Pengembangan dan Implementasi RTOS untuk Pembatasan Jumlah Pengunjung pada Ruangan Berbasis Mikrokontroler

Dede Irawan Saputra[1], Fajar Gumilang[2]
[1,2]Program Studi Teknik Elektro, Universitas Jenderal Achmad Yani
Jl. Terusan Jenderal Sudirman PO BOX 148 Cimahi
dedeirawan.saputra@lecture.unjani.ac.id[1], fjrgumilang@gmail.com[2]

Abstrak

Seiirng dengan berkembangnya protocol Kesehatan bagi ruang public banyak kebijakan-kebijakan baru yang harus dipenuhi salah satunya adalah dengan membatasi jumlah pengunjung yang terdapat pada suatu ruangan khususnya jika ada peristiwa berkumpulnya manusia baik itu untuk perayaan, sekolah, atau hiburan semata. Untuk memudahkan salah satu pemenuhan dari kebijakan tersebut maka sistem peghitung pengunjung dan pembatasan pada ruangan diperlukan. Pada penelitian ini sistem itu telah berhasil dikembangkan dan dibuat dengan mengimplentasikan sistem kerja waktu nyata atau *Real Time Operating System*(RTOS). FreeRTOS dipilih pada penelitian ini dikarenakan sifatnya yang moduler, *Open Source*. Sistem RTOS yang diterapkan menggunakan dua buah *Task* yakni *Task* sensor masuk dan *Task* sensor keluar, *Task* 1 befungsi sebagai penghitung pengunjung dan mengontrol motor servo untuk menggerakan gerbang pintu masuk ruangan dan *Task 2* berfungsi untuk menghitung jumlah pengunjung yang keluar. Dari hasil pengujian dan pengamatan bahwasnya dengan menggunakan RTOS sistem akan menepati *deadline* yang ditentukan, jika sistem melebihi *deadline* dan parameter yang ditentukan maka sistem akan menunggu sampai kondisi untuk menjalankan *task* dipenuhi oleh karenanya kegagalan sistem dapat dihindari.

Kata kunci: Pembatasan Pengunjng, Protokol Kesehatan, RTOS, FreeRTOS,

Abstract

Along with the development of health protocols for public spaces, many new policies must be met, one of which is to limit the number of visitors in a room, especially if there is a gathering of people, be it for events, schools, or entertainment. To facilitate the fulfillment of one of these, a visitor counter system and restrictions policy in the room are needed. In this research, the system has been successfully designed and built by implementing a Real Time Operating System (RTOS). FreeRTOS was chosen in this study because of its modularity, Open Source etc. The RTOS system implemented uses two tasks, namely the ingoing task sensor and the outgoing task sensor, Task 1 functions as a visitor counter and controls the servo motor to move the room entrance gate and Task 2 works to count the number of visitors leaving. From the test results and observations that using the RTOS system will meet the specified deadline, if the system exceeds the specified deadline and parameters, the system will wait until the conditions for carrying out tasks are met by system failure can be avoided.

***Keywords**: FreeRTOS, Health Protocol, RTOS ,Visitor Restrictions*

I. PENDAHULUAN

Berkumpul nya manusia pada satu tempat karena suatu peristiwa dalam kondisi pandemi merupakan salah satu faktor terpenting dalam penyebaran COVID-19[1] khususnya pada suatu ruangan publik memiliki protokol tersendiri, seperti pengecekan suhu tubuh, menerapkan pembatasan jarak[2] oleh karenanya untuk mempermudah dalam menerapkan protokol kesehatan yang berlaku untuk suatu ruangan tertutup maka dibutuhkan suatu sistem yang simpel untuk menghitung jumlah manusia yang berada dalam satu ruangan. Sampai pada beberapa waktu lalu protokol pembatasan yang

berlaku pada tempat publik telah dilakukan dengan cara perhitungan manual dan penjagaan pada setiap titik masuk dari suatu ruangan, kendati demikian hal yang terjadi di lapangan menunjukkan bahwa proses berkerumun tetap terjadi yang diakibatkan karena sukarnya menghitung jumlah manusia pada suatu ruangan pada satu peristiwa.

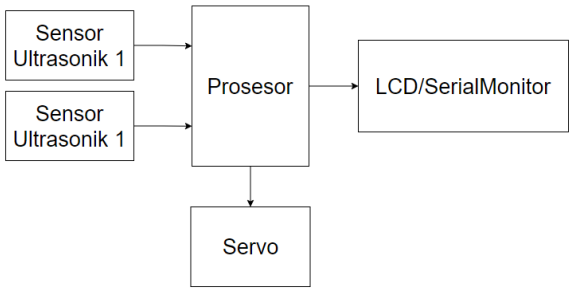
Oleh karenanya penelitian yang dibutuhkan adalah bagaimana cara mengembangkan dan implementasi akan sebuah sistem untuk menghitung dan membatasi jumlah pengunjung pada suatu ruangan yang bersifat simpel mudah dan berskalabilitas tinggi. Secara spesifik penelitian ini untuk menganalisis suatu sistem yang simpel dengan pengimplementasian sistem kerja waktu nyata atau *Real Time Operating System* (RTOS). RTOS merupakan suatu sistem kerja yang menentukan suatu set perkerjaan dalam satu jendela waktu tertentu [3]. Pada penelitian ini dipergunakan FreeRTOS sebagai platform sistem kerja yang digunakan pada mikrokontroller ATMega328, dengan menggunakan RTOS pada sistem penghitung untuk jumlah pengunjung sistem dapat menjalankan dua buah sensor yang dipasang pada dua tempat yang berbeda dan dijalankan secara bersamaan pada satu waktu. Ketika RTOS menjalankan dua tugas tersebut secara bersamaan, sejatinya tugas tersebut disimpan dalam salah satu kondisi yaitu, *running*, *ready*,*block*,*suspended*, jika tugas dimasukkan ke dalam kondisi *running* berarti prosesor sedang dipakai untuk tugas tersebut, lalu jika tugas dimasukkan ke dalam kondisi *ready* maka tugas tersebut siap untuk dijalankan karena tidak sedang diblok atau dalam kondisi *suspended.* Apabila tugas disematkan dalam kondisi *block* maka tugas tersebut menunggu suatu *Trigger*. Terakhir tugas tidak akan keluar atau masuk dalam kondisi *Suspended* jika tidak dipanggil dengan suatu kode [4].

Penelitian ini bertujuan untuk mempermudah pembuatan sistem yang simpel untuk penghitungan dan pembatasan jumlah orang pada suatu ruangan dengan cara mengimplantasikan FreeRTOS pada mikrokontroller ATMega328, seiring dengan meningkatnya relevansi dan penting nya RTOS pada suatu sistem tertanam maka adalah hal yang masuk akal apabila sistem dibuat dengan cara *multitasking*[5]. Penelitian akan RTOS telah banyak dibuat dan perbandingan nya dilakukan seperti oleh Purnomo D et al, yang menemukan bahwa unjuk kerja dari RTOS lebih baik 40% dibandingkan dengan menggunakan fitur *interupt*[6]. Juga penelitian yang dilakukan oleh Maruyama et al yang meneliti akan konsumsi energi yang dipakai oleh RTOS[7] selain daripada itu penelitian yang dilakukan oleh de Oliveira Turci akan implementasi pada alarm kebakaran skala kecil[8] dan pada tahun 2014, Atmadja et al menganalisis RTOS pada sistem tertanam Linux untuk implementasi pada robot mobile[9]

Pada penelitian ini akan dianalisis unjuk kerja daripada RTOS yang memilki dua buah tugas untuk dijalankan secara bersamaan dan juga untuk mempermudah proses pembuatan sistem perhitungan dan pembatasan jumlah manusia pada ruangan.

II. METODE PENELITIAN

Pada penelitian ini ada beberapa komponen yang dibutuhkan diantaranya adalah LCD/Serial Monitor sebagai antarmuka, dua buah sensor ultrasonik sebagai sensor pendeteksi jumlah orang masuk dan keluar terakhir dipilih ATMega 328 sebagai prosesor utama pada sistem ini seperti tertera pada blok diagram pada Gambar 1.



Gambar 1. Blok Diagram Sistem

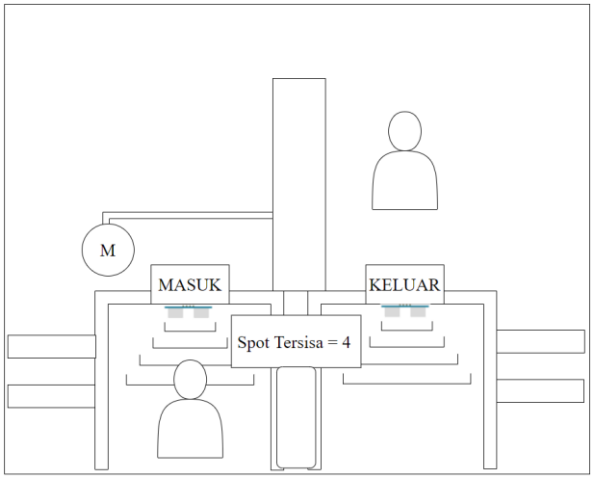
Pada Tabel 1 berisikan spesifikasi dari sensor ultrasonik yang dipakai. Sensor ultrasonik yang dipakai merupakan sensor yang berjenis HC-SR04

Tabel 1. Spesifikasi Sensor HC-SR04

Parameter	Nilai
Tegangan	5V DC
Arus Statis	<2 mA
Level Outpui	5V – 0 V
Sudut Sensor	< 15 Derajat
Jarak Deteksi	2CM – 450 CM
Keakuratan	3mm

Skema pengambilan data untuk implementasi dari sistem terdapat pada Gambar 2 yang mana sistem diaplikasikan pada studi kasus satu ruangan yang memiliki satu pintu masuk dan satu pintu keluar. Sensor dipasang pada masing-masing pintu dan servo digunakan sebagai palang pembatas pintu

masuk lalu jumlah pengunjung pada ruangan ditampilkan pada antarmuka LCD/Serial monitor.



Gambar 2. Skema dari Sistem

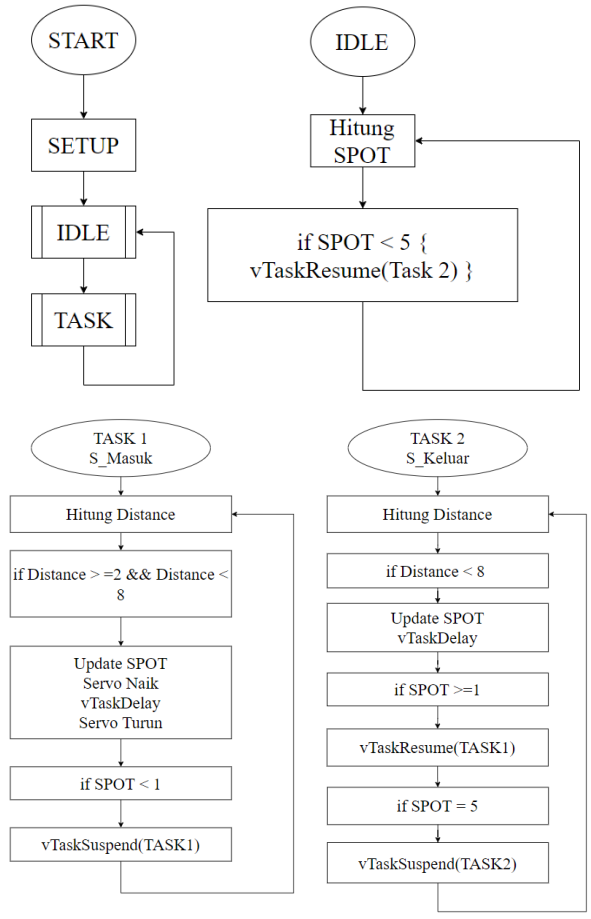
Dalam skema sistem ini mikrokontroller akan diimplemtasikan sistem *multitasking* yang berisikan tugas pengontrolan kedua buah sensor, unjuk kerja dari sistem akan dianalisis ketika mikrokontroller mengeksekusi dua buah tugas secara bersamaan. Sejati nya ada dua buah *Task* yang berjalan bersamaan, pertama sistem harus mengoperasikan motor servo sesuai dengan parameter yang di program motor ini akan bergerak dari nol derajat sampai 90 derajat dimana motor ini menggerakan palang pembatas sebagai pintu masuk motor ini akan bergerak setelah mendeteksi adanya pengunjung pada pintu masuk setelah motor mencapai derajat ke 90 maka beberapa saat kemudian motor akan bergerak dari derajat 90 ke derajat 0 yang mana akan menutup pintu masuk. Pada task kedua sensor di pasang pada pintu keluar disini sensor dipakai hanya untuk menghitung jumlah orang keluar yang mana pada antarmuka akan ditampilkan sebagai penambahan jumlah spot tersisa pada ruangan tersebut.

A. *FreeRTOS*

Penelitian ini pada dasarnya merupakan implementasi freeRTOS pada suatu sistem yang simpel. FreeRTOS digunakan dikarenakan banyak kelebihan dan kemudahan untuk pemrosesan secara waktu nyata *Real Time* dan untuk *Multitasking* selain itu pula sistem ini bersifat *Open Source* dan juga beberapa keuntungan kecil lainya [10].

Dalam FreeRTOS *multitasking* dapat dicapai dengan cara menggunakan beberapa API *Reference* yang tersedia, pada kasus penelitian ini *xTaskCreate*, *xTaskSuspend* dan *xTaskResume* dipakai. Pada *xTaskCreate* ada beberapa parameter yang bisa diatur, dari mulai alokasi memori pada suatu *Task* sampai prioritas eksekusi dalam satu waktu pada tiap tiap *Task* dapat diatur disini. Prioritas diperuntukan sebagain *Scheduler Policy* yang dimilik ioleh freeRTOS, terdapat dua buah algoritma *scheduler policy* yaitu algoritma *Round Robin* dan algoritma *Fixed Priority Preemptive Schedulin.* Akan tetapi pada penelitian ini digunakan *Scheduling Policy* yang mencampurkan kedua buah algoritma tersebut yakni *Priorized Preemptive Scheduling with Time Slicing*. Dikarenakan algoritma ini juga menggunakan *Preemptive Scheduling* oleh karena itu ketika terdapat *task* yang memiliki prioritas tinggi *task* yang memiliki prioritas rendah akan disematkan pada kondisi *ready* sampai *task* berprioritas tinggi selesai di eksekusi, terakhir fitur *time slicing* dimana pada *task* yang berprioritas sama akan mendapatkan waktu pemrosesan yang sama pada CPU, hal ini dapat dicapai menggunakan *tick interrupt* dari freeRTOS[11]. Selanjut nya API *xTask Suspend* dan *xTaskResume* digunakan sebagai mitigasi mekasnisme error untuk sistem jika ada kondisi yang tidak sesuai dengan yang diinginkan.

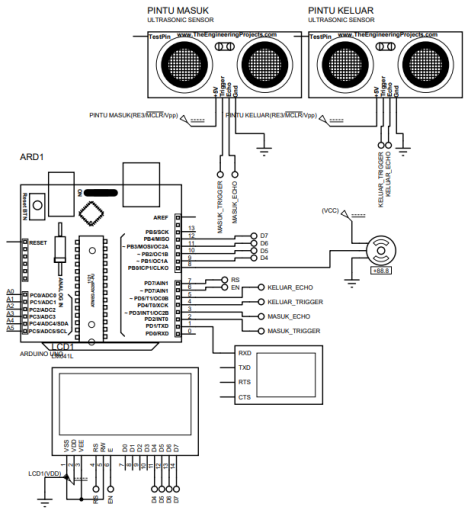
B. Perancangan Domain *Software*



Gambar 3. Diagram Alir dari Sistem

Pada Gambar 3 merupakan diagram alir dari sistem penghitung dan pembatasan jumlah pengunjung menggunakan freeRTOS, terdapat dua buah *task* dan kondisi *idle*. Pertama *Task* 1digunakan sebagai sensor masuk, Kedua *Task* 2 digunakan sebagai sensor keluar dan *Idle* hanya berisikan vTaskResume saja seluruh *Task* diberukan prioritas 2 yaitu prioritas normal dari skala 0-3, dimana skala 3 merupakan skala tertinggi. Program ini akan terus berjalan sampai mikrokontroller tidak dialiri energi listrik.

C. Perancangan Domain Elektronik



Gambar 4. Skematik Sistem

Pada Gambar 4 merupakan skematik pemasangan input output pada sistem dari Gambar 2 didapatkan daftar spesifikasi yang dibutuhkan untuk mengembangkan sistem ini.

III. HASIL DAN PEMBAHASAN

Pada Gambar 4 adalah fungsi API reference yang deprogram pada mikrokontroller sesuai dengan diagram alir yang dibuat dan dibahas sebelumnya dimana terdapat dua buah *Task* yang dibuat menggunakan *xTaskCreate*.

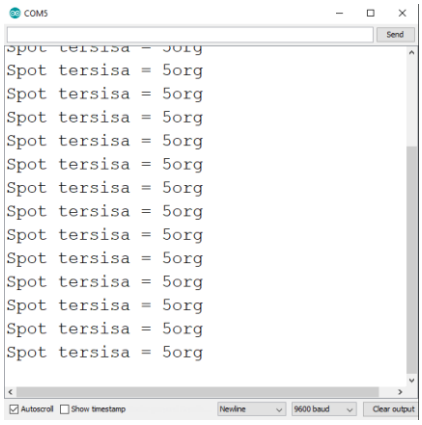
```
// set up Task untuk bekerja.
xTaskCreate(
  TaskSensorMasuk
  ,  "SensorMasuk"   // namaTask
  ,  128  // konfigurasi stack size
  ,  NULL
  ,  2  // Priority. (0-3) 3 paling Tinggi
  ,  &TaskSensorMasuk_handle );

xTaskCreate(
  TaskSensorKeluar
  ,  "SensorKeluar"
  ,  128  // Stack size
  ,  NULL
  ,  2  // Priority
  ,  &TaskSensorKeluar_handle );
```

Gambar 5. Membuat *Task* pada mikroontroller

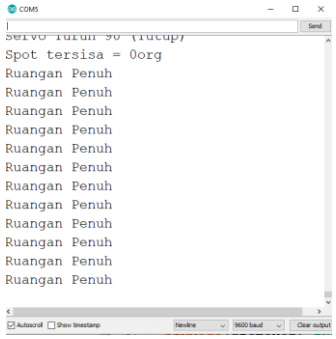
Pada Gambar 5 merupakan miniature sistem dari yang dibuat yang mana hasil akan ditampilkan melalui serial monitor pada mikontroler seperti pada Gambar 6

Gambar 5. Minatur sistem



Gambar 6. Ketersediaan SPOT pada satu ruangan

Pada Gambar 6 Juga menunjukan bahwa dalam program yang dibuat diasumsikan bahwa batas banyaknya orang dalam satu ruangan sebanyak 5 orang dan pada Gambar 7 menunjukan jika tidak ada lagi spot yang tesisa pada ruangan tersebut maka sistem akan menampilkan status atau kondisi ruangan penuh.



Gambar 7. *Display* Status pada saat ruangan penuh

Dengan mengimplentasikan vTaskSuspend dan vTask Resume dalam eksperimentasi ini maka didapatkan hasil yang dapat menjawab masalah apa yang terjadi apabila keadaan ruangan penuh, dari hasil pengamatan yang terjadi adalah jika ruangan penuh maka sensor masuk akan disematkan pada kondisi *Suspended* dimana *Task* tersebut akan tidak aktif atau tidak dapat di eksekusi yang mengakibatkan terdiamnya motor servo atau gerbang dalam keadaam terkunci/tertutup, selanjutnya apabila ada pengunjng yang keluar dan terdeteksi oleh sensor keluar maka *Task 1* akan Kembali bisa aktif sesuai dengan perancangan pada diagram alir sebelumnya pada Gambar 3. Demikian pula untuk senor keluar jika keadaan riangan kosong atau SPOT yang tersisa adalah sama dengan set point maka yang terjadi adalah *Task* 2 akan disematkan pada kondisi *Suspended* sampai ada pengunjung yang masuk.

IV. KESIMPULAN

Dari penelitian yang dilakukan dan juga pengujian hasil penelitian ditemukan bahwa dengan menggunakan freeRTOS maka pembuatan sistem yang bersifat waktu nyata dan *mulstifask* atau bekerja seacara bersamaan dapat mudah diimplentasikan pada mikorokontroller. Dengan menggunakan API reference yang tersedia maka sistem dapat mudah untuk diadaptasi jika ada penambahan fitur untuk penelitian selanjutnya.

Untuk pengembagan selanjutnya ialah dengan menambahkan fitur *Low Power* mengingat sistem ini harus bekerja secara independent. Dan juga untuk menambahkan sensor *Thermal* demi terjaga nya protocol Kesehatan yang berlaku yang di edarkan oleh kementrian Kesehatan.

UCAPAN TERIMA KASIH

Dengan selesainya penelitian ini penulis ucapkan terima kasih pada seluruh pihak yang telah membantu dalam kelancaran proses penelitan ini.

REFERENSI

[1] Domènech-Montoliu S, Pac-Sa MR, Vidal-Utrillas P, Latorre-Poveda M, Del Rio-González A, et al. (2021) “Mass gathering events and COVID-19 transmission in Borriana (Spain): A retrospective cohort study”. PLOS ONE 16(8): e0256747. https://doi.org/10.1371/journal.pone.0256747.

[2] KMK No. HK 01 07-MENKES-328-2020 ttg Panduan Pencegahan Pengendalian COVID-19 di Perkantoran dan Industri.

[3] S.L. Tan, & T.N.B. Anh, “Real-time operating system (RTOS) for small (16-bit) Microcontroller” In Proceeding of The 13th IEEE International Symposium on Consumer Electronics, pp. 1007-1011, 2009 (2018) The IEEE website. [Online]. Available: http://www.ieee.org/

[4] Barry, R. (2016). Mastering the FreeRTOS™ Real Time Kernel A Hands-On Tutorial Guide. © Real Time Engineers.

[5] Carbone J (2015) Multitasking—Essential to Any RTOS. https://www.electronicdesign.com/technologies/embedded-revolution/article/21800584/multitaskingessential-to-any-rtos. Accessed 9 Sep 2021

[6] Purnomo, D., Alhamidi, M., Jati, G., Habibie, N., Hardjono, B., & Wibisono, A. (2015). Comparative Study Of Rtos And Primitive Interrupt In Embedded System. Jurnal Ilmu Komputer dan Informasi, 8(1), 35-44. doi:https://doi.org/10.21609/jiki.v8i1.282.

[7] N. Maruyama, T. Ishihara, H. Yasuura, “An RTOS in Hardware for Energy Efficient Software-based TCP/IP Processing” In Proceeding of 8thIEEE Symposium on Application Specific Processor (SASP), pp. 58-63, 2010

[8] de Oliveira Turci, Luca. (2017). Real-Time Operating System FreeRTOS Application for Fire Alarm Project in Reduced Scale. International Journal of Computing and Digital Systemss. 6. 198-204. 10.12785/IJCDS/060405.

[9] W. Atmadja, B. Christian, L. Kristofel, “Real Time Operating System on Embedded Linux with Ultrasonic Sensor for Mobile Robot” In Proceeding of IEEE International Conference on Industrial Automation and Information & Comunication Technology (IAICT), pp. 22-25, 2014

[10] Real Time Engineers Ltd, FreeRTOS, http://www.freertos.org., retrieved March 3, 2021

[11] FreeRTOS scheduler: Learn to CONFIGURE scheduling algorithm. In: Microcontrollers Lab. https://microcontrollerslab.com/freertos-scheduler-learn-to-configure-scheduling-algorithm/. Accessed 9 Sep 2021