

# IMPLEMENTASI ALLJOYN FRAMEWORK PADA LINGKUNGAN RUMAH CERDAS SECARA PERVASIVE

Nur Lailatul Choiriyah, Sabriansyah Rizqika Akbar, S.T., M.Eng., Adharul Muttaqin, S.T., M.T.

Jurusan Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya

Jalan Veteran 8, Malang 65145, Indonesia

e-mail: [ilatul3@gmail.com](mailto:ilatul3@gmail.com), [sabrian.akbar@gmail.com](mailto:sabrian.akbar@gmail.com), [adharul@gmail.com](mailto:adharul@gmail.com)

**Abstrak** Komunikasi antar perangkat dengan konsep *pervasive* dalam sistem rumah cerdas sudah banyak diaplikasikan di dunia nyata. Konsep *pervasive* memungkinkan *user* dapat mengakses sistem dimana saja dan kapan saja saat dibutuhkan. Banyak *framework* yang mendukung konsep *pervasive*, diantaranya adalah DLNA, UPnP dan Bonjour. Namun *framework* tersebut masih belum mendukung komunikasi *multi-platform* dan integrasi perangkat masih rumit. Untuk itu dikembangkan teknologi baru bernama Alljoyn Framework. Alljoyn merupakan *framework* yang membuat aplikasi dapat mengenali perangkat terdekat meski menggunakan *platform* dan bahasa pemrograman yang berbeda. Untuk membuktikan Alljoyn mendukung komunikasi *multi-platform*, pada penelitian ini dibuat sistem rumah cerdas yang terdiri dari Raspberry Pi sebagai *service* dan aplikasi pada *smartphone* Android sebagai klien. Sebagai *service*, dalam sistem ini Raspberry Pi menyediakan peralatan rumah cerdas yaitu LED dan sensor. Sensor yang digunakan adalah sensor yang dapat mewakili data sensor lain, yaitu sensor gerak PIR. Komponen LED dan sensor tersebut terhubung pada pin GPIO Raspberry Pi. Komponen-komponen tersebut memiliki fungsi-fungsi yang dapat digunakan oleh klien yaitu fungsi nyala, mati, kontrol PWM LED, serta deteksi sensor. Sebagai klien aplikasi merupakan *user interface* yang digunakan oleh *user* untuk menggunakan *service* pada Raspberry Pi. Dalam berkomunikasi dan mendeteksi satu sama lain *service* dan klien menggunakan salah satu fitur Alljoyn, yaitu *Advertisement* dan *Discovery*. Alljoyn Framework yang digunakan pada penelitian ini adalah versi 15.04. Berdasarkan pengujian yang telah dilakukan dapat ditunjukkan bahwa *user* dapat melakukan kontrol terhadap perangkat-perangkat pada Raspberry Pi menggunakan aplikasi Android. Raspberry Pi juga dapat mengirimkan status keadaan perangkat kepada aplikasi. Dari hasil filter menggunakan Wireshark dapat ditunjukkan bahwa *advertisement* dan *discovery* perangkat menggunakan protokol Alljoyn Name Service.

**Kata Kunci** : alljoyn framework, rumah cerdas, sistem pervasive, raspberry pi, android, IoT

## 1. PENDAHULUAN

Internet of Things (IoT) merupakan paradigma yang menantang yang memungkinkan hal-hal seperti embedded device untuk selalu tersedia dimana saja dan kapan saja. IoT memungkinkan pengembangan user-driven applications yang menghubungkan beberapa sensor dan object untuk berkomunikasi bersama melakukan suatu fungsi tertentu sesuai dengan keinginan user. Object dan sensor terhubung melalui internet dengan menggunakan perangkat seperti Wi-Fi, Bluetooth, dan QR code. Saat ini, IoT sudah mencakup beberapa aplikasi dalam banyak bidang, salah satunya adalah adanya rumah cerdas.

Konsep Smart Home merupakan konsep yang sedang ramai diperbincangkan saat ini. Dikutip dari jurnal perusahaan Lamudi tahun 2014, Allied Market Research memperkirakan bahwa pasar untuk bangunan rumah atau gedung dengan konsep “smart” telah mengalami lonjakan nilai mencapai US\$ 7 Miliar. Angka tersebut diperkirakan akan terus tumbuh menjadi US\$ 35.5 miliar pada tahun 2020. Salah satu yang menjadi daya tarik dari

penggunaan rumah cerdas ini adalah keamanan dan kenyamanannya. Salah satu aplikasi trend masa kini adalah sistem rumah cerdas dengan konsep pervasive computing. Teknologi pervasive memungkinkan pengguna dapat mengakses sistem dimana saja, kapan saja saat dibutuhkan. Rumah cerdas dengan konsep pervasive ini sangat sesuai dan mendukung penerapan Internet of Things.

Untuk membuat sebuah sistem rumah cerdas dimana komunikasi perangkatnya menggunakan konsep *pervasive*, telah banyak *framework* yang ada saat ini. Beberapa *framework* yang ada diantaranya adalah Digital Living Network Alliance (DLNA), Universal Plug and Play (UPnP) dan Bonjour. Semua *framework* di atas menawarkan konsep komunikasi antar perangkat dengan sistem *pervasive* atau *zero configuration*. Namun *framework* di atas masih memiliki kelemahan. Kelemahan-kelemahan tersebut diantaranya adalah tidak mendukung *multi-wireless transports*, tidak mendukung *multi-platform*, manajemen jaringan yang masih cukup rumit, serta pengembangan teknologi yang lambat (Spencer, 2012).

Sebagai solusi dari permasalahan di atas, maka dikembangkanlah teknologi baru yaitu Alljoyn Framework. Alljoyn merupakan teknologi baru yang ditujukan untuk penerapan konsep IoT dan IoE. Dikutip dari jurnal berjudul Alljoyn Lambda: *an Architecture for the Management of Smart Environments in IoT*, Alljoyn merupakan *framework open source* yang dapat menjalankan aplikasi terdistribusi melalui bermacam-macam *embedded device*. Alljoyn memungkinkan *developer* dapat dengan mudah membuat aplikasi yang dapat mengenali perangkat lain yang berada di dekat perangkat tersebut. Aplikasi yang dibangun juga dapat mengkomunikasikan perangkat-perangkat secara langsung meskipun berbeda merk, protokol transport ataupun OS. Alljoyn menawarkan solusi untuk permasalahan jaringan, mobilitas, keamanan dan konfigurasi yang dinamis (Massimo Villari, 2014). Karena Alljoyn merupakan *open source* maka perkembangan teknologinya lebih cepat dari *framework IoT* yang lain.

Sebuah penelitian berhasil merancang sebuah prototype dari Smart Home dengan sistem client-server berbasis arduino uno dengan user interface android yang akan melakukan komunikasi data melalui wireless. Sistem menggunakan bahasa pemrograman C dan C++ sebagai bahasa pemrograman pada sisi server sedangkan pada sisi client menggunakan bahasa pemrograman java [1]. Dalam penelitian lain terdapat suatu sistem rumah cerdas dengan konsep smart dan pervasive yang terdiri dari perangkat kontrol, monitoring dan otomatisasi beberapa perangkat atau peralatan rumah yang saling berinteraksi dan dapat diakses melalui sebuah komputer[2]. Dalam sebuah jurnal dijelaskan bahwa karakteristik dari layanan komputasi pervasif adalah: pendekatan berorientasi layanan, deskripsi eksplisit aktivitas pengguna, alam luas, semantik, P2P, trust-awareness, dan jaringan sensor nirkabel.

Berdasarkan beberapa perkembangan teknologi di atas dan beberapa kajian yang ada maka perlu dibuat sistem rumah cerdas menggunakan aplikasi pada smartphone android sebagai pengendalinya serta sistem yang mengimplementasikan Alljoyn Framework.

## 2. DASAR TEORI ALLJOYN FRAMEWORK

Alljoyn (AJ) merupakan framework software yang open source yang memungkinkan untuk developer dapat dengan mudah membuat aplikasi yang dapat mengenali perangkat yang berada di dekat perangkat

tersebut. Aplikasi yang dibangun juga dapat mengkomunikasikan device-device secara langsung meskipun berbeda merk, protocol transport ataupun OS.

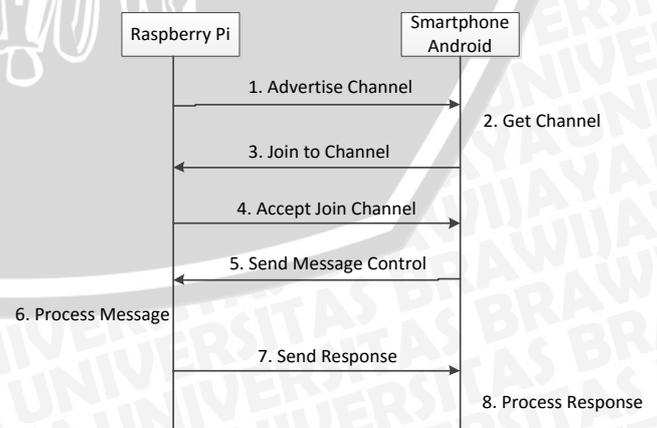
Dalam mengenali perangkat, Alljoyn Framework membuat session untuk komunikasi antar device. Session dapat berupa multiple connection session termasuk point-to-point (P2P) dan session group. Framework ini mendukung banyak mekanisme untuk keamanannya. AJ juga mendukung banyak platform mulai dari platform embedded RTOS sampai full-featured OS. AJ juga mendukung banyak bahasa pemrograman dan media transport data. Alljoyn dapat berjalan pada media Wi-Fi, Ethernet, serial, dan Power Line.

AllJoyn framework memiliki dua jenis framework yaitu Standard, untuk non-embedded devices (seperti Android, iOS, Linux) dan Thin, untuk embedded device dengan sumber daya terbatas seperti Arduino, ThreadX, Linux dengan memori yang terbatas.

Jaringan Alljoyn terdiri dari Aplikasi dan Router. Router Alljoyn dapat berjalan sendiri sebagai standalone router atau terkadang terikat dengan AllJoyn Core Library. Aplikasi Alljoyn terdiri dari Alljoyn App Code yang merupakan logic aplikasi dari Alljoyn application, AllJoyn Service Frameworks Libraries yang mengimplementasikan layanan seperti onboarding, notification, atau control panel dan AllJoyn Core Library yang menyediakan set API level paling rendah untuk berinteraksi dengan jaringan Alljoyn.

## 3. PERANCANGAN SISTEM

### 3.1. Perancangan Komunikasi Perangkat



Gambar 3.1 Model Komunikasi Sistem

Dalam sistem yang akan dibangun terdapat komunikasi yang terjadi antara Raspberry Pi dan aplikasi pada smartphone Android. User harus dapat melakukan kontrol terhadap komponen-komponen

yang terhubung dengan GPIO Raspberry Pi melalui aplikasi. Model komunikasi data pada sistem ini dijelaskan pada gambar 3.1. Pesan yang dikirim dan diterima oleh Raspberry Pi dan aplikasi Android memiliki tujuan-tujuan yang berbeda. Untuk itu perlu diberikan prefix untuk menentukan jenis pesan. Daftar prefix pesan yang digunakan dalam sistem dijelaskan dalam tabel 3.1.

**Tabel 3.1 Daftar Jenis Prefix Pesan**

Prefix Pesan	Nama Prefix	Kegunaan
D-	Device	Pesan tentang daftar device yang dapat dikontrol
A-	Action	Pesan tentang daftar kontrol yang dapat dilakukan pada device
S-	State	Pesan tentang keadaan device
E-	Event	Pesan ketika terjadi perubahan state device karena perubahan sensor input
C-	Control	Pesan tentang kontrol yang dilakukan oleh user

Tiap pesan juga memiliki id device karena terdapat beberapa device yang terhubung pada GPIO Raspberry Pi. Dengan adanya prefix dan id device maka pesan yang dikirim dan diterima oleh Raspberry Pi dan aplikasi memiliki format seperti pada Gambar 3.2.

Prefix Pesan	ID Device	Device/Action/State/Event
--------------	-----------	---------------------------

**Gambar 3.2 Format pesan**

Pesan yang dikirim berbentuk string. Dua karakter pertama adalah prefix pesan, karakter ketiga adalah id device, sedangkan karakter keempat dan seterusnya adalah isi dari pesan yang ingin dikirimkan. Detail daftar device, action, state dan control dalam sistem dijelaskan pada tabel 3.2.

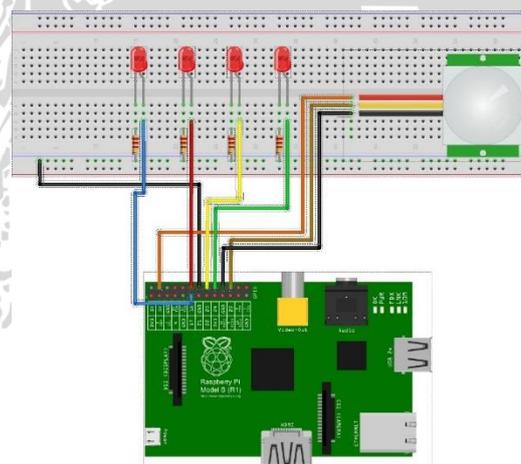
**Tabel 3.2 Daftar Device, Action, State dan Control**

Id	Nama Device	Action	State	Control
1	TV	- Matikan	- Mati	- 1
		- Nyalakan	- Nyala	- 2
2	Lampu Kamar Tidur	- Matikan	- Mati	- 1
		- Redupkan	- Nyala Redup	- 2
		- Remangkan		

		- Terangkan	- Nyala Remang	- 3
		- Nyalakan	- Nyala Terang	- 4
3	AC	- Matikan	- Mati	- 1
		- Nyalakan	- Nyala	- 2
4	Pintu	- Tutup	- Terbuka	- 1
		- Buka	- Tertutup	- 2

### 3.2. Perancangan Hardware

Hardware yang digunakan pada sistem ini adalah Raspberry Pi sebagai server dan controller, sensor pir sebagai input serta LED sebagai output. Sensor pir dan LED dihubungkan dengan Raspberry Pi melalui GPIO (General Pin Input Output). Masing-masing LED dan sensor terhubung pada pin yang berbeda. Rancangan hardware terdapat pada Gambar 3.3.



**Gambar 3.3 Skematik Perangkat Keras Sistem**

Detail peletakan komponen pada pin GPIO Raspberry Pi dijelaskan pada tabel 3.3.

**Tabel 3.3 Konfigurasi Pin GPIO Raspberry Pi**

Pin GPIO	Nama Pin	Kegunaan
2	5v DC Power	Power Sensor
6	Ground	Ground LED
11	GPIO17	LED 1 Output (hijau)
12	GPIO18	LED PWM Output (merah)
16	GPIO23	LED 2 Output (kuning)
18	GPIO24	LED Sensor Pir (hijau)

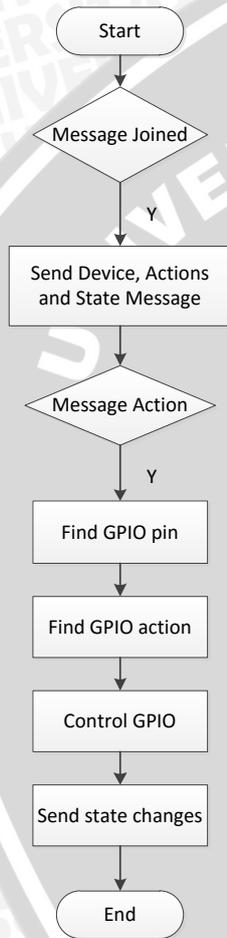


20	GROUND	Ground Sensor Pir
22	GPIO25	Sensor PIR

### 3.3. Perancangan Software

#### 3.3.1. Perancangan Program Handle Message pada Raspberry Pi

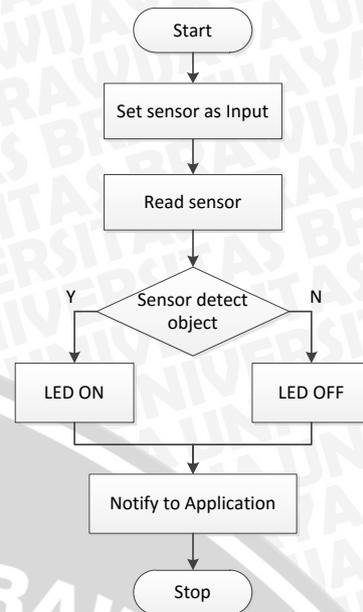
Terdapat beberapa jenis pesan yang dipertukarkan dengan masing-masing pesan memiliki prefix yang berbeda. Tiap jenis pesan diproses dengan cara yang berbeda. Alur perancangan penanganan pesan pada Raspberry Pi dijelaskan dengan gambar 3.4.



Gambar 3.4 Alur Penanganan Pesan Pada Raspberry Pi

#### 3.3.1. Perancangan Program Sensor Pir Pada Raspberry Pi

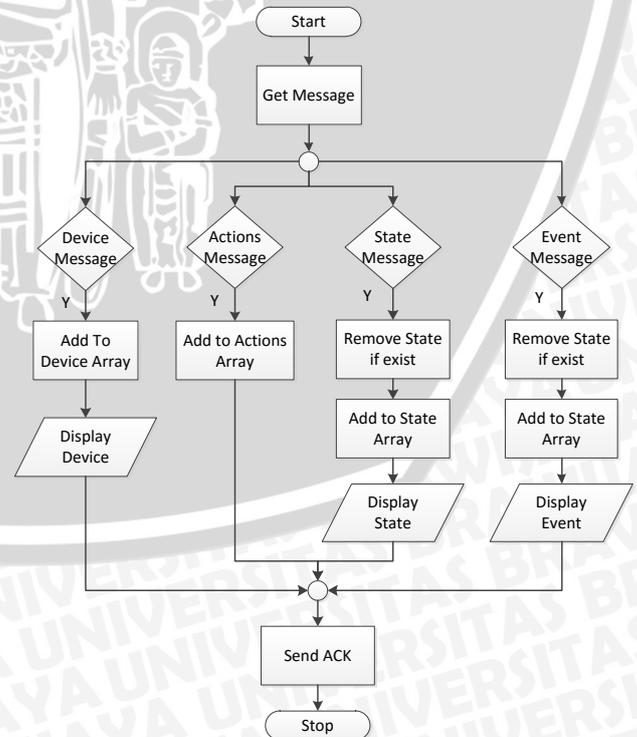
Sensor pir terhubung dengan pin GPIO Raspberry Pi. Untuk itu raspberry pi perlu mengkonfigurasi sensor ini agar dapat berfungsi sebagai input. RPI juga perlu mengkonfigurasi apa saja yang dapat dilakukan dan dihasilkan oleh sensor pir. Alur kerja konfigurasi sensor pir pada raspberry pi dijelaskan dengan gambar 3.5.



Gambar 3.5 Alur konfigurasi sensor pir pada Raspberry Pi

#### 3.3.1. Perancangan Program pada Aplikasi Android

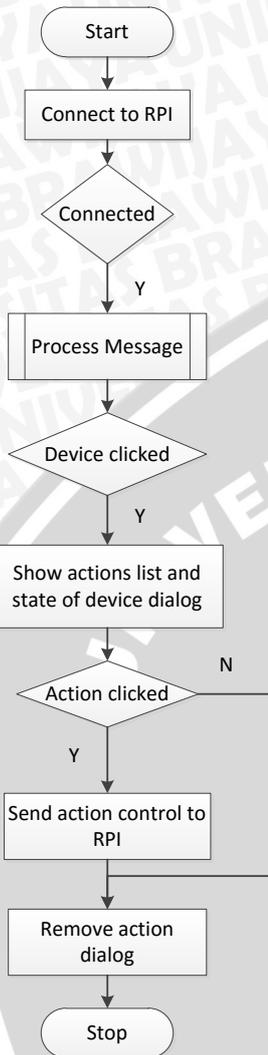
Aplikasi pada smartphone Android merupakan interface yang digunakan oleh user untuk melakukan kontrol terhadap perangkat yang terhubung dengan Raspberry Pi. Aplikasi ini juga menerapkan framework Alljoyn untuk berkomunikasi dengan RPI. Alur kerja aplikasi pada smartphone dijelaskan dengan gambar 3.7.



Gambar 3.6 Alur penanganan pesan pada aplikasi smartphone android



Sama seperti Raspberry Pi, aplikasi juga memperlakukan tiap jenis pesan dengan perlakuan yang berbeda. Alur penangan pesan pada aplikasi dijelaskan dengan gambar 3.6.



Gambar 3.7 Alur kerja aplikasi pada smartphone Andorid

#### 4. IMPLEMENTASI

##### 4.1. Implementasi Hardware

Hardware yang digunakan pada sistem ini meliputi Raspberry Pi 1b, LED Merah Kuning Hijau, Resistor dan Sensor Pir dan Smartphone. Komponen LED dan resistor diletakkan pada sebuah project board. Tabel 4.1 menjelaskan tentang spesifikasi dari Raspberry Pi 1B yang berperan sebagai controller sekaligus server Alljoyn. Raspberry Pi yang digunakan seperti pada Gambar 4.1.

Tabel 4.1 Spesifikasi Raspberry Pi 1B

1.	512 MB of RAM
2.	26 GPIO Pin

3.	100mb Ethernet port
4.	SD Card 12GB



Gambar 4.1 Raspberry Pi 1B

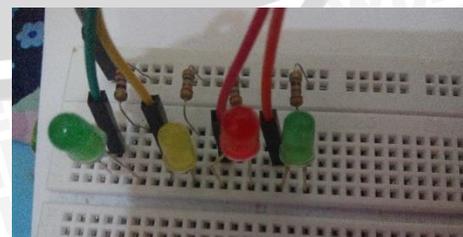
LED yang digunakan pada sistem ini berjumlah empat dengan dua LED berwarna hijau, satu berwarna kuning, dan satu berwarna merah. Masing-masing LED terhubung dengan resistor sebagai hambatan tegangan. Spesifikasi LED dan resistor ada pada Tabel 4.2. Resistor ditentukan 220 Ohm. Untuk menentukan besar resistor maka harus dihitung terlebih dahulu resistansi minimal dengan menggunakan hukum kirchoff yaitu sebesar

$$\begin{aligned}
 V_{resistor} &= V_{gpio} - V_{led} \\
 I \cdot R &= 5V - 2V \\
 0,02A \cdot R &= 3V \\
 R &= 3V / 0,02A \\
 R &= 150 Ohms.
 \end{aligned}$$

Agar tegangan yang masuk tidak terlalu besar maka resistansi ditentukan lebih dari resistansi minimal, yaitu 220 Ohms. LED dan resistor dipasang pada project board seperti pada Gambar 4.2.

Tabel 4.2 Spesifikasi LED dan Resistor

1.	2V power
2.	20 mA current
3.	530 nm LED diameter
4.	220 Ohm resistor



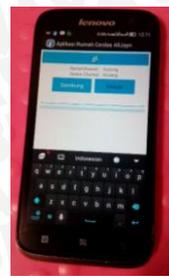
Gambar 4.2 LED dan Resistor

Tabel 4.3 menunjukkan spesifikasi sensor pir yang digunakan. Penelitian ini tidak berfokus pada

jangkauan sensing sensor. Oleh karena itu dipilih sensor dengan jarak sensing yang tidak terlalu jauh,

**Tabel 4.3 Spesifikasi Sensor Pir**

1.	5V-16V power
2.	Digital signal output is 3.3V
3.	Sensing range is about 7 meters
4.	Height 24.66mm/0.97in, Width 32.34mm/1.27in

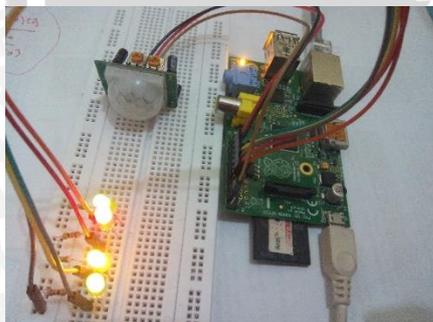


**Gambar 4.5 Smartphone Lenovo A859**



**Gambar 4.3 Sensor Pir**

Implementasi hardware secara keseluruhan digambarkan dengan gambar 4.4.



**Gambar 4.4 Implementasi Hardware Sistem**

Penelitian ini menggunakan *smartphone* sebagai perangkat untuk aplikasi. *Smartphone* yang digunakan adalah *smartphone* Lenovo A859. Gambar 4.5 menunjukkan implementasi perangkat *smartphone*. Tabel 4.4 menunjukkan spesifikasi dari *smartphone* Android yang digunakan pada penelitian ini.

**Tabel 4.4 Spesifikasi Smartphone Android**

Sumber : (Arena, 2014).

1.	1.3 GHz MTK6582 Quad Core Processor
2.	1 GB RAM
3.	5 Inch HD IPS Touch Screen Display
4.	Connectivity Bluetooth, WiFi, USB, GPS
5.	Data GPRS/EDGE/3G
6.	Operating System Android Jelly Bean v4.2
7.	Memory 8GB microSD

## 4.2 Implementasi Software Pada Raspberry Pi

Program yang dijalankan pada sisi Raspberry Pi adalah satu program. Program sudah mencakup source-codes untuk implementasi Alljoyn, source-codes untuk komunikasi RPI dan aplikasi, source-codes untuk kontrol GPIO, serta source-codes untuk pembacaan sensor. Program menggunakan bahasa C. Alljoyn framework diimplementasikan pada software dengan memanggil terlebih dahulu library-library Alljoyn seperti pada gambar 4.6.

```
#include <alljoyn/AllJoynStd.h>
#include <alljoyn/BusAttachment.h>
#include <alljoyn/BusObject.h>
#include <alljoyn/DBusStd.h>
#include <alljoyn/Init.h>
#include <alljoyn/InterfaceDescription.h>
#include <alljoyn/ProxyBusObject.h>
#include <qcc/Log.h>
#include <qcc/String.h>
#include <cassert>
#include <cstdio>
#include <cstdlib>
#include <signal.h>

#include <string>
#include <wiringPi.h> // Include WiringPi library!
#include <pthread.h>
```

**Gambar 4.6 Include library program Raspberry Pi**

Gambar 4.7 menunjukkan fungsi untuk memproses pesan jika pesan yang masuk adalah pesan “Joined”. Ketika pesan yang diterima berisi string “Joined” maka program mengirim daftar device, daftar actions dan daftar state. Semua pesan dikirim dengan format pesan yang telah ditentukan.



```

/* Jika pesan yang masuk adalah pesan Join */
if (strcasemp("Joined", responn) == 0) {
    prefix = "D-";
    for(i = 0; i < size_devices; i++){
        strcpy(ptr_message, devices[i].c_str());
        strcpy(ptr_prefix, prefix.c_str());
        snprintf(id_device, 3, "%d", i);
        SendChatSignal(strcat(strncat(ptr_prefix, id_device, 10), ptr_message)
    }
    prefix = "R-";

    for(i = 0; i < row_acts_device; i++){
        for(int j = 0; j < column_acts_device; j++){
            strcpy(ptr_message, acts_device[i][j].c_str());
            if (strlen(ptr_message) != 0){
                strcpy(ptr_prefix, prefix.c_str());
                snprintf(id_device, 3, "%d", i);

                SendChatSignal(strcat(strncat(ptr_prefix, id_device, 10), ptr_messa
            }
        }
    }

    prefix = "S-";
    int a = 0;
    for(i = 0; i < size_devices; i++){
        a = digitalRead(pins_gpio[i]);
        strcpy(ptr_message, state_device[i][a].c_str());
        strcpy(ptr_prefix, prefix.c_str());
        snprintf(id_device, 3, "%d", i);
        SendChatSignal(strcat(strncat(ptr_prefix, id_device, 10), ptr_messa
    }
}

```

Gambar 4.7 Kode Handling Message “Joined”

Jika pesan yang diterima adalah pesan control maka program akan mengidentifikasi device mana yang akan dikontrol dan action apa yang akan dilakukan. Setelah diidentifikasi maka program akan melakukan kontrol terhadap device yang terpasang pada pin GPIO. Proses handling pesan kontrol dijelaskan dengan gambar 4.8.

```

/* Jika pesan yang masuk pesan Action */
if (responn[0] == 'C') {
    /* Cari dulu yang mau dikontrol pin berapa */
    pin = (int)responn[2] - 49;
    set_pin = (int)responn[4] - 48;

    /* Kemudian lakukan control */
    if (sets_gpio[pin] == 1){
        digitalWrite(pins_gpio[pin], acts_gpio[pin][set_pin]);
        printf("digitalWrite %d Acts %d \n", pins_gpio[pin], acts_gpio[pin][set_p
    }
    else if (sets_gpio[pin] == 3){
        pwmWrite(pins_gpio[pin], acts_gpio[pin][set_pin]);
        printf("pwmWrite %d Acts %d \n", pins_gpio[pin], acts_gpio[pin][set_pin])
    }
    prefix = "S-";
    strcpy(ptr_message, state_device[pin][set_pin].c_str());
    strcpy(ptr_prefix, prefix.c_str());
    snprintf(id_device, 3, "%d", pin);
    SendChatSignal(strcat(strncat(ptr_prefix, id_device, 10), ptr_message));
}

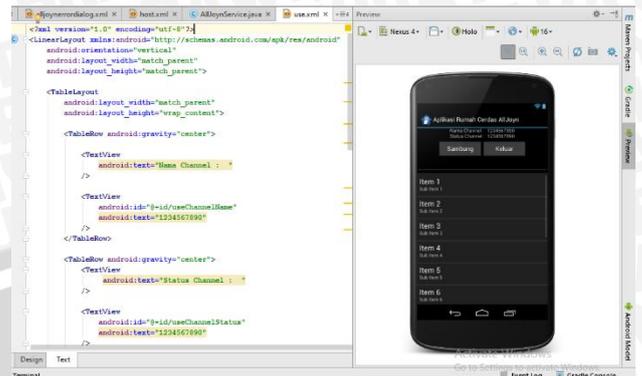
```

Gambar 4.8. Kode Handling Message Control

## 4.2 Implementasi Aplikasi pada Smartphone Android

Dalam membangun aplikasi untuk smartphone peneliti menggunakan Android Studio sebagai IDE. Langkah awal yang dilakukan adalah mendesain tampilan aplikasi. Gambar 4.9 merupakan desain untuk halaman depan aplikasi. Layout yang digunakan adalah LinierLayout. Nama channel dan status channel ditampilkan menggunakan EditText. Terdapat dua button untuk menyambungkan aplikasi

ke Raspberry Pi dan untuk memutuskan sambungan. Device list ditampilkan menggunakan ListView.



Gambar 4.9 Desain Halaman Depan Aplikasi

Untuk dapat mengimplementasikan Alljoyn pada aplikasi Android, maka library-library Alljoyn harus diimport terlebih dahulu. **Error! Reference source not found.** menunjukkan kode program untuk mengimport library-library yang dibutuhkan, baik library untuk Alljoyn maupun library untuk komponen aplikasi.

```

package org.alljoyn.bus.sample.chat;

import android.app.Notification;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Intent;
import android.os.Handler;
import android.os.HandlerThread;
import android.os.IBinder;
import android.os.Looper;
import android.os.Message;
import android.util.Log;

import org.alljoyn.bus.BusAttachment;
import org.alljoyn.bus.BusException;
import org.alljoyn.bus.BusListener;
import org.alljoyn.bus.BusObject;
import org.alljoyn.bus.MessageContext;
import org.alljoyn.bus.Mutable;
import org.alljoyn.bus.SessionListener;
import org.alljoyn.bus.SessionOpts;
import org.alljoyn.bus.SessionPortListener;
import org.alljoyn.bus.SignalEmitter;
import org.alljoyn.bus.Status;
import org.alljoyn.bus.annotation.BusSignalHandler;

```

Gambar 4.10 Import library pada aplikasi Android

Gambar 4.11 merupakan kode program untuk mengolah pesan yang masuk pada aplikasi. Pesan yang masuk memiliki tipe data String

```

public synchronized void newRemoteUserMessage(String message) {

    if(message.startsWith("A-")) {
        addAction(Module.USE, message);
    } else

    if(message.startsWith("D-")) {
        addHistoryItem(message.substring(3));
    } else

    if(message.startsWith("E-")) {
        addState(Module.USE, message);
        int id = Integer.parseInt(message.substring(2,3));
        Toast.makeText(getApplicationContext(), " " + getDevice(id) + " " +
    } else

    if(message.startsWith("E-")) {
        addEvent(Module.USE, message);
        notifyObservers(DEVICE_EVENT);
    }

    addOutboundItem("Diterima");
}

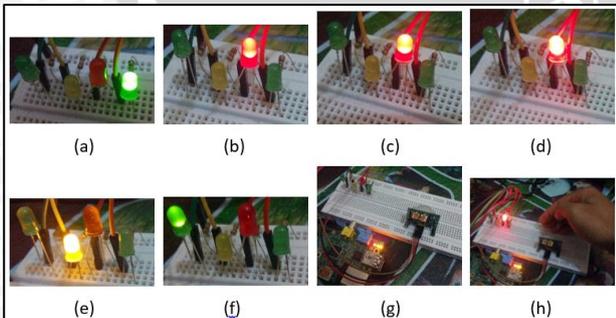
```

Gambar 4.11. Kode Handling Message

## 5. PENGUJIAN

### 5.1 Pengujian Fungsional Sensor dan LED

Pengujian ini bertujuan untuk memastikan Sensor Pir dan keempat LED dapat berfungsi dengan baik. Pengujian ini juga untuk memastikan LED dan sensor terpasang pada pin yang sesuai. Pengujian menghasilkan LED dan Sensor berfungsi dengan baik seperti pada Program pada Raspberry Pi juga menyertakan kode untuk mengecek fungsionalitas LED dan sensor pada bagian awal program. Sehingga fungsionalitas LED dan Sensor dapat dilihat ketika menjalankan program.

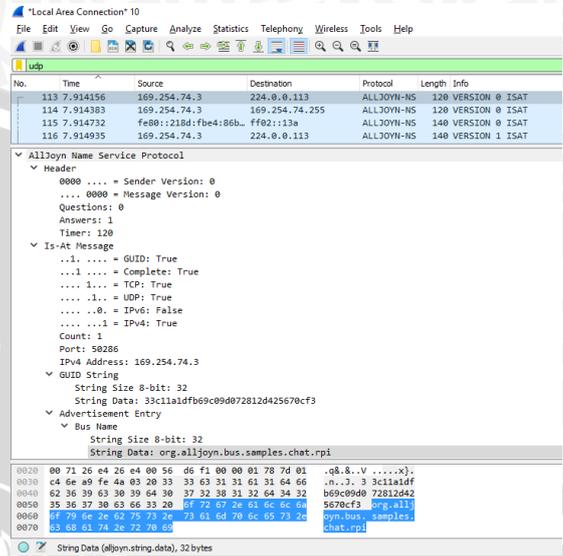


Gambar 5.1 Pengujian Fungsionalitas

### 5.2 Pengujian Alljoyn Service

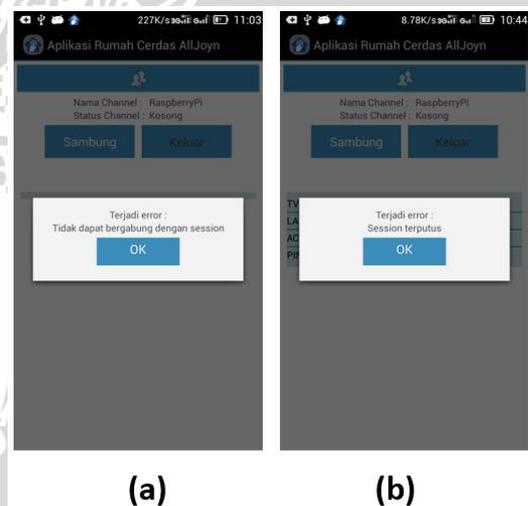
Pengujian ini bertujuan untuk memastikan framework Alljoyn berhasil diterapkan pada sistem dan dapat bekerja dengan baik. Pengujian dilakukan dengan menjalankan program Alljoyn pada Raspberry Pi. Kemudian dilakukan pengamatan menggunakan Wireshark pesan-pesan apa saja yang dikirim dan diterima dalam satu jaringan. Jika Alljoyn berhasil ditanamkan pada Raspberry Pi, maka akan terdapat protokol Alljoyn Name Service pada Wireshark yang melakukan advertise service

yang dimiliki program. Pengujian ini menunjukkan bahwa Alljoyn telah diimplementasikan pada Raspberry Pi dan Aplikasi pada Smartphone. Fungsi alljoyn yang diimplementasikan adalah fungsi Discovery dan Advertisement Service pada jaringan.



Gambar 5.2 Pengujian Alljoyn Pada Wireshark

### 5.3 Pengujian Kegagalan Alljoyn Service

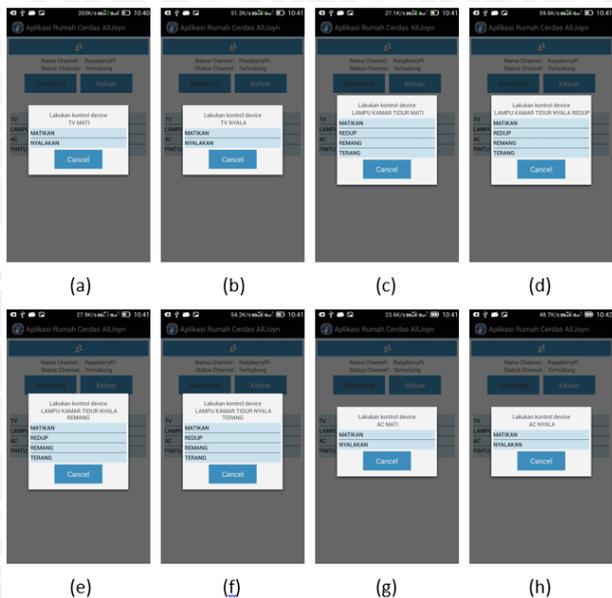


Gambar 5.3 Pengujian Kegagalan Alljoyn Service

Pengujian ini dilakukan untuk memastikan apa saja yang dapat menyebabkan service alljoyn tidak dapat berjalan dengan baik. Pengujian dilakukan pada dua program yaitu program pada aplikasi dan program pada Raspberry Pi. Pengujian ini menunjukkan ketika server mati secara-tiba-tiba atau koneksi terputus aplikasi menampilkan kotak dialog Session Terputus. Selain itu jika program RPI dan aplikasi menggunakan session port yang berbeda maka aplikasi dapat mendeteksi adanya channel dalam

jaringan, namun tidak dapat terhubung ke channel tersebut karena session port nya berbeda.

### 5.4 Pengujian Kontrol Device Melalui Aplikasi



Gambar 5.4 Pengujian Kontrol Device

Pengujian ini bertujuan untuk memastikan kontrol yang diberikan user melalui aplikasi sesuai dengan kontrol yang dilakukan oleh Raspberry Pi terhadap device-device yang terhubung pada GPIO. Pengujian juga dilakukan untuk memastikan bahwa state yang ditampilkan pada aplikasi sesuai dengan state pada device.

Tabel 5.1 Pengujian Kontrol Device

No	Kontrol	State Aplikasi	Device
1	Nyalakan TV	TV Nyala	Terkontrol
2	Matikan TV	TV Mati	Terkontrol
3	Redupkan lampu	Lampu Redup	Terkontrol
4	Remangkan lampu	Lampu Remang	Terkontrol
5	Terangkan lampu	Lampu Terang	Terkontrol
6	Matikan lampu	Lampu Mati	Terkontrol
7	Nyalakan AC	AC Nyala	Terkontrol
8	Matikan AC	AC Mati	Terkontrol

### 5.5 Pengujian Raspberry Pi mengirim pesan Event

Pengujian ini bertujuan untuk memastikan Raspberry Pi dapat memberikan notifikasi pada aplikasi ketika device input mengalami perubahan status. Dalam sistem ini Raspberry Pi harus dapat memberi informasi pada user ketika sensor pir mendeteksi adanya objek yang menyebabkan pintu terbuka (LED pintu menyala). Pengujian juga dilakukan untuk memastikan aplikasi dapat menampilkan pesan event dengan memunculkan dialog notifikasi. State device pintu (LED pintu) juga harus berubah ketika aplikasi menerima pesan event dari Raspberry Pi.

Tabel 5.2 Pengujian Pengiriman Pesan Event

No	Dialog	State Aplikasi	LED	Waktu (s)
1	Muncul	Pintu Terbuka	Nyala	1
2	Muncul	Pintu Terbuka	Nyala	1
3	Muncul	Pintu Terbuka	Nyala	3
4	Muncul	Pintu Terbuka	Nyala	3
5	Muncul	Pintu Terbuka	Nyala	1

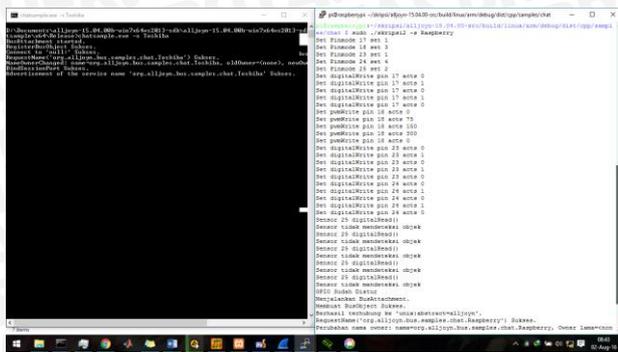


Gambar 5.5 Pengujian State Event

### 5.6 Pengujian Pervasive

Pengujian ini dilakukan untuk memastikan sistem telah menerapkan konsep *pervasive*. Pengujian ini menunjukkan ketika satu perangkat terputus atau terhubung maka informasi tentang perangkat tersebut akan terkirim ke perangkat lain. Sehingga hal ini menunjukkan bahwa sistem telah menerapkan konsep komunikasi perangkat secara *pervasive* dimana aplikasi dapat menerima informasi perangkat apa saja yang berada di dekatnya.





Gambar 5.6 Pengujian Pervasive

## 6. PENUTUP

### 6.1 Kesimpulan

1. Untuk merancang sistem peralatan rumah cerdas yang dapat dikenali secara *pervasive* dapat digunakan Alljoyn Framework. Alljoyn Framework membuat perangkat-perangkat dapat dideteksi secara *pervasive* menggunakan *advertisement* and *discovery service*. Advertisement dan discovery dilakukan dengan menggunakan *Alljoyn Name Service Protocol*.
2. Alljoyn Framework dapat diimplementasikan pada Raspberry Pi sebagai server peralatan rumah cerdas dan aplikasi pada *smartphone* Android sebagai *user interface*. Aplikasi mengirimkan pesan string untuk melakukan kontrol terhadap peralatan rumah cerdas. Raspberry Pi memberikan informasi tentang peralatan rumah cerdas kepada aplikasi. Informasi diberikan dengan prefix pesan yang berbeda-beda.
3. Alljoyn Framework mendukung komunikasi *multi-language* dan *multi-platform*. Raspberry Pi menggunakan Alljoyn Framework dengan platform sistem operasi Linux dan bahasa pemrograman C++. Aplikasi pada *smartphone* Android menggunakan Alljoyn Framework dengan platform sistem operasi Android dan bahasa pemrograman Java.
4. Pengujian fungsionalitas membuktikan sistem dapat mengenali perangkat secara *pervasive*. Pengujian kontrol peralatan rumah melalui aplikasi menunjukkan semua peralatan dapat dikontrol dan dapat diubah *state*-nya melalui aplikasi pada *smartphone*. Pengujian juga menunjukkan Raspberry Pi dapat mengirimkan informasi sesuai keadaan sensor kepada aplikasi pada *smartphone*.

### 6.2 Saran

1. Perlu dilakukan penelitian lebih lanjut tentang implementasi Alljoyn Service Framework Library.

2. Alljoyn Framework perlu diimplementasikan lebih dalam agar implementasi tidak hanya berproses pada pertukaran pesan.
3. Perlu dilakukan penelitian tentang implementasi Alljoyn Framework menggunakan Multi Controller dan Multi Application.

## DAFTAR PUSTAKA

- [1] Aditya, F. G., 2015. *Analisis Dan Perancangan Prototype Smart Home Dengan Sistem Client Server Berbasis Platform Android Melalui Komunikasi Wireless*, Bandung: Telkom University.
- [2] Alayderous, A., 2004. *Rancang bangun dan analisa sistem rumah cerdas dengan penerapan konsep pervasive*, Jakarta: Perpustakaan Universitas Indonesia.
- [3] Alliance, A., 2015. *Alljoyn Architecture*. [Online] Tersedia: <https://allseenalliance.org/framework/documentation/learn/architecture> [Diakses pada 20 Januari 2016].
- [4] Arena, F., 2014. *Fone Arena*. [Online] Tersedia: <http://www.fonearena.com/lenovo-a859-4419.html> [Diakses pada 10 July 2016].
- [5] Ebling, M. R., 2016. *Pervasive Computing and the Internet of Things*, s.l.: IEEE CS.
- [6] Ilham Megantara, M. T. B. S. N. A., 2014. *Makalah Raspberry Pi*, Bandung: Telkom University.
- [7] Lamudi, 2014. *Lamudi Indonesia*. [Online] Tersedia: <http://www.lamudi.co.id/journal/rumah-pintar-jadi-konsep-rumah-masa-depan/> [Diakses pada 20 Januari 2015].