

Ulusan Penelitian Sistem Operasi Waktu Nyata di Indonesia

Review of Real-Time Operating System Research in Indonesia

Esther Sanda Manapa¹, Eliyah Acantha M Sampetoding², Tommy Wijaya Sagala³, Hoxy Ryner Taluay²

¹Transportasi, Sekolah Pascasarjana, Universitas Hasanuddin, Makassar

²Teknik Informatika, STIK Rajawali, Talaud

ABSTRAK

Selama 30 tahun terakhir, kemajuan dalam pembuatan manufaktur semi konduktor telah menghasilkan komputer yang lebih kecil dan lebih cepat. Kehadirannya menghasilkan pengembangan perangkat yang kompleks dan memproses data dalam jumlah besar dengan cepat dan handal. Semakin banyak fungsionalitas pada perangkat sistem, diperlukan mengelola sumber daya yang efektif dan efisien. Dalam beberapa kasus, hanya sistem operasi waktu nyata (RTOS) yang dapat memenuhi batasan waktu dan sumber daya dari sistem tersebut. Baik penggunaan software pada perangkat otomatisasi, atau level middleware untuk membantu berkomunikasi antar program, atau penggunaan perangkat lunak, akan bermanfaat untuk mengetahui kapan RTOS menggunakan platform yang tepat untuk aplikasi. Makalah ini dan fokus pada berbagai penelitian dan pengembangan RTOS selama sepuluh tahun terakhir yang terdapat di Indonesia. Tujuan utama adalah memungkinkan pembaca untuk memahami konsep dasar sistem operasi waktu nyata dan merangsang penelitian lebih lanjut di Indonesia mengenai sistem operasi waktu nyata pada perangkat otomatisasi.

Keywords: Sistem Operasi Waktu-Nyata, Sistem Waktu-Nyata

ABSTRACT

Over the past 30 years, development of semiconductor manufacturing has resulted in smaller and faster computers. Its presence results in the development of sophisticated devices and processing large amounts of data quickly and reliably. More and more functionality on the system devices, needed effective and efficient resources. In some cases, only real-time operating systems (RTOS) can gather time and resources from these systems.. Whether using the software on an automation device, or middleware level to help communicate between programs, or using the software, it will be useful to know when RTOS uses the right platform for the application. This paper focus on various RTOS research and development over the past ten years in Indonesia. The main objective is to enable readers to understand the basic concepts of real time operating systems and stimulate further research in Indonesia regarding real time operating systems on automation devices.

Keywords: Real-Time Operating System, Real-Time System

1. PENDAHULUAN

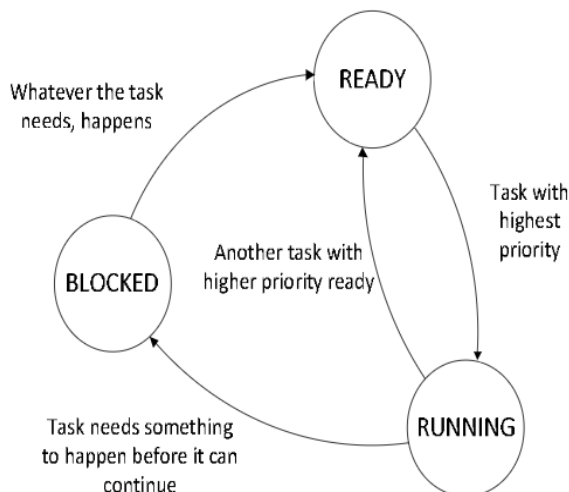
Sistem waktu-nyata (*Real time systems*) adalah sistem yang harus menghasilkan respon yang tepat dalam batas waktu (*deadline*) yang ditentukan. Sistem waktu-nyata merespon secara logis yang didasarkan pada kebenaran hasil keluaran sistem dan ketepatan waktu hasil yang dikeluarkan. Sistem waktu-nyata mulai dikembangkan pada tahun 1950, dengan konsumen saat itu adalah industri militer [1]. Berikut contoh aplikasi yang membutuhkan sistem waktu-nyata misalnya motor, kontrol industri, lampu merah transportasi dan

peralatan telekomunikasi. Sistem Operasi pertama kali diluncurkan pada tahun 1970 dengan nama GUI (*Graphical User Interface*) yang dikembangkan oleh peneliti PARC [2].

Sistem Operasi adalah perangkat lunak yang bertugas mengendalikan perangkat keras secara langsung dan sebagai jembatan antara program aplikasi dan perangkat sistem komputer. Sistem Operasi bertujuan menyediakan lingkungan perangkat lunak yang efektif dan nyaman untuk proram aplikasi dan pengguna (*user*) [3]. Sistem Operasi bertujuan untuk mempermudah penggunaan sistem komputer oleh

programmer, memberikan layanan (services) bagi program aplikasi untuk memanfaatkan sumber daya komputer dan mengusahakan agar sumber daya sistem komputer dapat digunakan secara efisien [4]. Ada terdapat beberapa tipe Sistem Operasi yakni komputer notebook atau komputer desktop, server jaringan, sistem tertanam atau komputer kecil.

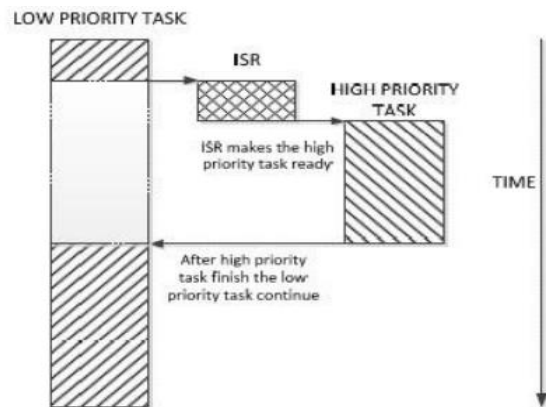
Pada perancangan sistem tertanam (*embedded system*) dimasukan perangkat lunak sistem operasi yang disebut *real time operating sistem* (RTOS) [1]. Pada RTOS terdiri dari manajemen proses, manajemen *memory*, sistem *file*, sistem komunikasi jaringan, dan sistem pengamanan [3, 5].



Gambar 1. Siklus state pada RTOS [7]

RTOS adalah sistem operasi yang bekerja dalam waktu nyata, dengan batasan deterministik yang membutuhkan penggunaan waktu dan daya yang efisien untuk memproses data masuk dan menyampaikan hasil yang diharapkan tanpa *delay* yang tidak diketahui atau tidak terduga [1, 3]. Perangkat lunak RTOS bergantung pada waktu, artinya harus memproses *input* dan menawarkan *output* dalam waktu singkat yang ditentukan sebelumnya. Hal paling penting dari RTOS adalah bahwa permintaan dan respons terhadap data dijamin akan dilaksanakan [6]. Misalnya ChibiOS memiliki permintaan dan respons panggilan yang cepat 90% dari waktu, namun 10% sisanya dari waktu permintaan *input* / *output* terlalu lama, maka aplikasi real-time tersebut tidak berfungsi dengan benar [4]. Sehingga, RTOS tidak hanya masalah

kecepatan, tetapi lebih penting bisa diandalkan.



Gambar 2. Prinsip kerja preemptive kernel [7]

2. SISTEM OPERASI WAKTU NYATA

RTOS adalah hasil pengembangan sistem operasi yang diperuntukan untuk operasi *real time*. RTOS bisa diadaptasikan pada bidang otomatisasi dan pengontrolan [8, 6].

2.1 Komponen RTOS

RTOS mencakup banyak komponen.

2.1.1 Scheduler

Penjadwalan adalah elemen utama pada RTOS yang menentukan urutan pelaksanaan tugas yang didasarkan pada skema prioritas. Algoritma penjadwalan yang digunakan adalah *round-robin* [1].

2.1.2 Task

Task adalah sebuah objek yang dapat dieksekusi. Tujuan RTOS yaitu membagi semua pekerjaan dalam aplikasi tersebut menjadi beberapa bagian task [7]. Tiap task merupakan pengulangan, terdapat tiga buah keadaan pada *task* yakni (1) *Running*, merupakan keadaan di mana sebuah task dengan prioritas tertinggi berjalan; (2) *Ready*, merupakan keadaan yang dialami sebuah task jika terdapat sebuah *task* lain sedang running dan *task* yang berada pada *ready* akan melanjutkan pengerjaan *task* yang sempat tertunda oleh *task* yang lebih tinggi prioritasnya; (3) *Blocked*, merupakan keadaan di mana jika sebuah task membutuhkan *event* atau data maka akan masuk ke dalam *blocked* hingga *event* atau data yang dibutuhkan telah tersedia.

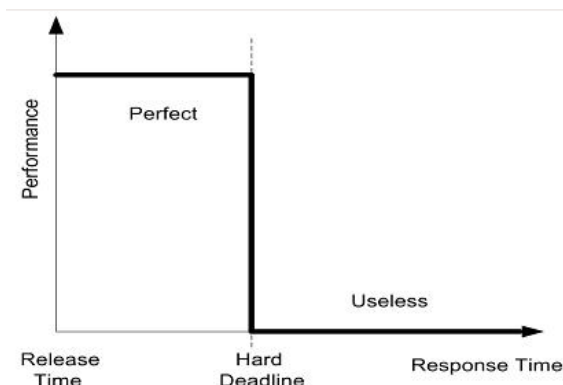
2.1.3 Symmetric Multiprocessing

RTOS harus memiliki kemampuan dalam menangani dan memisahkan beberapa tugas

atau threads sehingga dapat dijalankan pada beberapa inti *processor* sehingga mampu memproses kode secara paralel, contohnya *multitasking*.

2.1.4 Function Library

Fungsi RTOS dapat berisi *application program interface* (API) yang menghubungkan kode aplikasi dan kernel. Setiap entitas kode aplikasi mengarahkan permintaan ke kernel melalui *function library* untuk meminta aplikasi memberikan perilaku terprogram yang diinginkan [9].



Gambar 3. *Hard Real-Time* [3]

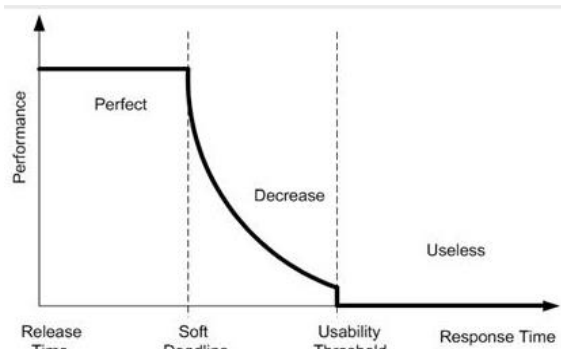
2.1.5 Fast dispatch latency

Latensi pengiriman menunjukkan waktu dari saat sistem operasi mengidentifikasi bahwa suatu tugas telah selesai hingga thread yang siap dijalankan dimulai atau ketika suatu peristiwa apabila adanya tugas prioritas lebih tinggi untuk menghentikan tugas yang sedang berjalan. Cara kerjanya adalah memanggil waktu yang diperlukan yang dijadwalkan untuk beralih dari satu thread yang sedang berjalan ke thread lain. Dalam RTOS, perpindahan waktu harus tetap dapat diandalkan.

2.1.6 User-defined data objects and classes

RTOS menggunakan bahasa pemrograman struktural yang disusun berdasarkan jenis operasinya. Pengguna mendefinisikan suatu objek menggunakan bahasa pemrograman yang ditentukan seperti C++ yang akan digunakan RTOS untuk mengontrol aplikasi yang ditentukan.

2.1.7 Memory management



Gambar 4. *Soft Real-Time* [3]

Memory management diperlukan untuk mengalokasi memori pada setiap program yang akan dijalankan atau objek yang akan dimasukkan dalam memori. Hal tersebut penting karena RTOS tidak seperti sistem operasi umum seperti Windows yang tidak memiliki memori masuk dan memori keluar karena mengarah ke perilaku *non-deterministic*.

2.2 Klasifikasi RTOS

Berdasarkan waktu yang dimiliki, sistem waktu-nyata dibagi menjadi tiga kelas [1], yaitu:

2.2.1 Hard Real-Time

Hard Real-Time adalah sistem yang dibatasi waktu dan deterministik yang merespons dalam kerangka waktu yang ditentukan. Kelas ini ditentukan oleh tenggat waktu, *latency*, dan batasan waktu. Misalnya, jika *output* diharapkan dalam 10 detik, maka sistem harus memproses *input* dan memberikan *ouput* pada detik ke-10. *Output* tidak boleh dirilis pada detik ke-9 atau ke-11 untuk mencegah kegagalan sistem. Sistem penerapannya adalah pada pesawat terbang yang tidak boleh ada kegagalan sistem kendali yang dapat menyebabkan korban jiwa.

2.2.2 Soft Real-Time

Semua sistem komputer dapat dikategorikan dalam kelas *Soft Real-Time*. Contohnya adalah saat melakukan permainan game online. Delay selama setengah detik tidak akan menyebabkan kegagalan yang fatal hingga menimbulkan korban jiwa, tetapi akan menggaung kenyamanan pengguna yang sedang bermain.

2.2.3 Firm Real-Time

Firm Real-Time dibatasi dengan batasan yang sama seperti *Hard Real-Time*, bedanya sistem

masih diberikan toleransi beberapa kali jika tidak memenuhi batasan tersebut, tetapi bila melewati dari jumlah toleransi yang telah diberikan maka sistem akan gagal. Contohnya, apabila dosen mengeluarkan peraturan bahwa mahasiswa boleh tidak masuk dalam pertemuan kelas maksimal 5 kali toleransi, tetapi apabila melebihi 5 kali maka mahasiswa itu dianggap gagal dalam mata kuliah dosen tersebut.

2.3 Aplikasi Pada RTOS

RTOS bisa fleksibel tetapi biasanya dirancang untuk tujuan tertentu. Sebagian besar subsistem RTOS memiliki tugas tertentu [1]. RTOS menawarkan sebagian besar solusi operasional, termasuk aplikasi seperti:

2.3.1 Control systems

RTOS digunakan untuk memantau dan menjalankan perintah pada sistem kontrol. RTS digunakan untuk mengontrol aktuator dan sensor untuk fungsi seperti pengontrol digital. Sistem yang dikendalikan termasuk pesawat terbang, rem, dan mesin. Sistem yang dikendalikan dipantau dengan bantuan sensor dan diubah oleh aktuator. RTOS membaca data dari sensor dan kemudian melakukan perhitungan dan menggerakkan aktuator sehingga gerakan dalam pesawat terbang dapat disimulasikan.

2.3.2 Image processing

Komputer, Perangkat handphone, dan kamer harus mencapai tugas yang diinginkan secara tepat waktu, yang berarti input visual diperlukan secara real-time dengan ketepatan terbaik sehingga otomasi industri, misalnya dapat mengontrol apa yang terjadi pada citra satelit kemacetan misalnya. Pemrosesan gambar secara waktu nyata sangat penting



Gambar 5. Flight Simulator

untuk membuat penyesuaian waktu nyata untuk objek bergerak.

2.3.3 Voice over IP (VoIP)

VoIP bergantung pada protokol Internet untuk mengirimkan suara secara tepat waktu. VoIP dapat diimplementasikan pada jaringan IP. Suara digital, dikompresi dan dikonversi ke paket IP secara real time sebelum dikirim melalui jaringan IP, misalnya *voice message* pada MMS.

3. PERTIMBANGAN PEMILIHAN RTOS

Ada banyak pertimbangan ketika akan memilih RTOS yang akan digunakan [10], yakni:

3.1 Performansi

Kemampuan Performansi adalah faktor utama yang harus dipertimbangkan ketika memilih RTOS. Aspek kunci untuk RTOS adalah bahwa keandalannya dapat menjamin permintaan dan tanggapan data terjadi dalam periode waktu tertentu, apapun yang terjadi dalam sistem komputer. Performansi RTOS harus ditentukan oleh ketergantungan sistem dalam melakukan panggilan dalam periode tertentu, terlepas dari apa pun yang terjadi pada sistem.

3.2 Fitur

Setiap RTOS memiliki fitur unik yang menentukan cara kerjanya untuk menjalankan perintah. Ketika memilih RTOS, fitur harus dievaluasi terlebih dulu dengan memilih fitur yang diperlukan. Tujuan memilih fitur yang relevan, agar sistem yang akan dijalankan bisa berjalan lebih efektif. RTOS yang baik harus memiliki sistem perlindungan memori yang efisien.

3.3 Kemudahan

RTOS yang baik harus mudah diatur dan diintegrasikan dengan sistem waktu nyata yang akan dibuat. RTOS yang mudah dipelajari fiturnya akan mudah diintegrasikan dengan aplikasi.

3.4 Middleware

Hampir semua RTOS menampilkan komponen Middleware atau komponen ketiga yang terintegrasi dengan RTOS. Middleware harus dievaluasi dan memastikan bahwa middleware memiliki metode integrasi yang mulus. RTOS harus memiliki dukungan middleware, sehingga proses integrasi aplikasi tidak memakan waktu. Middleware harus memiliki komponen seperti

TCP / IP dan sistem file.

4. PENELITIAN RTOS DI INDONESIA

Selama 10 tahun terakhir (2009 – 2019), hanya terdapat sedikit penelitian, pengembangan dan ujicoba menggunakan RTOS yang dilakukan peneliti Indonesia dan dipublikasi di dalam negeri.

Pamungkas dkk [7], melakukan penerapan ChibiOS RTOS dengan sistem pengontrolan motor DC sebagai kontrol otomatis. Penelitian mendesain dan mendekomposisi multitasking pada perangkat lunak sistem mikrokontroler AVR dengan parameter PI.

Leksono dkk [9], melakukan penerapan ChibiOS RTOS pada Mikrokontroler AVR. Hasil pengujian *task* didapatkan bahwa AVR tipe Atmega128L dapat ditanamkan sebuah RTOS.

Suyoto *et al* (2011) [11] melakukan perancangan dan implementasi sistem komunikasi Wireless pada SBC (Single Board Computer) Alix3d2 untuk sistem komunikasi stasiun cuaca. Sistem tertanam yang digunakan ada uClinux OS.

Kurniawan dkk [12], melakukan penerapan dan mengamati karakteristik CocoOS RTOS yang berhasil dilakukan pada mikrokontroler jenis AVR Atmega16. Pengujian dilakukan pada *Timing Diagram* dan *Clock Tick*.

Putra dkk [13], melakukan implementasi RTOS pada UGMSat-1. CoCoX CoOS adalah RTOS dengan tiga tugas yakni Memeriksa Kondisi, Pembersihan dan Tugas Komunikasi. Hasil percobaan menunjukkan bahwa OBC / OBDH dapat mengelola paket data dan mengirimkan ke Stasiun monitor.

Manapa dkk [14], melakukan Implementasi dan Analisis Chibi RTOS pada Mikrokontroler ARM jenis CORTEX-4 STM32F4 *Discovery*. Pengujian dilakukan pada *Preemptive Kernel*, *Semaphore*, dan *Clock Tick*.

Habibie dkk [15], melakukan penelitian membandingkan komunikasi USART pada mikrokontroler dengan RTOS dan dengan hanya menggunakan *interrupt*. Hasil penelitian menjelaskan bahwa sistem dengan RTOS memiliki waktu ping yang jauh lebih cepat dari

yang menggunakan *interrupt* pada kasus sistem yang memiliki banyak *task*.

Mufida dkk [10], melakukan perancangan sistem monitoring konsentrasi gas karbon monoksida. Sistem monitoring nya menggunakan sensor dan STM32F4 *Discovery*. Sistem monitoring ini dilengkapi *data logger* menggunakan *SD Card* dan komunikasi antar *mobile* menggunakan *SMS Gateway*.

Ramadhan dkk [16], merancang sistem kebocoran gas menggunakan Arduino Uno yang ditanam FreeRTOS. Hasil penelitian menunjukkan sistem dengan RTOS memerlukan waktu yang lebih lama dibandingkan sistem yang tidak menggunakan RTOS.

Hifdzullisan [17], merancang sebuah RTOS berbasis penjadwalan dengan tipe *soft-real time system* untuk mengoperasikan sebuah *embedded system* biomedis berupa elektrokardiogram. Task yang diberikan pada RTOS adalah melakukan pengecekan terhadap elektroda yang digunakan sebagai sensor untuk membaca etangan listrik pada jantung dan ditampilkan dalam Graphical LCD 128x64.

Septiawan [18], melakukan implementasi RTOS ke dalam Arduiono menggunakan FreeRTOS. Hasil pengujian, sistem yang menerapkan RTOS berjalan dari awal program sampai akhir membutuhkan total waktu yang lebih sedikit dibandingkan sistem yang tidak menerapkan RTOS.

Anshori dkk [19], menggunakan RTOS dalam memonitoring tanaman jamur dan penyiraman secara otomatis dengan menggunakan FreeRTOS. Hasil pengujian, RTOS memiliki waktu pelaksanaan lebih lama dari tanpa penggunaan RTOS. Akurasi dan deadline dengan FreeRTOS lebih tepat pada pembacaan sensor.

5. KESIMPULAN

Makalah ini memberikan ulasan tentang Sistem Operasi Waktu-Nyata (RTOS) dan pengembangannya di Indonesia. RTOS yang baru diujicobakan oleh peneliti Indonesia yakni ChibiOS, CocoOS, CoCoX dan FreeRTOS.

Ada banyak RTOS dan aplikasi nyata bisa menggunakan RTOS khususnya seperti pada bidang teknologi SDN-*Software Defined Network* [20], sehingga peluang melakukan penelitian dan pengembangan pada bidang RTOS masih banyak.

DAFTAR PUSTAKA

- P. A. Laplante, *Real-Time Systems Design and Analysis: An Engineer's Handbook*, Wiley-IEEE Press, 1996.
- I. Sudirman dan R. S. Whono, *Sejarah Komputer*, IlmuKomputer.com, 2003.
- E. Ariyanto, "Sistem Operasi Waktu-Nyata," Institut Teknologi Telkom, Bandung, 2010.
- A. Wiedjaja, M. Handi, L. Jonathan, B. Christian dan L. Kristofel, "Kajian dan Implementasi Real Time Operating System Pada Single Board Computer Berbasis ARM," *ComTech: Computer, Mathematics and Engineering Applications*, vol. 5, no. 1, pp. 91-99, 2014.
- T. W. Sagala, E. A. Manapa, V. Y. P. Ardhana dan G. Lewakabessy, "Perbandingan Implementasi Manajemen Pengetahuan pada Berbagai Industri," *JTIM: Jurnal Teknologi Informasi dan Multimedia*, vol. 1, no. 4, pp. 327-335, 2020.
- W. Jatmiko, P. Mursanto, G. Jati, D. M. J. Purnomo, M. R. Alhamidi, N. Habibie dan K. Dwinto, *Real Time Operating Systems (RTOS) Teori dan Aplikasi*, Depok: Fakultas Ilmu Komputer, Universitas Indonesia, 2015.
- H. Pamungkas, I. Setiawan dan B. Setiyono, "Desain dan Implementasi Perangkat Lunak pada Sistem Mikrokontroler Berbasis CHIBIOS/RT (Studi Kasus Pengontrolan Motor DC)," *Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro*, 2011.
- IntervalZero Company, "Real-Time Operating Systems (RTOS)," IntervalZero, [Online]. Available: <https://www.intervalzero.com/products/rtos-platform-vision/>. [Diakses 22 12 2019].
- B. P. Leksono, I. Setiawan dan B. Setiyono, "Penerapan Real Time Operating System (RTOS) Pada Mikrokontroler AVR (Studi Kasus ChibiOS/RT)," *Makalah Seminar Tugas Akhir, Jurusan Teknik Elektro, Universitas Diponegoro*, 2011.
- L. Mufida, *Rancang Bangun Sistem Monitoring Konsentrasi Gas Karbon Monoksida (Co) Sebagai Emisi Gas Buang Menggunakan Sensor Gas MQ-7 Berbasis Mikrokontroler STM32F4 Discovery*, Surabaya: Diploma thesis, Institut Teknologi Sepuluh Nopember, 2017.
- Suyoto, A. Heryana dan Y. Ferdian, "Desain dan implementasi Sistem Komunikasi Wireless Pada SBC Alix3d2," *INKOM - Jurnal Informatika, Sistem Kendali dan Komputer*, vol. 5, no. 1, pp. 9-15, 2011.
- F. R. Kurniawan, I. Setiawan dan Sumardi, "Multitasking Pada Mikrokontroler ATMEGA16 Menggunakan Real Time Operating System (RTOS) Jenis Cooperative," *Makalah Seminar Tugas Akhir, Jurusan Teknik Elektro, Universitas Diponegoro*, 2012.
- E. A. Putra, T. K. Priyambodo dan N. Mestika, "Real-time operating system implementation on OBC/OBDH for UGMSat-sequence," dalam *AIP Conference Proceeding*, 2016.
- E. A. Manapa, E. Ariyanto dan E. M. Jadied, "Implementasi dan Analisis Sistem Operasi Waktu-Nyata CHIBI Pada Mikrokontroler ARM," *e-Proceeding of Engineering, Telkom University*, vol. 3, no. 2, pp. 3672-3680, 2016.
- N. Habibie, M. R. Alhamidi, D. M. J. Purnomo dan M. F. Rachmadi, "Performance Comparison Of Usart Communication Between Real Time Operating Systems and Native Interrupt," *Jurnal Ilmu Komputer dan Informasi*, vol. 9, no. 1, pp. 43-51, 2016.

L. I. Ramadhan, D. Syauqy dan B. H. Prasetio, "Sistem Pendeteksi Kebocoran Gas LPG Menggunakan Metode Fuzzy yang," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 11, pp. 1206-1213, 2017.

J. Hidzullisan, S. Sumaryo dan A. Rizal, "Desain Dan Implementasi Sistem Operasi Waktu Nyata Sederhana Untuk Sistem Tertanam Biomedis Berbasis Penjadwalan," *eProceedings of Engineering, Telkom University*, vol. 5, no. 3, pp. 3851-3859, 2018.

I. W. Septiawan, S. R. Akbar dan S. Dhanial, "Rancang Bangun Sistem Multi-Sensor Untuk Pengukuran Jarak Secara," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 11, pp. 5394-5401, 2018.

M. S. Anshori, S. R. Akbar dan R. Maulana, "Implementasi Sistem Sensor Dan Aktuator Real Time Pada Tanaman," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 2, pp. 1471-1479, 2019.

E. A. Manapa, S. Wahjuni dan S. N. Neyman, "Rancang Bangun Protokol Perutean SDGR+R pada Vehicular AD-HOC Network Berbasis Arah," *Jurnal Teknologi Informasi dan Ilmu Komputer*, 2020.