

PENGUJIAN SISTEM PENGENDALI PERANGKAT ELEKTRONIK MENGGUNAKAN ARDUINO UNO DAN RANGKAIAN PENGKONDISI SINYAL

David¹, A. Murtado²

^{1,2}Program Studi Teknik Informatika STMIK Pontianak

^{1,2}STMIK Pontianak, Jln. Merdeka No. 372 Pontianak, Kalimantan Barat

david_liaw@yahoo.com dan david_liaw@stmikpontianak.ac.id

²a.murtado.resha@yahoo.com

ABSTRACT

System Control is a system designed and built to control object or objects in a specific environment. This research, consist of control system design on electronic appliances that are controlled via desktop application. The application designed in this research enable user to control every electronic appliance connected to the system. The designed application also enables the user to obtain information of the environment through sensors that are attached within the system. The methods used in this research are experimental methods, study on relevant literatures, and documentation. Design process was divided into two parts namely, software design and hardware design, both using Object Oriented Analysis and Design Method. Hardware used for the control system are signal conditioning circuit and Arduino UNO as a component to receive and to transmit data from sensors, actuator, and as translator between hardware and software. As for the software, used to control the system by the user, is built by combining Java and C++.

Keywords: Control System, signal conditioning circuit, AT Mega 8 Microcontroller, Smart Home, Java, C++, OOAD, UML, Ultrasonic Sensor, Temperature Sensor, PIR.

ABSTRAK

Sistem kendali merupakan sebuah sistem yang dirancang dan dibangun untuk melakukan pengendalian pada suatu objek atau lingkungan. Penulis melakukan penelitian berupa perancangan sistem pengendalian peralatan elektronik yang dapat dikendalikan melalui aplikasi desktop. Aplikasi desktop yang dirancang memungkinkan pengguna untuk mengendalikan peralatan-peralatan elektronik yang terhubung kepada sistem yang telah dirancang pada penelitian ini. Aplikasi rancangan pada sistem ini juga memungkinkan pengguna memperoleh informasi dari lingkungan pengendalian dengan menggunakan sensor-sensor yang terpasang pada lingkungan sistem. Penelitian ini menggunakan metode eksperimental dengan melakukan studi literature dan studi dokumentasi. Metode perancangan sistem dan software menggunakan metode OOAD (Object Oriented Analysis and Design), dimana dalam penelitian, perancangan sistem terdiri atas 2 bagian, yaitu perancangan hardware dan perancangan software. Sistem pengendalian menggunakan rangkaian pengkondisi sinyal dan arduino. UNO sebagai bagian dari hardware yang berfungsi menerima data dari sensor, mengirimkan sinyal kepada actuator, dan sebagai penerjemah sinyal antara hardware dan software. Sedangkan untuk software, digunakan bahasa pemrograman java dan C++ dalam membangun aplikasi berbasis desktop.

Kata Kunci: Sistem Kendali, Rangkaian Pengkondisi Sinyal, Mikrokontroller AT Mega 8, Smart Home, Java, C++, OOAD, UML, Sensor PING, Sensor Suhu, PIR.

PENDAHULUAN

Sistem pengendali merupakan sistem yang melakukan pengendalian berdasarkan tujuan dari dirancangnya sistem tersebut dengan komputer sebagai pusatnya. Pada sistem pengendali, perancang dapat menggunakan sensor sebagai salah satu alat input kepada sistem. Sensor merupakan sebuah perangkat keras yang dirancang untuk dapat menerima input dari lingkungannya, tergantung apa tujuan dibuatnya sensor itu. Input yang diterima oleh sensor, akan diteruskan kedalam sistem dalam bentuk sinyal digital yang kemudian dapat dibaca oleh komputer dan diproses lebih lanjut. Secara umum sensor dikelompokkan menjadi 3 jenis, yaitu sensor thermal, sensor mekanis (gerak), dan sensor optic(cahaya) (Choi, 2011). Pada penelitian ini, penulis tertarik untuk menggunakan beberapa perangkat yang akan berguna sebagai sensor pada sistem ini, perangkat-perangkat tersebut antara lain PIR *Motion Sensor* sebagai sensor keberadaan manusia (Monk, 2011), sensor suhu, sensor cahaya, dan sensor pengukur jarak dengan inframerah dan ultasonik. Sensor-sensor tersebut akan diaplikasikan pada sistem pengendali peralatan elektronik untuk membantu dalam pengendalian sistem smart home (Oxer dan Blemings, 2009).

Perangkat lunak yang dirancang oleh penulis, tidak dapat secara langsung melakukan pengendalian terhadap peralatan elektronik yang diinginkan karena untuk dapat berkomunikasi dengan peralatan elektronik, sebuah komputer memerlukan driver yang dapat membantu komputer untuk mengenali perangkat keras yang terhubung ke komputer. Selain itu data digital yang diberikan oleh komputer sebagai output juga tidak dapat langsung dikenali oleh perangkat elektronik, sehingga penulis juga telah menyiapkan sebuah perangkat keras Rangkaian Pengkondisi Sinyal, yaitu rangkaian yang menerjemahkan sinyal digital dari komputer menjadi sinyal yang dapat men-drive perangkat elektronik sehingga peralatan elektronik akan merespon terhadap sinyal yang diberikan dari komputer. Perangkat keras tersebut berfungsi sebagai interface antara peralatan elektronik dan komputer. Perancangan perangkat lunak pada penelitian ini menggunakan bahasa pemrograman Java dengan menggunakan konsep pemrograman berorientasi objek, dan menggunakan bahasa pemrograman ASM di dalam C++.

Alat yang dibuat tentunya harus dapat dioperasikan dengan mudah oleh penggunanya. Oleh karena itu diperlukan serangkaian pengujian terhadap alat dan software yang telah dibuat (Teikari. et.al, 2012)

TINJAUAN PUSTAKA

Software Testing Guidelines

Dalam melakukan proses pengujian, ada 6 pedoman (*guideline*) pengujian software yang umum, yang patut diikuti guna meningkatkan pengujian software, antara lain (Perry 2006 : 63-34) :

- a. Testing should reduce Software Development Risk
- b. Testing should be performed effectively
- c. Testing should uncover defects
- d. Testing should be performed using business logic
- e. Testing should occur throughout the development lifecycle
- f. Testing should test both structure and function

Pengujian fungsional dikenal juga dengan pengujian *black box* (Pressman, 2009), dikarenakan pengujian dilakukan tanpa menggunakan pengetahuan logika dari sistem untuk membangun kasus pengujian. Sebagai contoh, sebuah fungsi kunci tertentu, apabila ditekan akan memberikan hasil yang spesifik, pengujian fungsional yang dilakukan adalah untuk memastikan ekspektasi ini dengan menekan fungsi kunci dan memeriksa hasilnya. Ketika melakukan pengujian fungsional, penguji akan menggunakan teknik validasi sesering mungkin.

Sebaliknya, pengujian struktural dikenal juga dengan pengujian *white box* (Pressman, 2009), karena pengetahuan dari logika internal sistem digunakan untuk membangun kasus pengujian dengan hipotesa yang dibutuhkan. Pengujian struktural secara dominan menggunakan teknik verifikasi (Ammann dan Offutt, 2008).

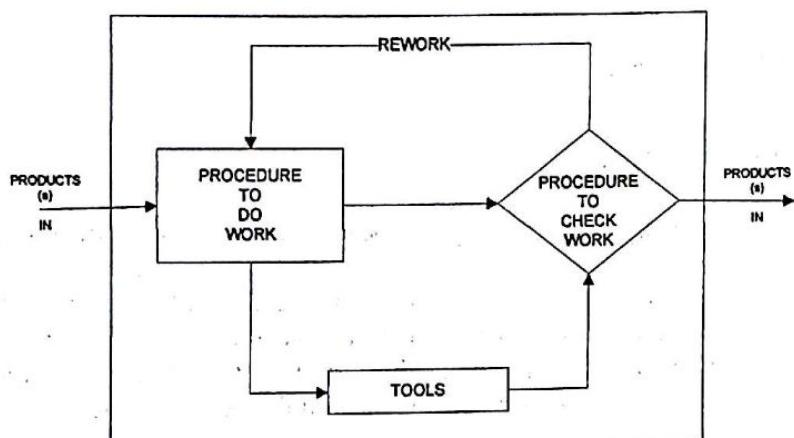
Apabila tim pengembang perangkat lunak membuat sebuah blok kode yang mengijinkan sistem untuk memproses informasi dengan cara khusus, maka sebuah tim penguji akan

memverifikasi strukturnya dengan membaca kode, dan dengan diberikannya struktur dari sistem, mereka akan memastikan apakah kode tersebut dapat berjalan dengan baik. Apabila mereka menganggap kode tersebut dapat berjalan, maka mereka akan memasukkan kode tersebut kedalam sistem dan menjalankan aplikasi untuk memvalidasi kode secara terstruktur.

Pengujian menggunakan teknik verifikasi guna mengkonfirmasi berjalan atau tidaknya sebuah implementasi kode dalam suatu sistem dengan melakukan pemeriksaan pada struktur dan logika. Pada sisi lain, mereka menggunakan teknik validasi untuk melakukan pemeriksaan fisik dari sebuah sistem untuk menentukan apakah hasil yang diinginkan dapat diperoleh dengan memberikan masukan-masukan tertentu.

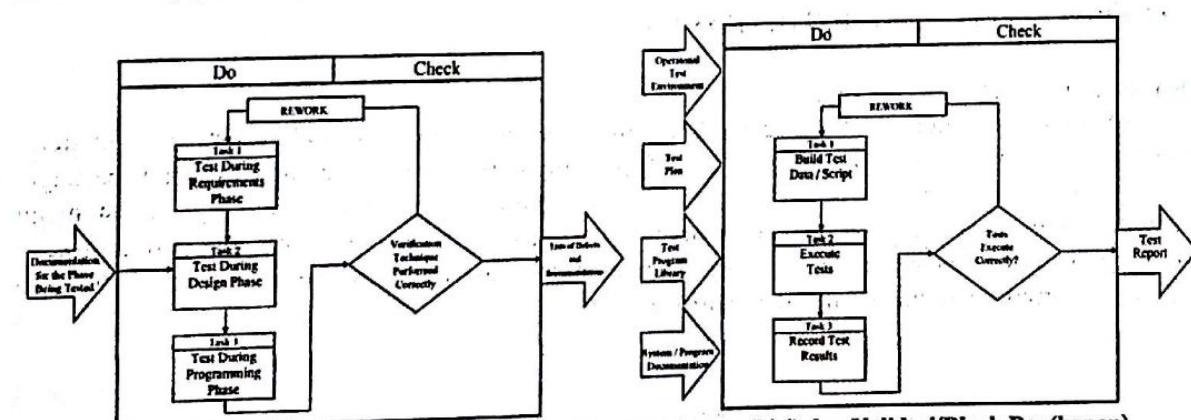
Workbench Concept

Dalam metode pengujian perangkat lunak, agar dapat mengeri konsep pengujian perangkat lunak yang baik dan benar, pengujian juga harus mengerti konsep *workbench*. Menurut Perry (2006:71), “*Workbench* adalah suatu cara untuk mengilustrasikan dan mendokumentasikan bagaimana suatu aktivitas dilakukan secara spesifik”. Dalam bidang IT, workbench merujuk kepada fase, langkah-langkah, atau tugas.



Gambar 1. Workbench untuk pengujian Perangkat Lunak

Sumber : Perry (2006 : 71)

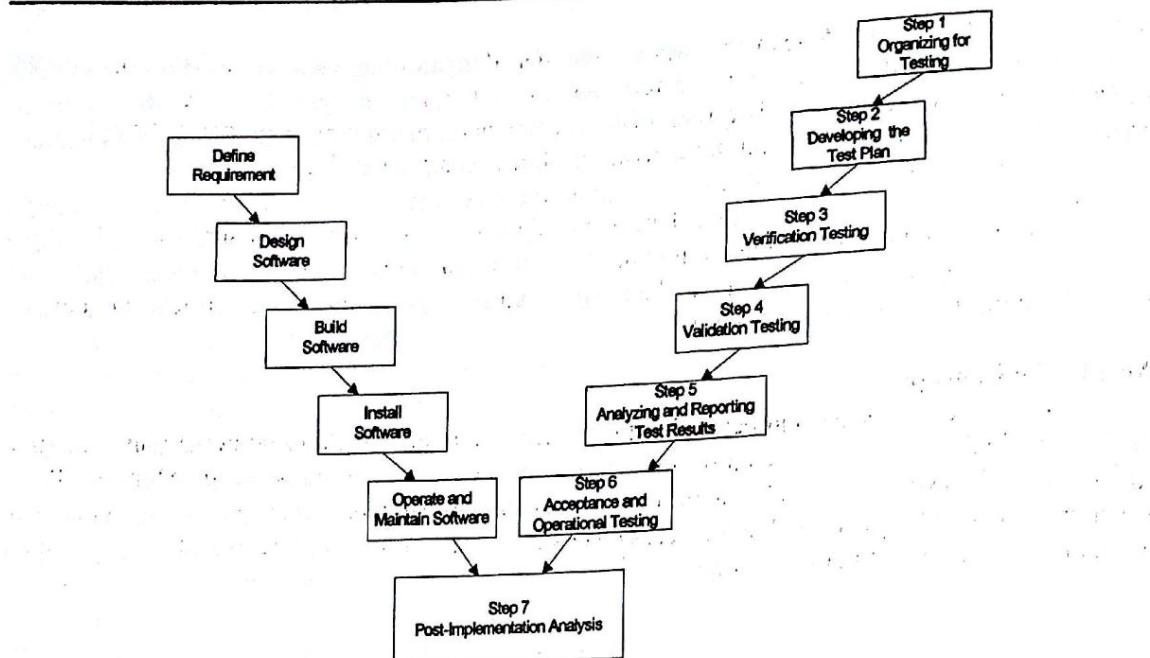


Gambar 2. Skema Workbench untuk pengujian Verifikasi/White Box(kiri) dan Validasi/Black Box(kanan)

Sumber : Perry (2006:295)

Tujuh Langkah Proses Pengujian Perangkat Lunak (The Seven Step)

Perry (2006:156-158) menjelaskan bahwa dalam proses pengujian perangkat lunak, “terdapat tujuh langkah pada proses pengembangan perangkat lunak dan proses pengujian perangkat lunak yang mana digambarkan dalam diagram berbentuk huruf V, dengan tujuan untuk menentukan apakah pengembangan dan pengujian dapat dilakukan secara lebih efektif dimasa depan” Ketika dua proses tersebut akan tampil dalam bentuk diagram V, Gambar 4 mengilustrasikan diagram V yang merupakan tujuh-langkah pengujian perangkat lunak (the Seven-Step).



Gambar 3. *Seven-step Software testing process*

Sumber : Perry (2006)

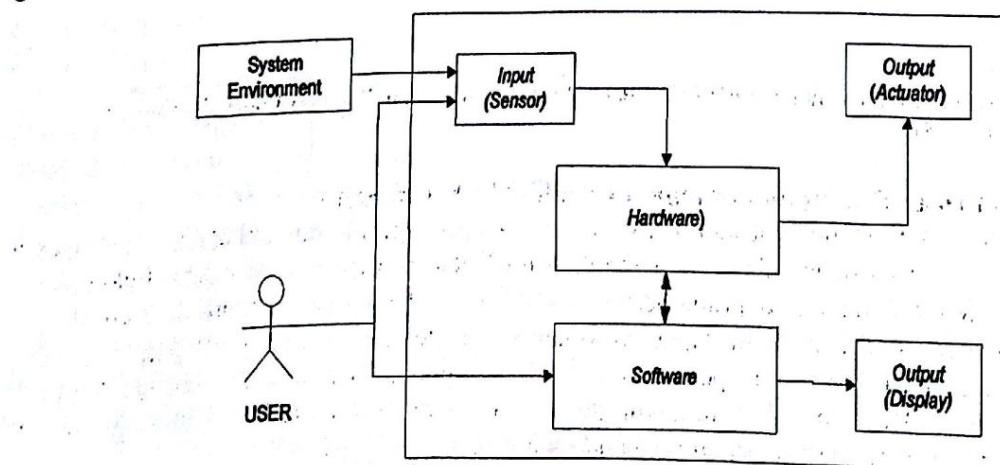
METODOLOGI PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah bentuk penelitian riset eksperimental, dimana penelitian akan dilakukan pada suatu variable, dalam hal ini adalah sistem pengendali peralatan elektronik, yang terdiri atas hardware dan software. Metode pengumpulan data yang digunakan yaitu metode pengumpulan data deskriptif dengan menggunakan jenis data sekunder. Teknik pengumpulan data yang digunakan yaitu studi literature dan studi dokumentasi. Peneliti mengumpulkan dan mempelajari dokumentasi-dokumentasi yang berhubungan dengan penelitian ini.

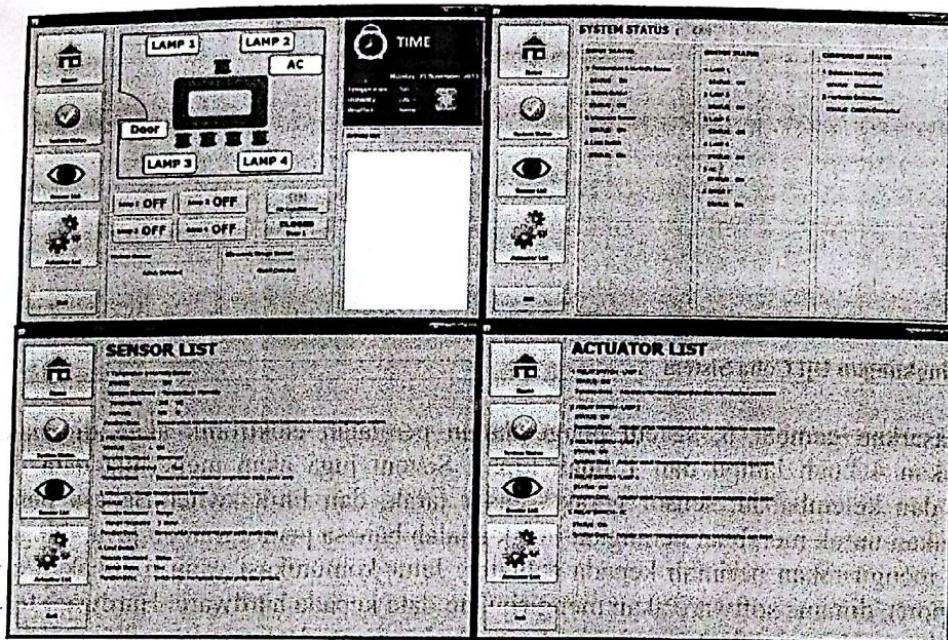
HASIL PENELITIAN DAN PEMBAHASAN

Perancangan Sistem

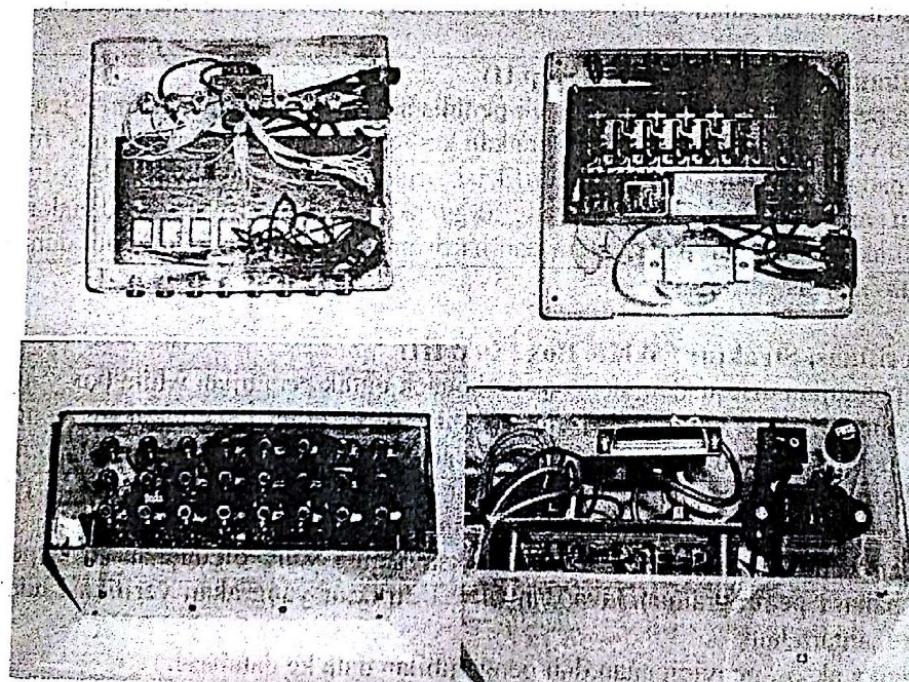
Sistem pengendalian yang akan dibangun pada penelitian ini adalah sistem pengendali peralatan elektronik yang dirancang untuk dapat melakukan pengendalian terhadap berbagai peralatan elektronik yang tergabung kedalam sistem (Sommerville, 2011), dimana pada sistem ini terdapat sensor yang dapat menerima masukan dari lingkungan sistem. Gambaran sistem secara umum dapat dilihat pada gambar 5. Hasil rancangan software dapat dilihat pada gambar 6. Pada gambar 7 menunjukkan hasil rancangan hardwarenya.



Gambar 4. Rancangan Arsitektur Sistem



Gambar 5. Hasil Rancangan Software Pengedali



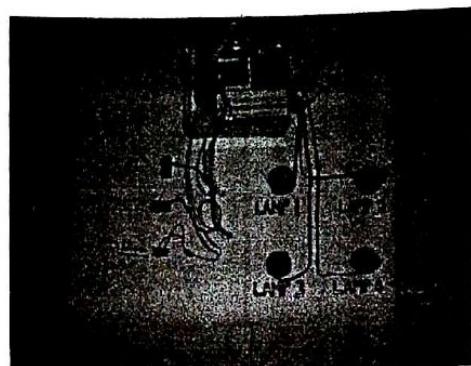
Gambar 6. Hasil Rancangan Hardware Pengendali

Pengujian Sistem

Pengujian sistem pada penelitian ini dilakukan dengan menggunakan metode seven-step seperti yang telah dijelaskan pada bab 2, namun dikarenakan penelitian ini bukanlah sebuah proyek besar yang terdiri atas sistem yang kompleks dan mendalam, maka langkah pengujian yang akan digunakan hanya sebagian dan disesuaikan dengan penggunaannya pada sistem pengendali peralatan elektronik.

Mengorganisir Pengujian (Step I)

Mengorganisir pengujian dilakukan penulis dengan menyusun sistem sesuai dengan rancangan sistem dengan menerapkan sistem pada lingkungan pengujian. Lingkungan pengujian yang dimaksud adalah penulis membangun sistem pada suatu lingkungan uji coba yang dikhawasukan untuk dilakukan pengujian pada sistem. Melalui lingkungan uji coba ini penulis akan melakukan uji coba setiap komponen sistem.



Gambar 7. Lingkungan Uji Coba Sistem

Berdasarkan gambar 6, sistem pengendalian peralatan elektronik akan digunakan untuk mengendalikan 4 buah lampu dan 1 buah kipas. Sistem juga akan menerima data dari sensor temperatur dan kelembaban, sensor gerak, sensor jarak, dan limit switch. Bahasa pemrograman yang digunakan untuk membuat perangkat lunak adalah bahasa pemrograman java, dengan bantuan C++ untuk mengirimkan perintah kepada actuator. Jalur komunikasi yang digunakan adalah port serial (usb port), dimana software akan mengirimkan data kepada hardware. Hardware terdiri atas 2 bagian, yaitu arduino UNO yang bertugas menerima data dari software dan sensor, menerjemahkannya menjadi kode digital atau analog. Bagian ke 2 nya adalah hardware yang bertugas mengirimkan data analog/digital yang telah diterjemahkan untuk dieksekusi oleh actuator.

Mengembangkan Rencana Pengujian (Step II)

Rencana pengujian yang dirancang oleh penulis adalah dengan melakukan 2 jenis pengujian, yaitu pengujian verifikasi struktur sebagai langkah ke-3 pada pengujian dan validasi fungsi sebagai langkah ke-4 pada pengujian. Pengujian verifikasi struktur akan dilakukan pada software dengan memeriksa dan memverifikasi struktur yang digunakan pada software. Sedangkan pengujian validasi fungsi akan dilakukan pada sistem dan hardware dengan memeriksa input dan output yang akan dihasilkan oleh sistem.

Pengujian Verifikasi Struktur / White Box (Step III)

Urutan pengujian disesuaikan dengan workbench untuk pengujian white box yang dijelaskan pada bab 2, dimana pada langkah pertama, kedua, dan ketiga peneliti akan menguji struktur program dari perancangan hingga menjadi software utuh yang siap digunakan, kemudian pada langkah berikutnya apabila pengujian verifikasi berjalan dengan lancar maka peneliti akan membuat daftar kerusakan dan rekomendasi yang mungkin dilakukan untuk sistem.

Pengujian yang dilakukan adalah memeriksa logika yang digunakan pada penelitian ini, yaitu dengan bahasa pemrograman Java dan C++. Struktur yang akan verifikasi adalah struktur utama software, antara lain:

1. Struktur logika akses, penyimpanan, dan pengambilan data ke database.
2. Struktur logika pembacaan dan pengiriman data melalui serial port.
3. Struktur logika pembaca sensor.

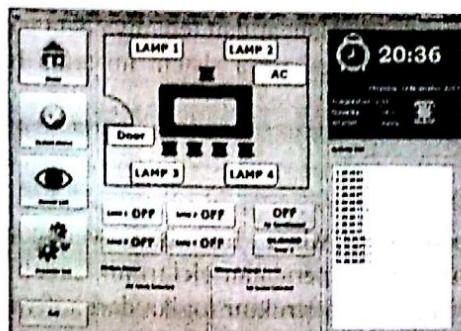
Pengujian Validasi Fungsi / Black Box (Step IV)

Urutan pengujian yang akan dilakukan, disesuaikan dengan workbench pengujian untuk black box seperti pada yang digambarkan pada bab 2. Pengujian dilakukan dengan memberikan perlakuan-perlakuan berupa input dan memeriksa apakah sistem berhasil mengolah input tersebut secara benar. Validasi fungsi dilakukan dengan memeriksa kecocokan hasil dengan hasil yang diinginkan, namun peneliti tidak akan memeriksa isi dari struktur logika lagi.

Pengujian validasi akan dilakukan dalam beberapa fungsi, antara lain :

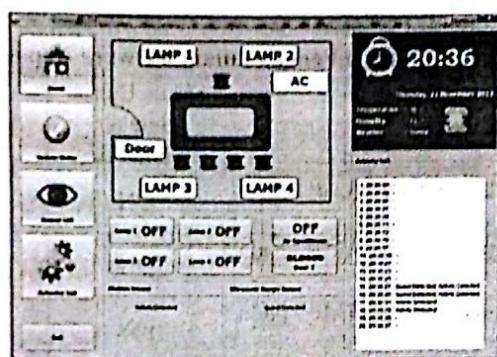
1. Fungsi input sensor kepada software.
2. Fungsi pengendalian software kepada actuator

Pengujian dimulai dengan membuka software dan membiarkannya berjalan tanpa memberikan stimuli kepada sensor-sensor. Tampilan ketika software dalam kondisi stabil dapat dilihat pada gambar 8.



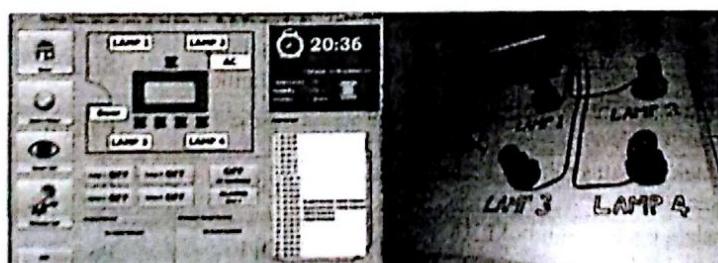
Gambar 8. Tampilan Software Sebelum Dilakukan Perlakuan.

Peneliti kemudian memberikan stimuli. Pengujian sensor jarak dilakukan dengan meletakkan objek didepan sensor, pengujian sensor PIR dengan menggerak-gerakkan tangan diatas sensor, dan pengujian sensor temperature dan kelembaban dilakukan dengan menggunakan hairdryer untuk menaikkan suhu temperature. Setelah memberikan perlakuan, tampilan software menunjukkan adanya perubahan, seperti pada Gambar 9.

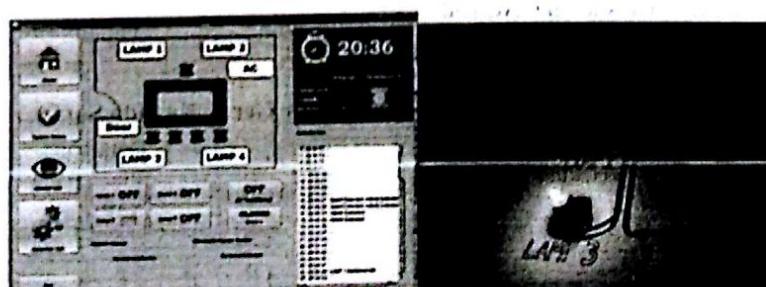


Gambar 9. Tampilan Software Setelah Diberikan Perlakuan

Pengujian berikutnya adalah melakukan validasi pengendalian software kepada actuator, yaitu menguji berhasil tidaknya fungsi software sebagai pengendali actuator. Peneliti akan melakukan pengendalian dari software dan memvalidasi hasilnya secara langsung pada actuator dengan melihat lampu yang menyala sebagai indikator keberhasilan pengendalian.



Gambar 10. (Kiri) Tampilan Software Sebelum Dikendalikan Dan (Kanan) Keadaan Sistem Uji Coba Sebelum Dikendalikan



Gambar 11. (Kiri) Tampilan Software Setelah Dikendalikan Dan (Kanan) Keadaan Sistem Uji Coba Setelah Dikendalikan.

Berjalannya pengendalian dengan lancar dan tepatnya data yang ditampilkan pada tampilan software menunjukkan bahwa fungsi-fungsi pada sistem ini telah berjalan dengan baik dan siap diimplementasikan.

Analisa Hasil Pengujian Hasil Pengujian White Box

Setelah melakukan pengujian white box atau verifikasi struktur logika, peneliti menyimpulkan bahwa sistem pengendali peralatan elektronik sudah dapat berjalan dengan lancar dengan software yang ada saat ini, dimana struktur logika dari software sudah mampu mendukung tujuan dari sistem ini, yaitu pengendalian dan pengiriman serta penerimaan data. Ketiga struktur utama yang dilakukan pengujian verifikasi struktur yaitu akses, penyimpanan, dan pengambilan data dari database, pembacaan dan pengiriman data melalui serial port, dan pembacaan data dari sensor, telah berjalan dengan baik.

Hasil Pengujian Black Box

Setelah melakukan pengujian black box atau validasi fungsi pada fungsi pengendalian software kepada actuator, dan fungsi input sensor kepada software, berikut adalah hasil yang dapat digambarkan dalam bentuk tabel

Tabel 1. Hasil Pengujian Fungsi Pengendalian Software Kepada Actuator Ke 1

No.	Perlakuan	Status Button Setelah dikendalikan	Keadaan Actuator Pada Sistem	Kesesuaian dengan harapan
1.	Peneliti menyalakan status tombol LAMP1 pada software menjadi ON	ON	Menyala	True
2.	Peneliti menyalakan status tombol LAMP2 pada software menjadi ON	ON	Menyala	True
3.	Peneliti menyalakan status tombol LAMP3 pada software menjadi ON	ON	Menyala	True
4.	Peneliti menyalakan status tombol LAMP4 pada software menjadi ON	ON	Menyala	True
5.	Peneliti menyalakan status tombol AC pada software menjadi ON	ON	Menyala	True

Tabel 2. Hasil Pengujian Fungsi Pengendalian Software Kepada Actuator Ke 2

No.	Perlakuan	Status Button Setelah dikendalikan	Keadaan Actuator Pada Sistem	Kesesuaian dengan harapan
1.	Peneliti menyalakan status tombol LAMP1 pada software menjadi OFF	OFF	Padam	True
2.	Peneliti menyalakan status tombol LAMP2 pada software menjadi OFF	OFF	Padam	True
3.	Peneliti menyalakan status tombol LAMP3 pada software menjadi OFF	OFF	Padam	True
4.	Peneliti menyalakan status tombol LAMP4 pada software menjadi OFF	OFF	Padam	True
5.	Peneliti menyalakan status tombol AC pada software menjadi OFF	OFF	Padam	True

Tabel 3. Hasil Pengujian Fungsi Input Sensor Gerak Kepada Software

No.	Lama waktu (menit)	Perlakuan Khusus	Status Sensor	Data yang dibaca pada software	Kesesuaian dengan harapan
1.	1	-	ON	-	True
2.	2	-	ON	-	True
3.	3	Peneliti menggerakkan tangan 1 meter dari sensor	ON	Movement Detected	True

4	4	-	ON	-	True
5	5	Peneliti menggerakkan tangan 2 meter dari sensor	ON	Movement Detected	True
6	6	-	ON	-	True
7	7	Peneliti menggerakkan tangan 3 meter dari sensor	ON	Movement Detected	True
8	8	-	ON	-	True
9	9	Peneliti menggerakkan tangan 5 meter dari sensor	ON	-	False
10	10	-	ON	-	True

Tabel 4. Hasil Pengujian Fungsi Input Sensor Jarak Ultrasonik Kepada Software

No.	Lama waktu (menit)	Perlakuan Khusus	Status Sensor	Data yang dibaca pada software	Kesesuaian dengan harapan
1.	1	Peneliti memberikan object didepan sensor sejauh 1 meter	ON	Ping 97cm	True
2.	2	Peneliti memberikan object didepan sensor sejauh 2 meter	ON	Ping 196cm	True
3.	3	Peneliti memberikan object didepan sensor sejauh 3 meter	ON	Ping302cm	True
4	4	Peneliti memberikan object didepan sensor sejauh 4 meter	ON	Ping 400cm	True
5	5	Peneliti memberikan object didepan sensor sejauh 5 meter	ON	Ping 400cm	True
6	6	Tidak ada object didepan sensor	ON	Ping 400cm	True
7	7	Tidak ada object didepan sensor	ON	Ping 400cm	True
8	8	Tidak ada object didepan sensor	ON	Ping 400cm	True
9	9	Object diletakkan 0,5 cm didepan sensor	ON	Ping 0cm	False
10	10	Object diletakkan 1 cm didepan sensor	ON	Ping 0cm	False

SIMPULAN

Hasil Pengujian menunjukkan bahwa Pengujian white box dan black box adalah pengujian yang saling melengkapi, terutama dalam menguji sebuah embedded system dan software, terutama untuk sistem yang terdiri dari komponen hardware dan komponen software. Melakukan Pengujian secara mendetail dan lengkap merupakan langkah terbaik dalam menentukan berjalan atau tidaknya suatu sistem dan bagaimana kedepannya sistem tersebut mampu bertahan.

DAFTAR RUJUKAN

- Ammann, Paul., Offutt, Jeff., 2008, *Introduction to Software Testing*, Cambridge University Press, Cambridge, UK, ISBN 0-52188-038-1.
- Choi, Chiu H., 2011, Reference designs for embedded controls, Proceeding 2011 Frontiers in Education Conference, Rapid City, SD, USA, October 12-October 15, ISBN: 978-1-61284-468-8, pp. S2F-1-S2F-6
- Monk, Simon., 2011, *Programming Arduino Getting Started with Sketches*, McGraw-Hill, ISBN-10: 0071784225, ISBN-13: 978-0071784221
- Oxer, Jonathan., Blemings, Hugh., 2009, *Practical Arduino: Cool Projects for Open Source Hardware (Technology in Action)*, Apress, ISBN-10: 1430224770, ISBN-13: 978-1430224778
- Pressman, Roger., 2009, *Software Engineering: A Practitioner's Approach*, 7th Edition, McGraw-Hill, ISBN-10: 0073375977, ISBN-13: 978-0073375977

- Perry, William E., 2006, *Effective Methods for Software Testing*, 3rd Edition, Wiley Publishing, Canada
- Sommerville, Ian., 2011, *Software Engineering*, 9th Edition, Pearson Education Inc., Massachusetts.
- Teikari, Petteri., Najjar, Raymond P., Malkki, Hemi., Knoblauch, Kenneth., Dumortier, Dominique., Gronfier, Claude., Cooper, Howard M., 2012, An inexpensive Arduino-based LED stimulator system for vision research, *Journal of Neuroscience Methods*, Volume 211, Issue 2, 15 November 2012, Pages 227-236