Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 2, No. 11, November 2018, hlm. 5871-5876 http://j-ptiik.ub.ac.id

Implementasi *Hardware Redundancy* Pada Sistem Akuisisi Data Sensor Dengan Menggunakan Metode *Hot Standby Sparing*

Arie Prayogo Pangestu[1], Sabriansyah Rizqika Akbar[2], Primantara Hari Trisnawan[3]

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email : [1]ariepangestu.app@gmail.com, [2]sabrian@ub.ac.id, [3]prima@ub.ac.id

Abstrak

Saat ini banyak muncul produk untuk memonitoring suatu objek dari berbagai vendor penyedia sistem layanan yang digunakan. Produk yang dimaksud yaitu sensor yang mampu melakukan *sensing* pada suatu kondisi objek, salah satunya digunakan untuk monitoring kondisi air di tambak. Ketika air pada objek tambak mulai keruh, kotor, suhu air meningkat dan kadar garam tidak teratur maka akan mempengaruhi kualitas dari hasil tambak tersebut. Terkadang beberapa sensor pada sistem monitoring dapat mengalami kerusakan, bisa disebabkan oleh faktor dari dalam maupun dari luar. Oleh karena itu dibutuhkan sebuah sistem yang mampu meredundansi sistem utama apabila terjadi kesalahan. Pada penelitian ini terdapat tiga sensor yaitu sensor suhu, kekeruhan dan sensor kadar garam pada setiap modul *master* dan *slave* serta menggunakan metode *Hot Standdby Sparing.* Metode ini merupakan cara penyelesaian agar modul yang akan dijadikan sebagai *slave modul* tetap hidup atau dalam kondisi *idle* walaupun masuk kedalam kondisi *sleep* dan siap menggantikan modul master ketika terdapat kesalahan sistem. Dari hasil pengujian, modul *master* dan modul *slave* dapat berfungsi dengan baik. Modul *slave* mampu meredundansi modul *master* ketika terdapat kesalahan pada sensor serta adanya gangguan daya pada modul *master*, dan sistem dapat menghindari kondisi *race* antara modul *master* dan *slave* ketika pertama kali dihidupkan.

Kata kunci : *Redundancy, Fault Tolerance, Standby Redundancy, Hot Standby Sparing*

Abstract

Currently there are many products to monitor an object from various vendor service system providers that are used. The intended product is a group of sensors with interconnected microcontrollers to provide information of what the sensors capture an object condition, one of which is used for monitoring water conditions. When the water in the ponds is turbid, dirty, the water temperature increased and the salt content is irregular it will affect the quality of the fishpond. Sometimes some sensors on the monitoring system can be damaged, due to factors from within and from outside. Therefore, a system that is able to redundancy the main system in case of error. In this study there are three sensors namely temperature sensor, turbidity and salinity sensors in each master module and slave and using Hot Standdby Sparing method. This method is a way of completion so that the module that will be used as slave module remain alive or in idle condition even though it goes into sleep condition and ready to replace master module when there is system error. From the test results, the master module and slave module can work properly. The slave module can redundate the master module when there is an error on the sensor as well as the power failure of the master module, and the system can avoid race conditions between master and slave modules when first turned on.

Keywords *: Redundancy, Fault Tolerance, Standby Redundancy, Hot Standby Sparing.*

1. PENDAHULUAN

Monitoring saat ini menjadi salah satu kebutuhan bagi kalangan tertentu yang membutuhkan informasi mengenai data suatu objek. Salah satu objek yang sering dimonitoring ialah kondisi air. Kondisi air dapat dimonitoring dengan menggunakan bantuan sensor seperti sensor suhu, kekeruhan dan kadar garam. Terkadang terjadi kesalahan pada sensor baik karena faktor kesalahan dari luar maupun dari dalam sistem. Dampak dari kesalahan sistem sensor tersebut dapat mengakibatkan

kerugian material. Hal ini diperkuat dengan adanya data tentang kerugian tambak udang dengan luas 4 hektar dapat mengakibatkan kerugian hingga puluhan juta rupiah. (Atiek, 2017)

Dalam hal ini dapat diimplementasikan beberapa metode yang mampu mentolerir adanya kesalahan pada sistem sensor. Diantaranya *software* redundansi, *hardware* redundansi, *information* redundansi, dan *time* redundansi. Perdedaan dari empat metode tersebut yaitu terletak pada inti permasalahan yang mengakibatkan adanya kesalahan pada sistem tersebut. Kesalahan dapat terjadi pada sisi *software*, *hardware*, information, maupun *time.*

Berdasarkan permasalahan yang telah dipaparkan sebelumnya, maka penulis memilih menggunakan *hardware* redundansi sebagai acuan dalam menyelesaikan kesalahan pada sistem sensor. Dalam *hardware* redundansi terdapat beberapa metode penyelesaian antara lain *hot standby spare*, dan *cold standby spare*. Perdedaan dari kedua metode tersebut yaitu terletak pada kondisi sistem cadangan. Pada metode *hot standby spare*, sistem cadangan berada pada kondisi sleep. Sedangkan pada metode *cold standby spare*, sistem cadangan berada pada kondisi mati.

Oleh karena itu, penulis memilih menggunakan *hardware* redundansi dengan metode *hot standby spare* dalam menyelesaikan permasalahan pada sistem akusisi data sensor. Dengan menerapkan mekanisme *hardware* redundansi pada sistem akuisisi data sensor ini, diharapkan mampu meningkatkan *availability* sistem saat mengalami kegagalan dari sisi sensor. Karena nilai *availability* sangat berguna bagi sensor yang ada pada sistem monitoring.

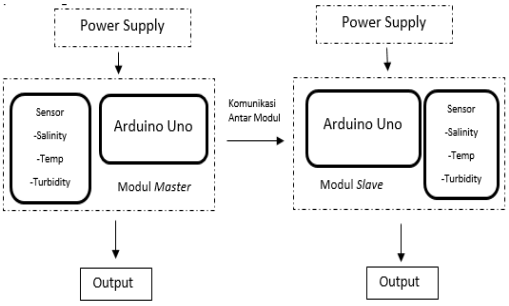
2. METODOLOGI

Penelitian ini merupakan suatu jenis penelitian implementatif pada konteks penelitian pengembangan lanjut yang merupakan pengembangan terhadap sistem yang sudah pernah diteliti sebelumnya dengan menambah beberapa fitur baru. Diagram alur penelitian dapat dilihat pada Gambar 1.berikut ini :

Gambar 1. Diagram Alur Penelitian

3.1 Perancangan dan Implementasi

Pada tahap ini menerapkan semua tahapan dari studi literatur sampai pada tahap analisis kebutuhan dalam melakukan perancangan pengembangan penelitian *"Implementasi Hardware Redundancy Pada Sistem Akuisisi Data Sensor Dengan Menggunakan Metode Hot Standby Sparing"*. Berikut adalah diagram blok perancangan sistem.



Gambar 2. Diagram Blok Perancangan Sistem

Berdasarkan Gambar 4, sistem akan dirancang dengan menggunakan dua mikrokontroler yang dikomunikasikan secara serial. Komunikasi serial ini dilakukan secara *wired*. Pada kedua arduino akan dirancang sistem yang sama persis dapat menangkap hasil dari pembacaan sensor salinitas, suhu dan kekeruhan. Sehingga sistem akan dirancang dengan beberapa proses yaitu :

1. Modul *Master*.

Pada proses ini akan didapatkan data dari sensor yang kemudian akan dikirimkan kepada

user melalui serial monitor yang ada pada fitur aplikasi Arduino IDE.

2. Modul *Slave*.

Pada proses ini, proses yang dilakukan sama dengan modul *master,* tetapi pada proses ini modul *slave* akan menunggu triger dari modul *master*, apabila triger bernilai 0 atau diasumsikan terjadi *fault* pada modul *master* maka modul *slave* akan mengambil alih pekerjaan untuk mengirimkan hasil dari pembacaan sensor kepada *user*.

3. Komunikasi

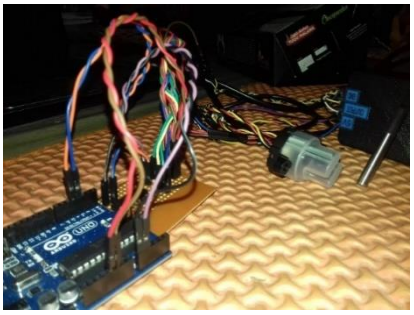
Pada sisi komunikasi antar modul, setiap modul berkomunikasi menggunakan *wire* sebagai media *interupt* yang dapat mengaktifkan modul *slave* ketika terdapat kesalahan pada modul *master*.

4. IMPLEMENTASI

Pada implementasi sistem dilakukan saat semua proses pada tahap perancangan sistem telah selesai atau telah dipenuhi. Pembahasan tentang implementasi pada sub bab ini akan membahas mengenai implementasi *hardware* redundansi pada sistem akuisisi data sensor dengan menggunakan metode *hot standby sparing* baik dari implementasi modul *master* dan *slave*, implementasi *interrupt* pada modul *master* serta implementasi komunikasi antara modul *master* dan modul *slave*.

4.1 Implementasi Modul *Master*

Pada implementasi modul *master*,sensor akan dihubungkan ke IC 7408 dan ke pin input pada Arduino. Tujuan dari dihubungkannya data sensor ke IC 7408 ialah untuk deteksi kesalahan yang terjadi pada sensor, apabila ada salah satu atau beberapa sensor mengalami *fault* maka output yang dikeluarkan IC 7408 ialah bernilai *low*, nilai ini menjadi input dari pin *interrupt* dimana ketika *input* yang diterima bernilai *low* maka *interrupt* akan membuat Arduino masuk dalam keadaan *sleep* dan sebaliknya. Sedangkan tujuan dari dihubungkannya data sensor ke pin *input* data sensor ialah agar hasil data *input* sensor dapat diamati melalui serial monitor yang ada pada fitur aplikasi Arduino IDE. Hasil implementasi modul *master* dapat dilihat pada Gambar 3.



Gambar 3. Implementasi Modul *Master*

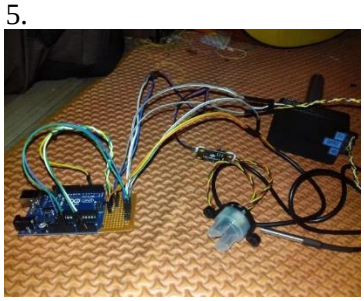
Untuk kode pemrograman yang diupload pada modul *master*, penulis menjabarkan penggalan program pada aplikasi Arduino IDE yang menjelaskan tentang fungsi-fungsi pin pada Arduino UNO yang digunakan sebagai modul *master* seperti pada Gambar 4.

```
void setup() {
  Serial.begin(9600);
  pinMode(4,INPUT);
  digitalWrite(4,HIGH);
  pinMode(A0,INPUT);
  analogWrite(A0,HIGH);
  pinMode(A1,INPUT);
  analogWrite(A1,HIGH);
  pinMode(3,INPUT);
  digitalWrite(3,HIGH);
  pinMode(5,OUTPUT);
  digitalWrite(5,HIGH);
}
```

Gambar 4. Potongan Program Fungsi Modul *Master*

4.2 Implementasi Modul *Slave*

Pada implementasi modul *slave*, sensor tidak dihubungkan ke IC 7408, data dari sensor langsung masuk ke pin input data sensor. Tujuannya tidak digunakan IC 7408 pada modul *slave* ialah agar input pin interrupt yang diterima modul *slave* nantinya terfokus hanya dari output modul *master*. Dalam kondisi seperti ini masalah dibatasi yaitu seluruh sensor pada modul *slave* dianggap aktif dan tak mengalami kesalahan sedikitpun, sehingga ketika melakukan pengujian sistem dapat diamati bagaimana sistem *slave* dapat meredundansi sistem *master*. Berikut hasil dari implementasi modul *slave* dapat dilihat pada Gambar 5.



Gambar 5. Implementasi Modul *Slave*

Untuk kode pemrograman yang diupload pada modul *slave*, penulis menjabarkan penggalan program pada aplikasi Arduino IDE yang menjelaskan tentang fungsi-fungsi pin pada Arduino UNO yang digunakan sebagai modul *slave* seperti pada Gambar 6.

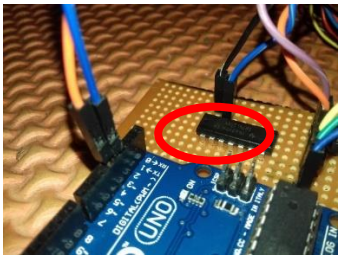
```
void setup() {
  Serial.begin(9600);
  pinMode(4,INPUT);
  digitalWrite(4,HIGH);
  pinMode(A0,INPUT);
  analogWrite(A0,HIGH);
  pinMode(A1,INPUT);
  analogWrite(A1,HIGH);
  pinMode(3,INPUT);
  digitalWrite(3,LOW);
}
```

Gambar 6. Potongan Program Fungsi Modul *Slave*

4.3 Implementasi *Interrupt* pada Modul *Master*

Pada implementasi komunikasi antara sensor dengan modul *master*, seperti sudah dijelaskan sebelumnya pada pembahasan tentang implementasi modul *master* bahwa output dari sensor akan masuk langsung ke *pin input* sensor dan juga masuk ke IC 7408. IC7408 merupakan sebuah IC gerbang logika *and*, digunakan gerbang logika *and* karena logika ini hanya akan menghasilkan *output high* jika dan hanya jika semua *input* dari *output* sensor yang masuk ke IC 7408 bernilai *high*, salah satu saja input bernilai *low*, maka output yang dikeluarkan IC 7408 akan bernilai *low* yang berarti akan mengaktifkan *pin interrupt* dan membuat modul *master* masuk dalam kondisi *sleep*.

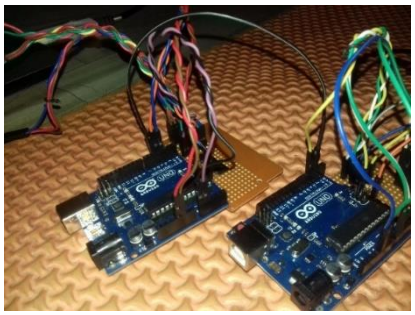
Sedangkan data sensor yang masuk langsung ke pin *input* sensor akan ditampilkan pada serial monitor yang ada di aplikasi Arduino IDE. Berikut hasil dari implementasi *interrupt* pada modul *master* seperti pada Gambar 7.



Gambar 7. Implementasi *Interrupt* Modul *Master*

4.4 Implementasi komunikasi Antara Modul Master dan Slave

Pada implementasi komunikasi antara modul *master* dan *slave*, komunikasi ini bertujuan untuk mengiplementasikan bagaimana *hardware* redundansi dapat bekerja pada sistem akuisisi data sensor khususnya pada metode *hot standby spare*. Terdapat *pin output* yaitu pin 5 pada modul *master* yang digunakan sebagai *input* dari *pin interrupt* yang ada pada modul *slave*, ketika *pin output* modul *master* bernilai *low* karena terdapat *fault* pada salah satu atau beberapa sensor pada modul *master*, maka *pin interrupt* yang ada pada modul *slave* akan aktif dan membangunkan modul *slave* yang sebelumnya ada dalam kondisi *sleep*, begitupun sebaliknya. Berikut implementasi dari komunikasi antara modul *master* dan *slave* dapat dilihat pada Gambar 8.



Gambar 8. Implementasi Komunikasi Modul *Master dan* Slave

5. PENGUJIAN DAN ANALISIS

Pengujian dan analisis akan dibagi menjadi empat bagian, yaitu fungsi modul *master*, fungsi modul *slave*, pengujian menghindari kondisi *race* dan pengujian redundnsi sistem.

5.1 Pengujian Fungsi Modul *Master*

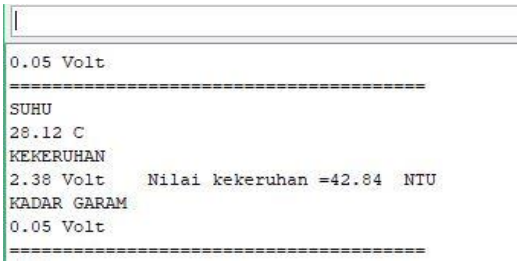
Tujuan dari pengujian fungsi modul *master* adalah untuk memastikan bahwa modul *master* dapat berfungsi sesuai dengan tujuan dibuatnya modul *master* yaitu dapat memonitoring kondisi air. Langkah dalam melakukan pengujian yang pertama Menghubungkan modul *master* dengan Laptop kemudian meng-*upload* program yang sudah dibuat pada aplikasi Arduino IDE dan terakhir buka serial monitor untuk mengamati hasil dari *sensing* sensor-sensor yang ada pada modul *master*.

```
0.05 Volt
=======================================
SUHU
27.06 C
KEKERUHAN
2.40 Volt    Nilai kekeruhan =42.37  NTU
KADAR GARAM
0.05 Volt
=======================================
```

Gambar 9. Hasil *Sensing* Sensor Modul *Master*

5.2 Pengujian Fungsi Modul *Slave*

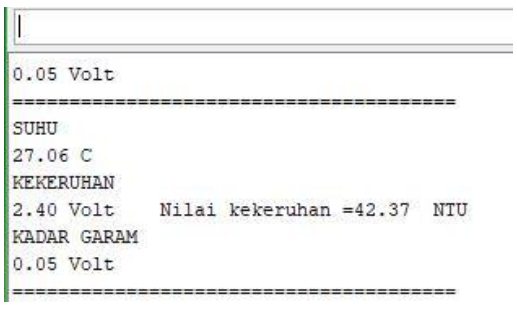
Tujuan dari pengujian fungsi modul *slave* adalah untuk membuktikan bahwa modul *slave* dapat berfungsi dengan baik dalam memonitoring kondisi air saat modul *master* mengalami masalah. Langkah dalam melakukan pengujian yang pertama dengan menghubungkan modul *slave* dan Laptop kemudian meng-*upload* program yang sudah dibuat pada aplikasi Arduino IDE dan terakhir buka serial monitor untuk mengamati hasil dari *sensing* sensor-sensor yang ada pada modul *slave*.



Gambar 10. Hasil *Sensing* Sensor Modul *Slave*

5.3 Pengujian Menghindari Kondisi *Race*

Tujuan dari pengujian menghindari kondisi *race* adalah untuk memastikan bahwa ketika sistem dinyalakan pertama kali baik modul *master* maupun modul *slave* secara bersamaan tidak akan terjadi kondisi *race* atau kedua modul tidak ada yang mengalah dan dalam kondisi aktif keduanya. Langkah pengujian hampir sama dengan pengujian fungsi modul *master* dan *slave*, hanya pada pengujian ini, kedua modul dinyalakan secara bersamaan untuk melihat apakah kondisi *race* dapat diatasi pada sistem.



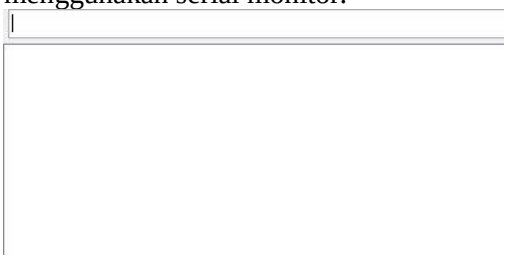
Gambar 11. Serial Monitor Modul *Master*

Gambar 12. Serial Monitor Modul *Slave*

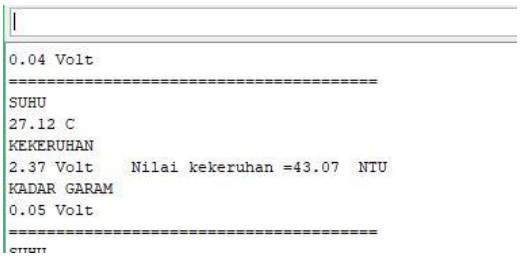
Dari hasil pengujian menghindari kondisi *race* dapat dilihat dari Gambar 13 dan Gambar 14 bahwa saat kedua modul tersebut dinyalakan bersama-sama, modul *master* akan langsung aktif dan menampilkan data dari hasil tangkapan data ketiga sensor sedangkan modul *slave* akan langsung masuk dalam kondisi *sleep* saat pertama kali dinyalakan. Hasil tersebut membuktikan bahwa sistem dapat menghindari kondisi *race* pada saat sistem dinyalakan bersamaan.

5.4 Pengujian Redundansi Sistem

Tujuan dari pengujian redundansi sistem ini adalah untuk memastikan bahwa modul *slave* mampu berjalan dengan baik untuk melakukan redundansi terhadap modul *master* ketika terdapat kesalahan dari salah satu atau beberapa sensor yang ada pada modul *master*. Langkah dalam melakukan pengujian yang pertama dengan menghubungkan kedua modul ke Laptop, kemudia buat kesalahan pada sensor-sensor yang ada pada modul *master* kemudian mengamati apakah modul *slave* mampu meredundansi modul *master* dengan menggunakan serial monitor.



Gambar 13. Serial Monitor Modul *Master Sleep*



Gambar 14. Serial Monitor Modul *Slave* Aktif

Dari hasil pengujian pada Gambar 15 dan Gambar 16 dapat dilihat bahwa ketika modul *master* mengalami masalah dari tiap sensor, maka modul *master* akan masuk dalam kondisi *sleep* dan modul *slave* akan terbangun karena menerima sinyal *low* dari modul *master*. Hasil pengujian tersebut membuktikan bahwa modul *slave* mampu dengan baik meredundansi modul *master* secara otomatis sehingga kesalahan pada modul *master* dapat diatasi dengan baik serta mampu meningkatkan ketersediaan dari sistem akuisisi data sensor yang dalam hal ini digunakan untuk memonitoring kondisi air.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan rumusan masalah yang diangkat di awal dan berdasarkan hasil dari pengujian dan implementasi yang telah dilakukan maka dapat ditarik beberapa poin kesimpulan, sebagai berikut :

Sistem ini dapat mendeteksi adanya kegagalan pada sistem dengan memanfaatkan komponen IC 7408 yang digunakan untuk memproses output sensor. Ketika terdapat kesalahan pada salah satu atau beberapa sensor yang ada di modul *master*,sistem dapat terus berjalan karena terdapat modul yang digunakan sebagai *slave*.Komunikasi modul *master* dan *slave* dilakukan dengan menggunakan media *wire* atau kabel.Modul *slave* akan menerima sinyal dari *output* modul *master*,ketika modul *master* mengalami kesalahan sistem maka modul akan masuk kedalam kondisi *sleep*, ini berarti modul *master* akan mengirimkan sinyal *low* kepada modul *slave* dan akan membangunkan modul *slave* yang pada awalnya berada dalam kondisi *sleep*.

6.2 Saran

Beberapa saran dari penulis yang dapat diberikan sebagai bahan untuk memperbaiki dan mengembangkan sistem adalah sebagai berikut:

1. Memperhatikan waktu peralihan dari modul *master* ke modul *slave* agar tidak terlalu lama,sehingga dapat meningkatkan *availability* sistem.
2. Media komunikasi antara modul *master* dan *slave* kedepannya menggunakan media nirkabel sehingga dapat mengurangi penggunaan kabel yang berlebih dan agar sistem terlihat lebih rapi.
3. Menggunakan *board* mikrokontroler selain Arduino UNO untuk dapat dibandingkan performanya baik dari segi ketepatan,kecepatan dan efisiensi dari penerapan sistem.

DAFTAR PUSTAKA

Abidin, Z., dkk. (2014). *Energy efficient communication protocol for wireless microsensor networks*. In: Proc 33rd Hawaii International Conference on Sistem Sciences (pp. 1-10).

Agustiningsih, E., 2016. *Perancangan Perangkat Monitoring Kualitas Air Pada Kolam Budidaya Berbasis Web Localhost.* Universitas Maritim Raja Ali Haji ,Tanjungpinang.
Diakses 08 September 2017.

Dubrova, E., 2012. *Fault-Tolerant Design*. s.l.:s.n.

https://www.arduino.cc/en/Guide/Introduction #, Diakses 20 Januari 218

Losq, J., 1975. *Influence of fault-detection and switching mechanisms on the reliability of standby systems*. In: Digest 5th International Symposium on Fault-Tolerant Computing, (pp. 81–86).

National Instrument. (2008). *Redundant System BasicConsepts*.http://www.ni.com/white-paper/6874/en/. Diakses 16 Januari 2018.

Sambora, Y., 2016. *Monitoring Kualitas Air Pada Budidaya Udang Berbasis ATMEGA328 yang Terkonfigurasi Bluetooth HC-05*. UNY,Yogyakarta.