Polling [10], [11]. Cocok digunakan untuk pertukaran data intensif.

D. Visual Studio

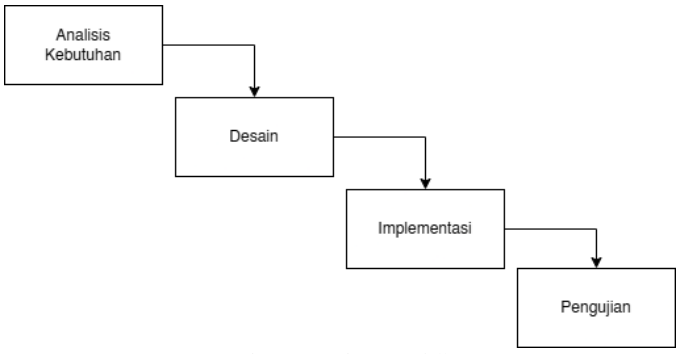
Visual Studio merupakan aplikasi *Integrated Development Environment* (IDE) yang dikembangkan oleh Microsoft untuk membantu para pengembang mengembangkan suatu aplikasi. IDE ini lebih ditujukan untuk mengembangkan aplikasi yang memiliki tampilan grafis (GUI). Mendukung berbagai macam bahasa pemrograman seperti Python, C#, Javascript, dan lain sebagainya [12].

E. C#

C# adalah bahasa pemrograman yang termasuk ke dalam keluarga C dan dikembangkan oleh Microsoft sebagai bagian dari platform .NET. Bahasa ini diciptakan untuk melanjutkan tongkat estafet dari C++ sekaligus bersaing dengan bahasa pemrograman modern lainnya [13]. C# hadir dengan berbagai peningkatan dan fitur baru yang lebih sederhana, aman, serta mendukung konsep Object-Oriented Programming (OOP) secara penuh untuk mempermudah pengembangan aplikasi yang terstruktur dan efisien..

III. *METODE PENELITIAN*

Penelitian ini termasuk dalam kategori penelitian rekayasa (engineering research) dengan pendekatan eksperimen. Fokus utama penelitian adalah merancang, membangun, dan menguji aplikasi konfigurasi QuickShifter yang berfungsi untuk mengatur parameter sistem QS melalui dua jalur komunikasi, yaitu Serial (UART) dan WebSocket.



Gbr. 1 Metode *Waterfall*

Penelitian ini memakai metode *Waterfall* yang memiliki alur linier, seperti yang terdapat pada Gbr. 1. Secara garis besar, metode *Waterfall* memiliki tahapan sebagai berikut:

A. Analisis Kebutuhan

Kebutuhan yang harus dipenuhi oleh aplikasi ini akan ditentukan pada bagian ini. Berdasarkan pada observasi lapangan dan juga studi literatur. Hasil dari analisis akan digunakan sebagai dasar dalam pembuatan desain aplikasi agar aplikasi yang dibuat sesuai dengan kebutuhan.

B. Perancangan Sistem (Desain)

Tahapan ini akan berfokus pada perancangan aplikasi untuk memenuhi kebutuhan yang telah ditentukan pada tahapan analisis. Perancangan ini akan mencangkum pembuatan diagram alur aplikasi dan desain antarmuka.

C. Implementasi

Aplikasi direalisasikan pada tahap ini. Pembuatan aplikasi dilakukan sesuai dengan rancangan awal yang telah disusun untuk memenuhi kebutuhan. Aplikasi dibuat menggunakan Visual Studio 2022 dengan bahasa pemrograman C# dengan tambahan .NET Core.

D. Pengujian

Pengujian dilakukan untuk memastikan bahwa aplikasi yang telah dibuat dapat memenuhi kebutuhan. Koneksi antar aplikasi dan perangkat diuji coba pada tahap ini. Aplikasi harus dapat melakukan komunikasi dua arah untuk melakukan pemantauan dan penyetelan.

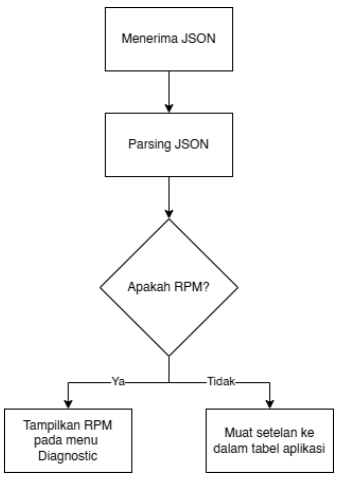
IV. HASIL DAN PEMBAHASAN

A. Kebutuhan

Aplikasi diharuskan untuk dapat melakukan parsing JSON dari modul untuk menerima data setelan yang terpasang pada modul. JSON juga digunakan untuk mengirim data RPM mesin yang berfungsi sebagai pemantauan. Aplikasi juga harus dapat menampilkan RPM mesin terkini berdasarkan hasil parsing tersebut. Aplikasi harus dapat mengirim perubahan data ke modul. Dan yang paling penting, aplikasi dan modul harus dapat terhubung melalui USB serial dan WebSocket.

B. Diagram Alur

Proses dari aplikasi akan dibedakan menjadi dua, yaitu proses masukan dan proses luaran.

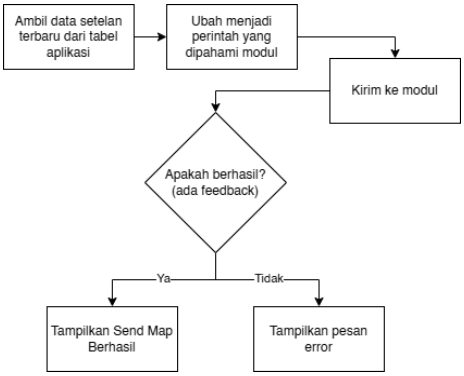


Gbr. 2 Diagram Alur Masukan

Dapat dilihat pada Gbr. 2, proses masukan akan memeriksa terlebih dahulu adanya data JSON yang diterima. Data yang

ISSN (Online) 2747-0563
Volume 5 Nomor 2 Tahun 2025

JSON yang diterima kemudian diparsing dan diperiksa jenisnya untuk memastikan isi serta tujuannya. Terdapat dua kemungkinan jenis data JSON yang dikirim oleh modul, yaitu data setelan yangsaat ini terpasang pada modul atau data bacaan RPM terkini yang dikirim secara berkala. Berdasarkan hasil parsing tersebut, aplikasi akan mengolah dan menampilkan data pada komponen antarmuka yang sesuai, sehingga pengguna dapat dengan mudah memahami informasi yang disajikan.



Gbr. 3 Diagram Alur Luaran

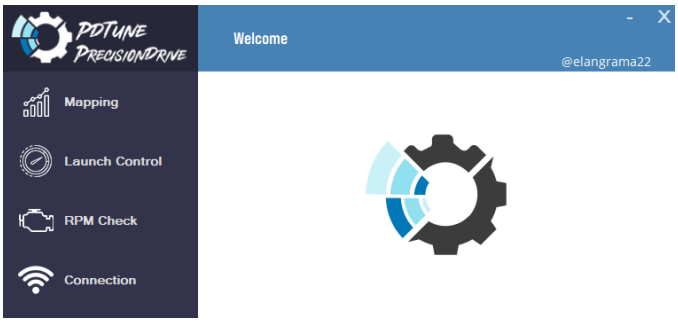
Sedangkan untuk Gbr. 3, proses luaran dari aplikasi berupa perintah yang dapat dipahami oleh modul untuk merubah setelan modul. Ketika tombol kirim ditekan, aplikasi akan melakukan pengambilan data dari tabel aplikasi lalu mengubahnya ke perintah yang dipahami oleh modul. Perintah akan dikirimkan satu per satu ke modul. Jika data diterima oleh modul secara utuh, modul akan memberikan umpan balik bahwa data selesai diterima. Dari umpan balik tersebut, aplikasi akan menampilkan pesan berhasil atau pesan error ketika pengiriman setelan gagal.

C. Implementasi

Aplikasi berhasil dibuat menggunakan bahasa C# pada Visual Studio 2022. Aplikasi terdiri dari beberapa bagian seperti berikut:

1. Welcome

Saat aplikasi dibuka, akan masuk ke halaman *welcome* yang menampilkan logo dari aplikasi. Tampilan dari bagian ini dapat dilihat pada Gbr. 4.

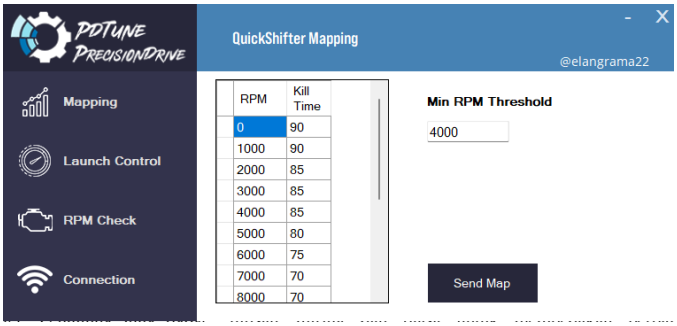


Gbr. 4 Halaman *Welcome*

Halaman ini dibuat untuk menyapa pengguna saat pertama kali dibuka dengan tujuan agar lebih ramah dan humanis

2. Mapping

Pada menu ini, pengguna dapat melakukan perubahan terhadap waktu *kill-time* pada modul QuickShifter sesuai dengan kebutuhan pengendara. Seperti yang ditunjukkan pada Gbr. 5, pengguna juga memiliki opsi untuk mengatur nilai RPM minimum agar fitur QuickShifter dapat aktif hanya pada putaran mesin tertentu. Dengan adanya pengaturan ini, pengguna dapat menyesuaikan respons dan karakteristik dari QuickShifter kendaraan.

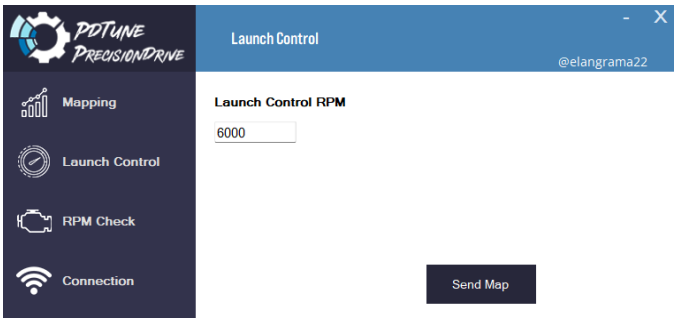


Gbr. 5 Menu *Mapping*

Jika pengaturan telah selesai, pengguna dapat melakukan *Send Map* untuk menyimpan perubahan ke modul.

3. Launch Control

Launch control merupakan fitur untuk menahan RPM mesin pada RPM tertentu ketika kopling ditekan. Bertujuan untuk melakukan *launch* cepat tanpa kehilangan traksi ataupun *wheelie*.



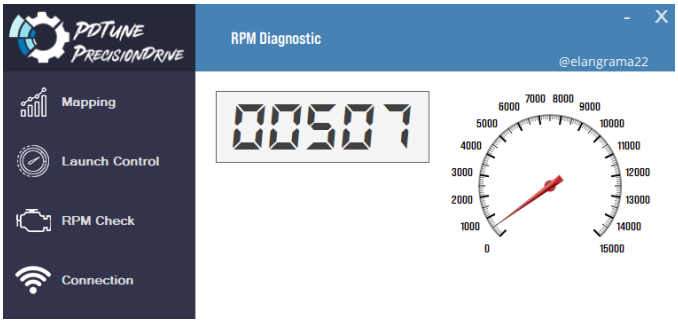
Gbr. 6 Menu *Launch Control*

Ketinggian RPM *launch control* dapat disesuaikan pada menu ini. Dapat dilihat pada Gbr. 6.

4. RPM Check

Agar penyetelan *kill-time* dapat dilakukan dengan lebih akurat, disediakan menu RPM *Check* yang berfungsi untuk menampilkan posisi atau nilai RPM mesin terkini secara real-time. Dengan adanya fitur ini, pengguna dapat memantau kondisi mesin saat melakukan penyetelan, sehingga pengaturan kill-time dapat disesuaikan secara presisi

ISSN (Online) 2747-0563
Volume 5 Nomor 2 Tahun 2025

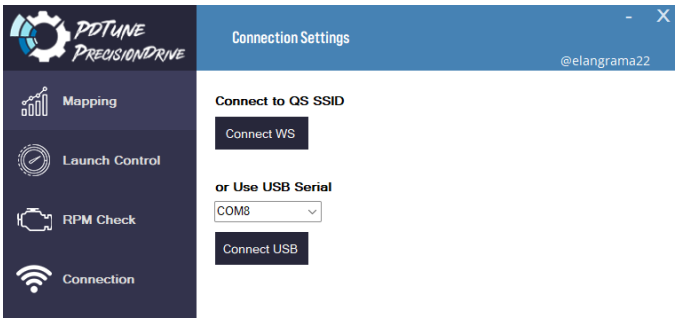


Gbr. 7 Menu RPM Check

RPM ditampilkan dalam bentuk analog dan digital seperti pada Gbr. 7.

5. Connection

Sebelum melakukan penyetelan pada modul, aplikasi dapat dihubungkan terlebih dahulu dengan modul melalui menu ini.



Gbr. 8 Menu *Connection*

Pada menu ini terdapat pilihan untuk melakukan koneksi ke modul melalui SSID (WebSocket) atau melalui USB Serial, seperti pada Gbr. 8.

D. Pengujian

Pengujian dilakukan untuk memastikan bahwa kebutuhan yang telah dirumuskan sebelumnya berhasil dicapai oleh aplikasi. Agar lebih mudah untuk dipahami, hasil dari pengujian akan disajikan dalam bentuk tabular. Sehingga hasil uji dari aplikasi yang telah dibuat dapat dilihat pada Tabel 1 sebagai berikut:

TABEL I
HASIL PENGUJIAN

ID Tes	Pengujian	Hasil yang Diharapkan	Status
1	Koneksi Serial	Aplikasi dapat berkomunikasi dengan modul melalui Serial Communication	Berhasil
2	Koneksi WebSocket	Aplikasi dapat berkomunikasi dengan modul melalui WebSocket	Berhasil
3	Menampilkan RPM	Aplikasi dapat menampilkan RPM mesin pada menu digital dan analog	Berhasil
4	Membaca data yang tersimpan pada modul	Aplikasi dapat menampilkan nilai tabel yang tersimpan pada modul	Berhasil
5	Mengirim perubahan setelan ke modul	Aplikasi dapat mengirimkan perintah ubah setelan yang dipahami oleh modul	Berhasil

Berdasarkan Tabel 1, dapat diketahui bahwa aplikasi telah berhasil memenuhi seluruh kebutuhan yang telah ditetapkan sebelumnya. Selain itu, aplikasi juga mampu melewati serangkaian tahapan pengujian dengan hasil yang menunjukkan bahwa aplikasi berjalan sesuai dengan rancangan awal tanpa mengalami kendala yang berarti.

V. KESIMPULAN

Aplikasi pengaturan QuickShifter berbasis komunikasi Serial dan WebSocket yang dikembangkan pada penelitian ini berhasil direalisasikan dengan baik. Aplikasi mampu melakukan pengaturan parameter utama seperti nilai RPM aktif, waktu pemutusan pengapian, dan pembacaan RPM *real-time* melalui antarmuka. Proses komunikasi dua arah antara aplikasi dan mikrokontroler juga berjalan stabil, baik menggunakan koneksi Serial maupun WebSocket, sehingga memberikan fleksibilitas bagi pengguna dalam melakukan konfigurasi. Secara keseluruhan, hasil penelitian menunjukkan bahwa sistem yang dirancang dapat berfungsi sesuai tujuan dan mampu meningkatkan efisiensi serta kemudahan dalam proses pengaturan QuickShifter.

UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih yang sebesar-besarnya kepada seluruh pihak yang telah memberikan dukungan dan

ISSN (Online) 2747-0563
Volume 5 Nomor 2 Tahun 2025

bantuan selama proses penyusunan penelitian ini berlangsung. Ucapan terima kasih khusus disampaikan kepada pihak SANTIKA yang telah menyediakan template dan panduan penulisan sehingga penyusunan penelitian ini dapat terlaksana dengan baik. Terima kasih juga kepada semua pihak yang secara langsung maupun tidak langsung telah memberikan dukungan, motivasi, serta masukan yang berharga selama proses penelitian ini. Akhir kata, penulis berharap penelitian ini dapat memberikan manfaat dan menjadi referensi bagi pihak-pihak yang membutuhkan.

REFERENSI

[1] M. Mnyakin, "Applications of AI, IoT, and Cloud Computing in Smart Transportation: A Review".

[2] F. H. Pratama, A. Abidin, dan M. H. Bahri, "Analisis Performa Sepeda Motor Sistem Injeksi 110 CC Menggunakan ECU Standar dan ECU Standar Remap," *J. Eng. Sci. Technol.*, vol. 2, no. 2, hlm. 46–52, Mei 2024, doi: 10.47134/jesty.v2i2.25.

[3] S. Ou *dkk.*, "Relationships between Vehicle Pricing and Features: Data Driven Analysis of the Chinese Vehicle Market," *Energies*, vol. 13, no. 12, hlm. 3088, Jun 2020, doi: 10.3390/en13123088.

[4] V. Mamtani, S. Agrawal, M. U. Malviya, dan S. K. Jindal, "Experimental Automatic Gear Shifting for a Combustion Race Car," dalam *2020 International Conference on Interdisciplinary Cyber Physical Systems (ICPS)*, Chennai, India: IEEE, Des 2020, hlm. 20–24. doi: 10.1109/ICPS51508.2020.00010.

[5] K. A. D. R. Shetkar, T. S. Pethker, S. S. Savant, dan N. R. Vishwakarma, "Implementation of automatic manual transmission for a two wheeler," *Proc. Inst. Mech. Eng. Part J. Automob. Eng.*, vol. 236, no. 5, hlm. 1040–1057, Apr 2022, doi: 10.1177/09544070211026521.

[6] R. A. Himawan dan N. I. Prasetya, "PEMBUATAN MODUL QUICKSHIFTER MENGGUNAKAN ARDUINO UNO SEBAGAI PENGGANTI KOPLING MOTOR".

[7] H. Jian, "Research of Serial Communication Based on STM32," dalam *Proceedings of the 7th International Conference on Education, Management, Information and Computer Science (ICEMC 2017)*, Shenyang, China: Atlantis Press, 2017. doi: 10.2991/icemc-17.2017.38.

[8] "Serial Communication by Using UART." Diakses: 29 Oktober 2025. [Daring]. Tersedia pada: http://ethesis.nitrkl.ac.in/65/1/10407023.pdf

[9] I. Cosmina, R. Harrop, C. Schaefer, dan C. Ho, "WebSocket," dalam *Pro Spring 5*, Berkeley, CA: Apress, 2017, hlm. 751–772. doi: 10.1007/978-1-4842-2808-1_17.

[10] N. Mitrovic, M. Eordevic, S. Veljkovic, dan D. Dankovic, "Implementation of WebSockets in ESP32 based IoT Systems," dalam *2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, Nis, Serbia: IEEE, Okt 2021, hlm. 261–264. doi: 10.1109/TELSIKS52058.2021.9606244.

[11] K. E. Ogundeyi dan C. Yinka-Banjo, "WebSocket in real time application," *Niger. J. Technol.*, vol. 38, no. 4, hlm. 1010, Des 2019, doi: 10.4314/njt.v38i4.26.

[12] M. A. Saragih, R. A. Khairiyah, dan S. V. Siahaan, "Perancangan Aplikasi Penjadwalan Daur Ulang Sampah Berbasis Visual Studio 2010".

[13] J. Bishop, R. N. Horspool, dan B. Worrall, "Experience in integrating Java with C# and .NET," *Concurr. Comput. Pract. Exp.*, vol. 17, no. 5–6, hlm. 663–680, Apr 2005, doi: 10.1002/cpe.858.