

Implementasi *Fault Tolerant System* menggunakan *Self-purging Redundancy* pada Sistem Monitoring Suhu Ruang Server

Alfan Rafi'uddin Ardhani¹, Agung Setia Budi², Fitri Utaminigrum³

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹alvanrafi@gmail.com, ²agungsetiabudi@ub.ac.id, ³f3_ningrum@ub.ac.id

Abstrak

Server merupakan sebuah sistem komputer yang melayani dan mengontrol akses *client* yang terhubung dengannya. *Server* umumnya ditempatkan pada suatu ruang khusus yang dilengkapi dengan pendingin ruangan. Hal tersebut dilakukan untuk menjaga suhu dari komponen *server* yang menghasilkan panas ketika bekerja. *Server* yang bekerja pada suhu tinggi dapat menyebabkan penurunan kinerja *server* dan bila dibiarkan dapat menyebabkan padamnya *server* hingga kerusakan pada komponennya. Pada beberapa penelitian yang telah dilakukan, penerapan *fault tolerant system* pada sistem monitoring suhu ruang *server* belum banyak digunakan dimana sistem hanya bergantung pada sebuah sensor untuk mendapatkan data suhu sehingga rentan terhadap terjadinya kesalahan. Oleh karena itu, pada penelitian ini digunakan metode *self-purging redundancy* untuk mengetahui pengaruh metode tersebut terhadap penanganan kesalahan pada sistem. Pada penelitian ini digunakan 3 buah modul redundan berupa Arduino Nano dan sensor suhu DHT-22 untuk mendapatkan data suhu yang selanjutnya dikirimkan ke sebuah voter berupa Arduino Uno. Dari penelitian yang telah dilakukan, didapatkan bahwa metode *self-purging redundancy* memiliki pengaruh terhadap penanganan kesalahan pada sistem. Dimana sistem dapat menoleransi kesalahan yang terjadi pada sebuah modul redundan dalam satu waktu. Selain itu, sistem memiliki nilai akurasi sebesar 85% dan rata-rata waktu komputasi sebesar 14,8 milidetik.

Kata kunci: *server, fault tolerant system, self-purging redundancy, sistem monitoring suhu, DHT-22*

Abstract

Server is a computer system that serves and controls access to clients that connected to it. Server is generally placed in a special room equipped with air conditioning. This is done to keep the temperature of server's components that generate heat when working. Server that work at high temperatures may cause a decrease in performance and if left unchecked can cause server outages and damages to its components. In several studies that have been done, the application of fault tolerant system is rarely used where the system only relies on a single sensor to obtain temperature data so it is susceptible to fault. Therefore, in this study, the self-purging redundancy method is used to determine the effect of the method on system's fault handling. In this research, 3 redundant modules are used in the form of Arduino Nano and temperature sensor DHT-22 to obtain temperature data which is then sent to a voter in the form of Arduino Uno. From this study, it is found that self-purging redundancy method has an effect on system's fault handling, where the system can tolerate error that occur in one redundant module at one time. In addition, the system has an accuracy of 85% and an average computation time of 14,8 milliseconds.

Keywords: *server, fault tolerant system, self-purging redundancy, temperature monitoring system, DHT-22*

1. PENDAHULUAN

Server merupakan sistem komputer yang berfungsi melayani dan mengontrol akses *client* yang terhubung dengannya. *Server* pada umumnya ditempatkan dalam ruangan khusus dengan sistem pendingin untuk menjaga suhu

dari komponen didalamnya. Hal ini dilakukan karena komponen pada *server* menghasilkan panas ketika bekerja. Menurut Pradana & Nurfiana (2019) nilai suhu yang ideal untuk *server* bekerja terdapat pada rentang 20 - 25°C. Jika suhu pada ruang *server* tidak dijaga dapat menimbulkan berbagai permasalahan seperti

penurunan kinerja *server*, kerusakan pada komponen *server*, dan hilangnya data pada *server*. Salah satu contoh dampak dari tidak dikontrolnya suhu pada ruang *server* terjadi pada tahun 2019. Dimana *data center* AWS di Tokyo, Jepang mengalami *overheating* sehingga mengganggu layanan *cloud* dari Amazon (Palmer, 2019). Oleh karena itu, diperlukan sistem monitoring suhu pada ruang *server* yang memiliki fitur *fault tolerant*.

Pada penelitian yang telah dilakukan sebelumnya, pengaplikasian sistem *fault tolerant* masih belum banyak dilakukan. Pada penelitian yang dilakukan oleh Pradana & Nurfiyana (2019) dibangun sebuah sistem monitoring suhu pada ruang *server* yang dapat diakses dari jarak jauh menggunakan perangkat *mobile*. Pada penelitian yang lain oleh Raharjo, Marwanto, & Romadhona (2019) dibangun sebuah sistem monitoring suhu dan kelembaban pada ruang *server* berbasis *Internet of Things* (IoT). Akan tetapi, pada penelitian-penelitian tersebut belum menerapkan *fault tolerant system* dimana data suhu yang didapatkan berasal dari sebuah sensor suhu saja dan tidak terdapat mekanisme apabila sensor suhu tersebut mengalami kesalahan atau eror.

Self-purging redundancy merupakan salah satu bentuk penerapan dari *fault tolerant system*. Pada penelitian yang dilakukan oleh Fadlan, Primananda, & Kurniawan (2020) metode *self-purging redundancy* mampu meningkatkan reliabilitas sistem sebesar 9,2%. Pada penelitian ini, penerapan metode *self-purging redundancy* diharapkan dapat meningkatkan reliabilitas dari sistem monitoring suhu ruang *server*, dimana sistem mampu menoleransi kesalahan yang mungkin terjadi pada pembacaan data sensor suhu.

2. PERANCANGAN DAN IMPLEMENTASI

2.1 Gambaran Umum Sistem

Sistem menerapkan metode *self-purging redundancy* menggunakan 3 buah modul redundan yang identik. Data dari modul redundan lalu dikirimkan ke *voter* untuk dilakukan proses *voting*. Hasil dari *voting* selanjutnya dikembalikan ke modul redundan untuk mendeteksi kesalahan. Proses mendeteksi kesalahan pada modul redundan dilakukan dengan melakukan perbandingan pada data hasil *voting* dengan data yang dikirimkan oleh modul

redundan.

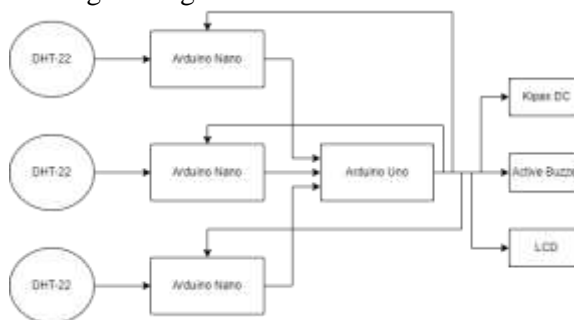
Sistem yang menerapkan metode *self-purging redundancy* dengan 3 buah modul redundan dapat menoleransi kesalahan pada sebuah modul redundan. Hal tersebut sesuai dengan rumus toleransi kesalahan yang terdapat pada Persamaan 1.

$$x = n - 2 \tag{1}$$

Dimana x adalah jumlah kesalahan yang dapat ditoleransi dan n adalah jumlah modul redundan pada sistem.

2.2. Perancangan Perangkat Keras

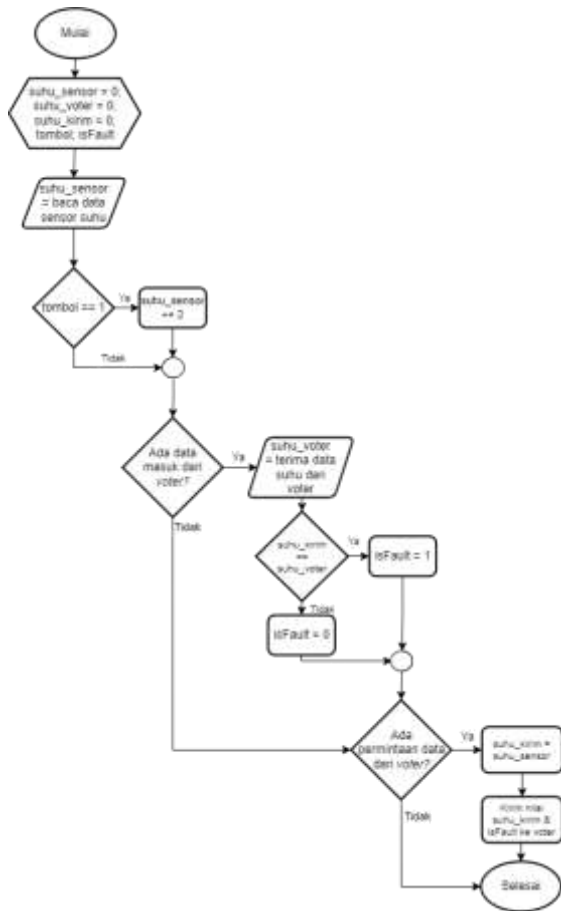
Perancangan perangkat keras dilakukan berdasarkan blok diagram sistem pada Gambar 1. Modul redundan pada sistem terdiri dari sensor suhu DHT-22 dan Arduino Nano. Sedangkan Arduino Uno berfungsi sebagai *voter* pada sistem. Kipas DC, *active buzzer*, dan LCD berfungsi sebagai keluaran dari sistem.



Gambar 1. Blok diagram sistem

2.3. Perancangan Perangkat Lunak

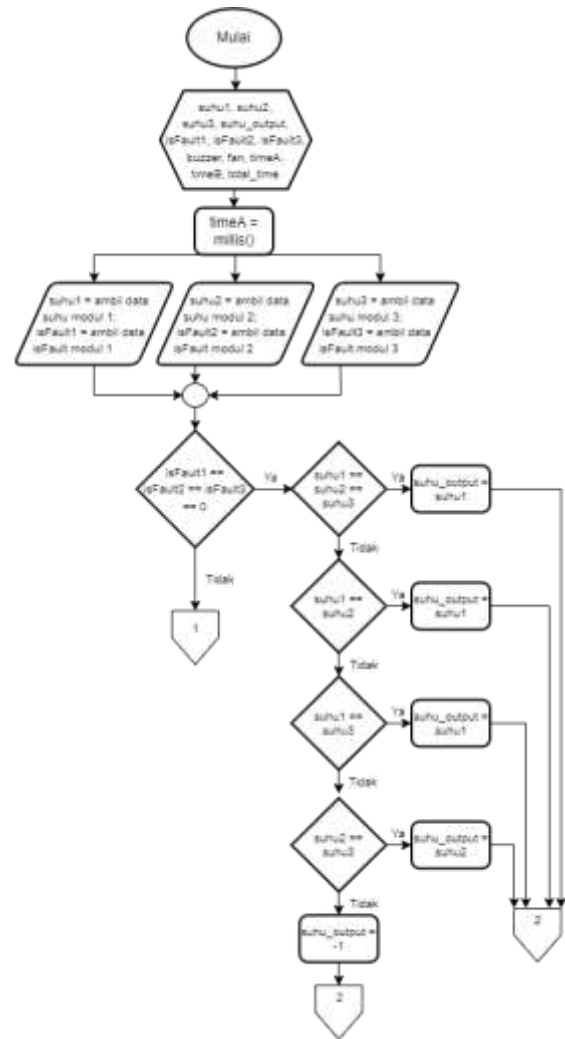
Terdapat 2 program yang dirancang pada penelitian ini, yaitu program *voter* dan program modul redundan. Program *voter* memiliki fungsi utama melakukan *voting* serta mengatur keluaran dari sistem. Sedangkan program modul redundan memiliki fungsi utama untuk mendeteksi kesalahan. Diagram alir dari program modul redundan dan program *voter* masing-masing terdapat pada Gambar 2, Gambar 3, dan Gambar 4.



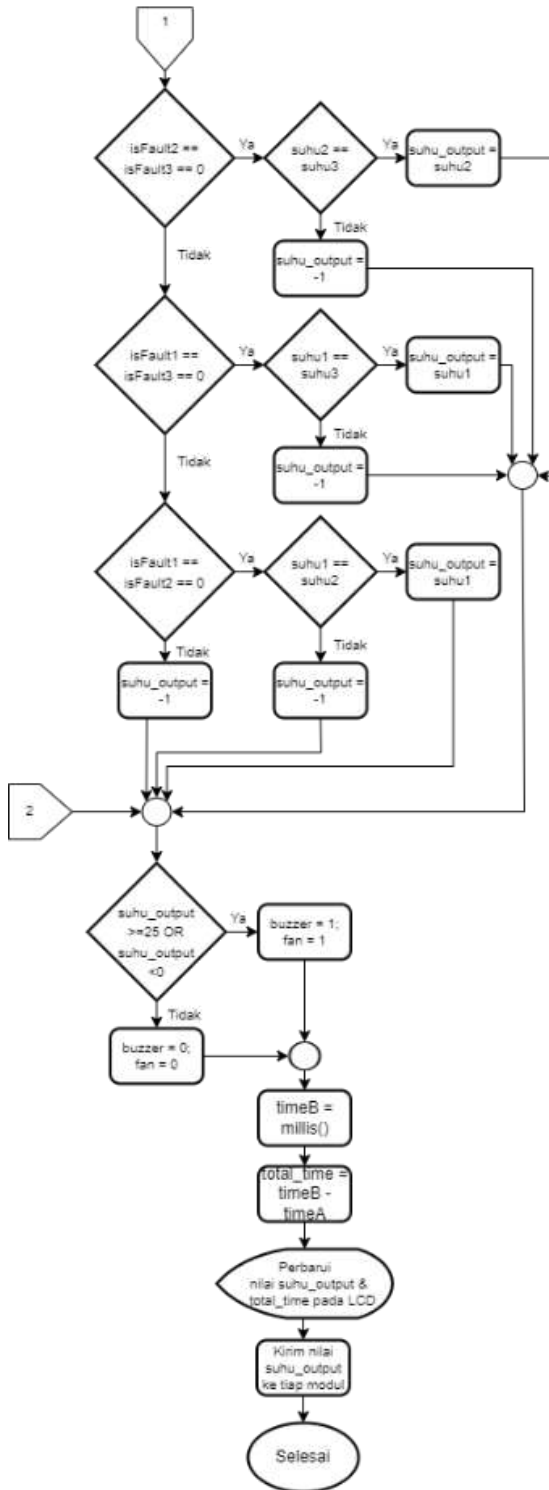
Gambar 2. Diagram alir program modul redundan

Pada Gambar 2 terdapat seleksi kondisi untuk mengetahui kondisi dari tombol yang terdapat pada masing-masing modul redundan. Tombol ini digunakan untuk pengujian *fault tolerant*, dimana jika tombol tersebut aktif maka suhu bacaan sensor diubah beberapa derajat sehingga modul redundan seolah mengalami kesalahan.

Pada diagram alir program *voter* terdapat 2 kali seleksi kondisi untuk menentukan suhu keluaran sistem. Seleksi kondisi yang pertama berfungsi untuk mengetahui apakah terdapat modul redundan yang mengalami kesalahan. Sedangkan seleksi kondisi yang kedua berfungsi untuk melakukan *voting* suhu keluaran sistem. Jika pada seleksi kondisi pertama terdapat modul redundan yang mengalami kesalahan, maka nilai suhu dari modul redundan tersebut tidak akan digunakan pada proses *voting* seperti pada Gambar 4. Pada diagram alir program *voter* juga terdapat fungsi *millis()* yang digunakan untuk menghitung waktu komputasi dari sistem.



Gambar 3. Diagram alir program voter



Gambar 4. Diagram alir program voter (lanjutan)

2.4. Implementasi Sistem

Dari perancangan sistem yang telah dilakukan, dibuat bentuk implementasi dari sistem yang terdapat pada Gambar 5 berikut.



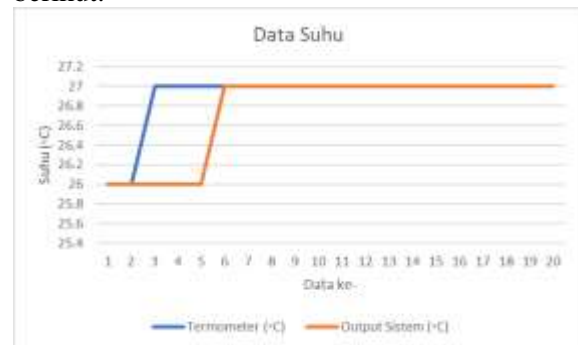
Gambar 5. Implementasi sistem

Program yang telah dirancang diimplementasikan pada masing-masing mikrokontroler menggunakan Arduino IDE. Data yang ditampilkan pada LCD adalah data suhu hasil voting dan waktu komputasi dari sistem.

3. PENGUJIAN

3.1. Pengujian Nilai Akurasi

Pengujian nilai akurasi dilakukan untuk mengetahui ketepatan nilai suhu sistem. Pengujian dilakukan dengan mengambil data suhu termometer ruangan dan data suhu keluaran sistem setiap menit selama 20 menit. Hasil pengujian nilai akurasi terdapat pada Gambar 6 berikut.



Gambar 6. Hasil uji nilai akurasi

Dari hasil pengujian terdapat 3 data salah, yaitu data ke 3, 4, dan 5. Dimana pada ketiga data tersebut terdapat perbedaan antara nilai suhu pada termometer ruangan dengan nilai suhu keluaran sistem. Sehingga nilai akurasi dari sistem dapat dihitung menggunakan Persamaan 2 berikut.

$$Akurasi (\%) = \frac{data\ tepat}{total\ data} \times 100\% \quad (2)$$

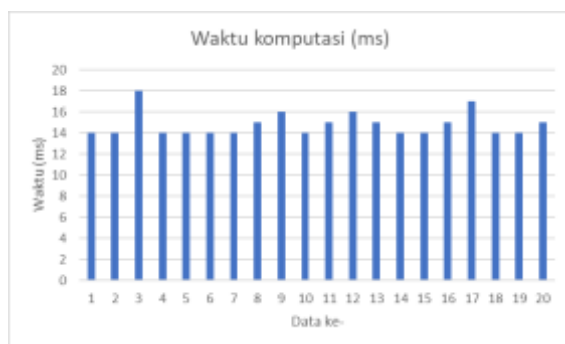
$$Akurasi (\%) = \frac{17}{20} \times 100\% = 85\%$$

Dimana data tepat adalah data yang memiliki nilai suhu sama pada termometer ruangan dan keluaran sistem. Sedangkan total data adalah jumlah seluruh pengambilan data yang dilakukan. Dari Persamaan 2 tersebut didapatkan

nilai akurasi dari sistem adalah 85%.

3.2. Pengujian Waktu Komputasi

Pengujian waktu komputasi dilakukan untuk mengetahui rata-rata waktu komputasi yang dibutuhkan oleh sistem dalam menjalankan fungsinya. Waktu komputasi dihitung dari sistem meminta data suhu hingga sistem menampilkan suhu keluaran pada LCD. Pengujian dilakukan dengan mengambil data waktu komputasi sistem setiap menit selama 20 menit. Hasil pengujian waktu komputasi digambarkan pada Gambar 7 berikut.



Gambar 7. Hasil uji waktu komputasi

Dari hasil pengujian pada Gambar 7 dapat dihitung rata-rata waktu komputasi sistem menggunakan Persamaan 3 berikut.

$$\bar{X} = \frac{\sum x}{n} = \frac{296}{20} = 14,8 \quad (3)$$

Dimana \bar{X} adalah rata-rata waktu komputasi, $\sum x$ adalah jumlah seluruh waktu komputasi yang didapat, dan n adalah jumlah seluruh data. Dari Persamaan 3 tersebut didapatkan rata-rata waktu komputasi sistem sebesar 14,8 milidetik.

3.3. Pengujian Fault Tolerant

Pengujian *fault tolerant* dilakukan untuk mengetahui pengaruh dari metode *self-purging redundancy* terhadap penanganan kesalahan pada sistem. Pengujian dilakukan dengan melihat kondisi sistem saat terdapat kombinasi modul yang mengalami kesalahan. Tombol pada setiap modul redundan digunakan untuk mensimulasikan kesalahan, dimana ketika tombol ditekan maka nilai suhu bacaan sensor pada modul tersebut diubah beberapa derajat. Hasil pengujian *fault tolerant* terdapat pada Tabel 1 berikut.

Tabel 1. Hasil uji *fault tolerant*

Tombol modul 1	Tombol modul 2	Tombol modul 3	Kondisi sistem
Mati	Mati	Mati	Normal
Hidup	Mati	Mati	Normal
Mati	Hidup	Mati	Normal
Mati	Mati	Hidup	Normal
Hidup	Hidup	Mati	Gagal
Hidup	Mati	Hidup	Gagal
Mati	Hidup	Hidup	Gagal
Hidup	Hidup	Hidup	Gagal

Kolom kondisi sistem bernilai normal jika sistem dapat menampilkan nilai suhu keluaran sistem dengan benar. Sedangkan bila sistem gagal menampilkan nilai suhu keluaran sistem dengan benar kolom maka kolom kondisi sistem bernilai gagal. Dari Tabel 1 tersebut didapatkan bahwa sistem mulai mengalami kegagalan ketika terdapat lebih dari 1 modul redundan yang mengalami kesalahan. Sehingga pengaruh metode *self-purging redundancy* pada sistem adalah metode tersebut membuat sistem mampu mendeteksi adanya kesalahan yang terjadi serta membuat sistem tetap dapat menampilkan suhu keluaran sistem dengan benar ketika terdapat sebuah modul redundan yang mengalami kesalahan.

4. PENUTUP

4.1. Kesimpulan

Berdasarkan pengujian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Sistem monitoring suhu ruang *server* memiliki nilai akurasi 85% setelah diterapkannya metode *self-purging redundancy*.
2. Rata-rata waktu komputasi yang dibutuhkan oleh sistem monitoring suhu ruang *server* sebesar 14,8 milidetik.
3. Pengaruh metode *self-purging redundancy* terhadap penanganan kesalahan pada sistem monitoring suhu ruang *server* adalah sistem dapat mendeteksi kesalahan yang muncul serta menoleransi kesalahan pada sebuah modul redundan dalam satu waktu.

4.2. Saran

Dari penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diterapkan pada pengembangan lebih lanjut dari penelitian ini. Pertama, penggunaan sensor suhu yang lebih akurat diharapkan dapat meningkatkan akurasi dari sistem. Selanjutnya, penambahan modul

redundan dapat dilakukan agar sistem dapat menoleransi lebih banyak kesalahan pada waktu yang sama. Terakhir, pengujian waktu komputasi dengan lebih banyak modul redundan dapat dilakukan untuk mengetahui jumlah modul redundan yang efisien untuk diterapkan pada sistem monitoring suhu ruang *server*.

5. DAFTAR PUSTAKA

- Dubrova, E., 2013. *Fault-Tolerant Design*. 1st ed. New York: Springer.
- Fadlan, I., Primananda, R. & Kurniawan, W., 2020. Implementasi Fault Tolerant System menggunakan metode Self-Purging Redundancy Pada Sistem Pendeteksi Kebakaran. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(6), pp. 1601-1608.
- Losq, J., 1976. A Highly Efficient Redundancy Scheme: Self-Purging Redundancy. *IEEE Transactions On Computers*, C-25(6), pp. 569-578.
- Nørveg, K., 2000. *An Introduction to Fault-Tolerant Systems*, Trondheim: Department of Computer and Information Science Norwegian University of Science and Technology.
- Palmer, D., 2019. *Amazon Cloud Outage Caused Major Issues at Some Crypto Exchanges*. [Online] Tersedia di: <<https://www.coindesk.com/markets/2020/08/23/amazon-cloud-outage-caused-major-issues-at-some-crypto-exchanges/>> [Diakses 4 September 2021].
- Pradana, A. & Nurfiana, 2019. Rancang Bangun Monitor dan Kontrol Suhu Ruang Server Menggunakan Perangkat Mobile Berbasis Internet Of Things (IOT). Dalam: *Seminar Nasional Riset Terapan 2019*. Banjarmasin, Indonesia, 7 November 2019. Banjarmasin: Politeknik Negeri Banjarmasin.
- Raharjo, E. B., Marwanto, S. & Romadhona, A., 2019. Rancangan Sistem Monitoring Suhu dan Kelembapan Ruang Server Berbasis Internet of Things. *Jurnal Teknik Atw*, 6(2), pp. 61-68.