

## IMPLEMENTASI PURWARUPA PERANGKAT RUMAH CERDAS PERVASIF BERBASIS PROTOKOL UNIVERSAL PLUG AND PLAY (UPnP) DAN RASPBERRY PI GENERAL PURPOSE INPUT/OUTPUT (GPIO)

Sabriansyah Rizzika Akbar<sup>1</sup>, Barlian Henryranu<sup>1</sup>, Maystia Tri Handono<sup>1</sup>, Achmad Basuki<sup>2</sup>

<sup>1</sup>Teknik Komputer Fakultas Ilmu Komputer Universitas Brawijaya

<sup>2</sup> Informatika/Illmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya

Email: <sup>1</sup>sabrian@ub.ac.id, <sup>1</sup>barlian@ub.ac.id, <sup>1</sup>115060900111021@mail.ub.ac.id <sup>2</sup>abazh@ub.ac.id

(Naskah masuk: 11 Juni 2015, diterima untuk diterbitkan: 22 Juli 2015)

### Abstrak

Rumah cerdas saat ini dapat terdiri dari berbagai macam peralatan pendukung didalamnya dimulai dari kontrol terhadap lampu, tv, speaker dan beberapa peralatan lainnya. Dengan semakin bervariasinya peralatan tersebut, implementasi peralatan dalam rumah cerdas tergolong rumit karena peralatan tersebut dibuat oleh berbagai vendor dan membutuhkan proses konfigurasi (terhadap jaringan) yang cukup menyita waktu. Penelitian kami menghasilkan purwarupa sistem peralatan rumah cerdas memanfaatkan mikrokomputer Raspberry pi yang mampu secara pervasif ditemukan oleh peralatan seperti komputer maupun smartphone yang dapat digunakan sebagai alat kontrol terhadap *switching* lampu memanfaatkan protokol Universal Plug and Play (UPnP) yang terintegrasi dengan Raspberry Pi *General Purpose Input/Output (GPIO)*. Pengembangan purwarupa dilakukan dengan memanfaatkan *framework* GUPnP untuk diintegrasikan dengan *library Wiring Pi* pada GPIO Raspberry Pi. Pengujian dilakukan dengan melakukan perhitungan waktu proses *switching* lampu selama 20 kali dengan menggunakan aplikasi generik *control point* UPnP Spy. Hasil penelitian menunjukkan bahwa integrasi protokol UPnP dan Raspberry Pi GPIO secara fungsional telah memiliki fitur pengenalan pervasif pada jaringan lokal dan mampu memberikan informasi perangkat dan jenis layanan yang diberikan oleh Raspberry Pi. Perubahan status pada GPIO dapat dilakukan dengan merubah parameter dari aplikasi kontrol. Berdasarkan perhitungan delay, didapatkan proses pengecekan status pada GPIO rata-rata adalah 5.1 ms, dan proses perubahan status GPIO memiliki rata-rata delay sebesar 14.8 ms. Penelitian ini diharapkan memiliki kontribusi menjembatani mekanisme pengenalan peralatan pada sistem rumah cerdas menjadi lebih pervasif dan mudah dikenali oleh peralatan yang lain dan dapat memberikan gambaran bahwa perangkat mikrokomputer Raspberry Pi dapat digunakan untuk pengembangan perangkat Rumah Cerdas yang memiliki fitur pervasif.

**Kata kunci:** *Smart home appliances, UPnP, Pervasive*

### Abstract

Smart home composed of a wide variety of appliance such as control of lights, tv, speaker, etc. With the increasing variability of the equipment, the implementation of smart home appliances is complicated caused it manufactured by various vendors that build their own method to control and find the devices. Some of the methods need to configure manually by the homeowner and it is quite time-consuming. Our research created a prototype system of smart home appliances by integrating microcomputer Raspberry Pi GPIO that able to pervasively found (the device and it services) by other control equipment such as computers and smartphones. Our prototype simulating a light switching by utilizing Universal Plug and Play (UPnP) Protocol that is integrated with the Raspberry Pi General Purpose Input / output (GPIO). Prototype development is done by utilizing the framework to be integrated which are GUPnP Library and Wiring Pi Library on the Raspberry Pi GPIO. We have done several experiment by calculating the processing time switching the lights on for 20 times using generic applications Spy UPnP control point. The results showed that the integration of the UPnP protocol and GPIO Raspberry Pi functional has had pervasive recognition features on the local network and is able to provide information devices and types of services provided by Raspberry Pi. Our prototype also able to change the GPIO status by changing status parameters from control applications. Based on the delay calculation, we obtained in the process of checking the status of GPIO need an average delay 5.1 ms, and the process of changing the status of GPIO has an average delay of 14.8 ms. This research is expected to contribute that equipment in smart home systems become more pervasif and easily recognizable by other devices. We also give an idea that the Raspberry Pi microcomputer devices can be used for the development of Smart Home devices with pervasif device and service discovery features by implementing UPnP Protocols

**Keywords:** *Smart home appliances, UPnP, Pervasive*

## 1. PENDAHULUAN

Sistem rumah cerdas dihubungkan dengan mekanisme komunikasi yang beragam. Seiring dengan perkembangan jaringan komputer dan Internet saat ini, banyak kita temui peralatan-peralatan rumah cerdas yang memanfaatkan arsitektur jaringan komputer dan Internet. Seperti halnya penelitian pengembangan sistem otomasi rumah berbasis Internet dengan *dynamic DNS service* (Zhang dkk, 2005) ataupun Sistem otomasi rumah melalui media sosial berbasis *embedded system* (Setiawan dkk, 2012). Perkembangan peralatan pada sistem rumah cerdas dan jaringan komputer saat ini, mengarah pada ranah penelitian *Internet of Things* yang menjadikan proses *discovery* peralatan-peralatan (*things*) seperti RF-ID, sensor, *smartphone*, dan lain sebagainya menjadi lebih pervasif melalui mekanisme khusus dan mampu untuk berinteraksi antara satu dengan yang lain dan saling bekerja sama untuk menggapai tujuan tertentu (Atzori dkk, 2010). Untuk membuat peralatan rumah cerdas dapat dikenali secara pervasif, kita dapat memanfaatkan protokol UPnP yang berbasis TCP/IP sesuai dengan perkembangan sistem rumah cerdas yang memanfaatkan jaringan komputer sebagai infrastruktur komunikasi. Protokol tersebut juga dipilih karena kemudahannya dalam proses pengenalan peralatan antara satu dengan yang lain dan telah menjadi arsitektur *home networking* selama 10 tahun terakhir (Kim, 2002).

Penelitian kami akan mengemukakan sebuah purwarupa peralatan rumah cerdas yang mampu dikenali secara pervasif dengan menekankan pada mekanisme kontrol terhadap lampu yang dapat dilakukan proses *switching* melalui proses pengenalan pervasif.

## 2. TINJAUAN PUSTAKA

"Internet of Things" (IoT) telah menjadi tren baru di perkembangan teknologi saat ini. Seperti halnya penelitian "*The Internet of Things: A Survey*" (Atzori dkk, 2010) mengemukakan bahwa integrasi teknologi dan solusi komunikasi, muncul sebuah konsep yang membuat berbagai macam peralatan seperti sensor (*things*) dikenali secara pervasif dan mampu berinteraksi antara satu dengan yang lain untuk memperoleh sebuah tujuan tertentu.

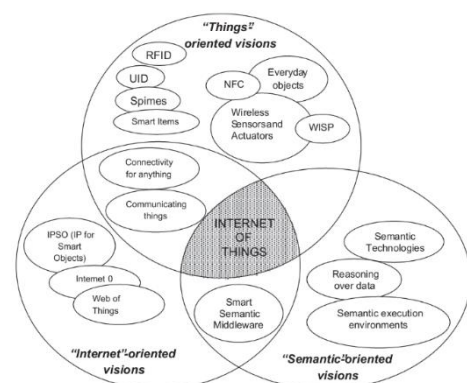
Sumber: The Internet of Things: A Survey (Atzori dkk, 2010) Gambar 1 menunjukkan bahwa IoT saat ini merupakan gabungan dari berbagai sudut pandang yang beragam. Penelitian tersebut menunjukkan bahwa proses penggabungan dari berbagai sudut pandang tersebut masih perlu dikaji oleh para peneliti. Dikarenakan permasalahan dari IoT yang masih diperlukan pengkajian lebih lanjut, peneliti akan mencoba untuk memperkecil lingkup IoT yang ada pada sebuah peralatan sistem rumah cerdas dikarenakan sistem rumah cerdas saat ini

menjadi salah satu riset topik yang terbaru dalam beberapa tahun terakhir. Pengertian dasar dari rumah cerdas adalah sistem perbaikan lingkungan tempat tinggal yang memungkinkan penghuninya untuk mengatur dan memantau temperatur, intensitas cahaya, jendela dan parameter lainnya (Shang dkk., 2012).

Di masa depan, rumah akan dipenuhi dengan peralatan-peralatan yang mampu beroperasi dan berkoordinasi untuk memudahkan pemilik rumah dalam menjalankan kegiatannya sehari-hari. Untuk menuju ke masa tersebut kita perlu menyadari bahwa beragamnya peralatan yang terdapat pada rumah akan menjadi kendala tersendiri dalam implementasi dan pengoperasian, dimana peralatan-peralatan tersebut dikembangkan oleh berbagai vendor dengan sistem yang berbeda-beda. Sebuah penelitian dengan judul "*Heterogenous device discovery framework for the smart homes*" (Warriach dkk, 2011) mengutarakan sebuah *framework* yang menangani perbedaan sistem tersebut dengan memanfaatkan protokol UPnP dan bluetooth. Penelitian ini memiliki fokus untuk menjembatani peralatan dengan basis bluetooth agar dapat dikenali antara satu dengan yang lain tetapi belum mengarah ke mekanisme kontrol terhadap peralatan-peralatan yang ada di rumah.

Protokol komunikasi, UPnP dipilih karena teknologinya yang terkait dengan rumah cerdas, *small office* dan jaringan area lokal. UPnP untuk pertama kalinya diperkenalkan oleh Microsoft di tahun 1999 dan saat ini UPnP berada di bawah Open Connectivity Foundation, sebuah organisasi yang memiliki fokus pada pembuatan spesifikasi dan pemberian sponsor pada proyek-proyek *open source* dan membantu pengembang maupun perusahaan (<http://www.openconnectivity.org>).

UPnP memiliki tujuan utama untuk menyediakan sebuah kerangka komputasi terdistribusi pada sebuah jaringan komputer skala terbatas. UPnP dapat membuat peralatan yang ada di rumah dapat melakukan konfigurasi secara otomatis dan mengatur dirinya sendiri. (Reynolds).



Gambar 1 Paradigma Internet Of Things

Sumber: (Atzori dkk, 2010)

Beberapa penelitian (Ahmed dan Ladhake, 2011; Siwen dan Yunhong, 2008) memanfaatkan mikrokontroler sebagai otak dari sebuah sistem otomasi rumah cerdas. Penggunaan sistem rumah cerdas berbasis menggunakan mikrokontroler memiliki keunggulan harga murah dan daya operasional rendah. Secara sederhana, sebuah jalur input/output mikrokontroler dihubungkan ke perangkat rumah yang ingin dikontrol. Mekanisme yang akan dilakukan sebagai kontrol pada mikrokontroler ini akan ditanam protokol UPnP sebagai penghubung antara pemilik rumah dengan peralatan yang tertanam pada sistem rumah cerdasnya.

Penelitian sebelumnya mengungkapkan bagaimana protokol pervasive dikembangkan seperti pada protokol UPnP dan aplikasi mikrokomputer yang diterapkan pada sistem rumah cerdas. Penelitian “Implementasi Purwarupa Perangkat Rumah Cerdas Pervasif Berbasis Protokol UPnP dan Raspberry Pi GPIO” memiliki kontribusi dalam mewujudkan kedua teknologi UPnP dan Mikrokomputer Raspberry Pi dalam membentuk perangkat rumah cerdas pervasive, dan melakukan pengujian waktu dalam proses kontrol yang dilakukan.

### 3. METODOLOGI PENELITIAN

Pengembangan sistem dalam penelitian ini diawali dengan studi literatur terkait dengan tinjauan pustaka dan dasar teori mengenai riset-riset sebelumnya yang terkait dengan *Internet of Things*, *UPnP* dan sistem rumah cerdas. Kemudian dilanjutkan dengan analisa kebutuhan fungsional sistem dan menggambarkan sistem secara garis besar dalam bentuk blok diagram. Blok diagram akan disusun berdasarkan masing-masing sub-sistem. Perancangan dilakukan dengan membuat masing-masing sub-sistem dan dirangkai secara utuh dalam sebuah sistem peralatan rumah cerdas memanfaatkan protokol UPnP. Pengujian sistem dilakukan secara bertahap untuk masing-masing sub-sistem yang telah dirancang sebelumnya.

#### 3.1 Analisa Kebutuhan Sistem

Analisa kebutuhan sistem diambil berdasarkan permasalahan pokok yang ada yaitu bagaimana sebuah perangkat pada rumah cerdas mampu secara otomatis mengumumkan keberadaannya dan bagaimana peralatan yang lain yang ada pada sistem mampu menerima *announcement* tanpa memerlukan konfigurasi maupun *setup* dari pengguna. Ketika peralatan yang sudah terpasang UPnP tergabung pada sebuah *network* maka peralatan tersebut akan memberikan *announcement* sehingga peralatan lain yang berada pada jaringan yang sama dapat mendapatkan dan menemukan peralatan baru tersebut.

Analisa kebutuhan sistem digambarkan secara fungsional dan global sebagai berikut:

1. Sistem rumah cerdas mampu untuk mengenali perangkat lain dengan protokol UPnP.
2. Sistem rumah cerdas mampu untuk dilakukan proses kontrol dari perangkat mobile atau perangkat yang lain.

Untuk mendukung analisa kebutuhan sistem diatas dibentuk lingkungan penelitian yang terbagi atas lingkungan penelitian perangkat keras dan perangkat lunak. Lingkungan perangkat keras yang digunakan adalah sebagai berikut:

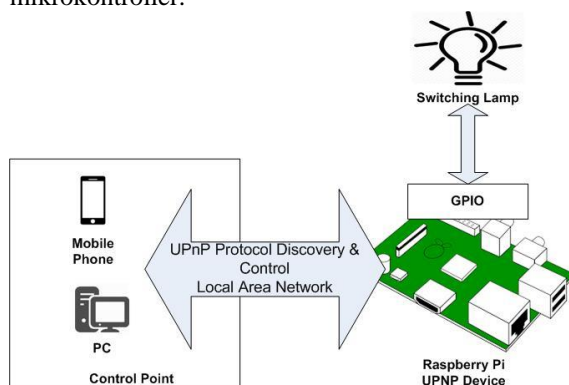
1. 1 Unit microcomputer Raspberry Pi
2. 1 Unit komputer
3. 1 Unit mobile phone atau dapat digunakan dengan perangkat raspberry pi
4. 1 Led untuk simulasi *Generic Purpose Input Output* pada Raspberry pi
5. Jaringan komputer dalam skala lokal dapat berupa *wired* atau *wireless*.

Lingkungan perangkat lunak yang digunakan adalah sebagai berikut:

1. Sistem operasi Raspbian linux yang dioperasikan pada Raspberry Pi
2. Sistem operasi Windows/Linux
3. GUPNP sebagai library UPnP

#### 3.2 Blok Diagram Sistem

Pengembangan blok diagram sistem ditunjukkan seperti pada Gambar 2. Gambar tersebut menjelaskan terdapat sebuah subsistem *smartphone* dan PC yang akan menemukan peralatan rumah cerdas secara pervasif memanfaatkan protokol UPnP dan kemudian dapat melakukan *wireless control* terhadap peralatan tersebut melalui komunikasi nirkabel. Pada subsistem peralatan rumah cerdas, lampu disimulasikan dalam *general purpose input output (GPIO)* dari mikrokontroler.



Gambar 2 Blok Diagram Purwarupa Sistem

#### 3.3 UPnP Device Description

Bagian perancangan UPnP *device description* menjelaskan bagaimana ketika peralatan yang dilengkapi dengan UPnP tergabung pada sebuah jaringan komputer. Peralatan tersebut akan mengirimkan paket *broadcast* berupa SSDP *announcement*. SSDP *announcement* tersebut mengandung URL yang mereferensikan dokumen

detail dari peralatan rumah cerdas yang baru saja bergabung. Dokumen detail tersebut berisi informasi antara lain nama *device*, tipe *device*, versi, *manufacturer*, *model name*, *number*, dan beberapa informasi yang lain. Pada dokumen deskripsi tersebut juga menyertakan layanan yang dapat diberikan oleh peralatan tersebut. Dokumen deskripsi peralatan tersebut berbentuk dokumen XML yang sudah didefinisikan terlebih dulu oleh UPnP forum. Standard template ini menyediakan beberapa layanan standar.

Perancangan aplikasi pada penelitian ini secara umum terdiri dari dua langkah yaitu aplikasi pada *control point* dan aplikasi pada *server*. Perancangan pada penelitian ini dibagi menjadi beberapa langkah antara lain perancangan aplikasi dan perangkat keras. Pada ini dibuat perancangan aplikasi untuk XML UPnP yang terdiri dari:

1. XML device description
2. XML Service List

File XML sudah disediakan sebagai template oleh Open Connectivity UPnP yang ditunjukkan pada Gambar 3.

```
<?xml version="1.0" encoding="utf-8"?>
<root xmlns="urn:schemas-upnp
org:device-1-0">
<specVersion>
<major>1</major>
<minor>0</minor>
</specVersion>
<device>
<deviceType>urn:schemas-upnp-
org:device:BinaryLight:1</deviceType>
<friendlyName>Laboratory Lamp PTIIK
UB</friendlyName>
<manufacturer>Lab Comp Network & Comp
Engineering</manufacturer>
<modelName>Virtual Light Smart
Lamp</modelName>
<UDN>uuid:cc93d8e6-6b8b-4f60-87ca-
228c36b5b0e8</UDN>
<serviceList>
<service>
<serviceType>urn:schemas-upnp-
org:service:SwitchPower:1</serviceType>
<serviceId>urn:upnp-
org:serviceId:SwitchPower:1</serviceId>
<SCPDURL>/SwitchPower1.xml</SCPDURL>
<controlURL>/SwitchPower/Control</contr
olURL>
<eventSubURL>/SwitchPower/Event</events
ubURL>
</service>
</serviceList>
</device>
</root>
```

**Gambar 3 UPnP Device Description Template**  
Sumber: (<http://openconnectivity.org/upnp>)

Perancangan untuk penelitian tersebut saat ini terdapat pada pengembangan XML untuk smart home system dengan penjelasan sebagai berikut:

1. Pada tag *specVersion* menjelaskan versi dari versi dokumen arsitektur perangkat.

2. Root *device* tag berisi metadata dari perangkat dapat pula diisi untuk sub *device*.
3. Tag *DeviceType* menspesifikasikan tipe dari perangkat.
4. Tag *FriendlyName* menjelaskan nama perangkat untuk mempermudah pembacaan bagi manusia.
5. *manufacturer* dan *model name* merupakan tag untuk pembuat dan nama model yang akan diterapkan pada perangkat.
6. Tag *UDN* atau *Unique Device Name* merupakan identifier untuk menspesifikasikan setiap peralatan secara unik. Ditambahkan dengan deskripsi *UUID* yang dapat juga diistilahkan sebagai serial number atau *MAC address*.
7. Tag *ServiceList* yang menjelaskan layanan yang diberikan oleh perangkat dimana setiap layanan akan memiliki tipe, identifier, dan *URL*.
8. Tag *SCPD URL* menjelaskan layanan secara lebih detail dan akan dijabarkan pada penelitian selanjutnya. Penelitian saat ini baru mencakup pada *Device Description XML*.

Template XML pada Gambar 3 menjelaskan bagaimana sebuah layanan diberikan untuk *smart home system device description*. Template XML pada Gambar 4 menjelaskan bahwa untuk penelitian ini mengambil *template* dasar *service list* UPnP yang telah disediakan dengan sebuah dengan jenis layanan "on" dan "off" pada lampu.

### 3.4 Skenario Pengujian Sistem

Pengujian sistem dilakukan dengan beberapa skenario antara lain:

1. Skenario pengujian fungsional:
  - a. Mendeteksi perangkat secara pervasive dengan klien berbasis PC dan Android.
  - b. Menjalankan fitur *switch* lampu pada Raspberry Pi GPIO dengan menggunakan perangkat control berbasis PC dan Android.
2. Skenario pengujian non-fungsional:
  - a. Dilakukan dengan melakukan perhitungan *delay* ketika perangkat berbasis PC melakukan aksi *invoke* atau pengecekan status pada Raspberry Pi GPIO melalui jaringan area lokal.
  - b. Dilakukan dengan melakukan perhitungan *delay* ketika perangkat berbasis PC melakukan aksi perubahan status *switch* menjadi *On* atau *Off* pada Raspberry Pi GPIO melalui jaringan area lokal.

## 4. HASIL DAN PEMBAHASAN

Pada Bab Hasil dan Pembahasan, akan dilakukan langkah-langkah antara lain analisa kebutuhan sistem, Blok diagram sistem secara lebih detail,

Perancangan alat, dan pengujian berdasarkan kebutuhan fungsional sistem.

#### 4.1 Implementasi Aplikasi

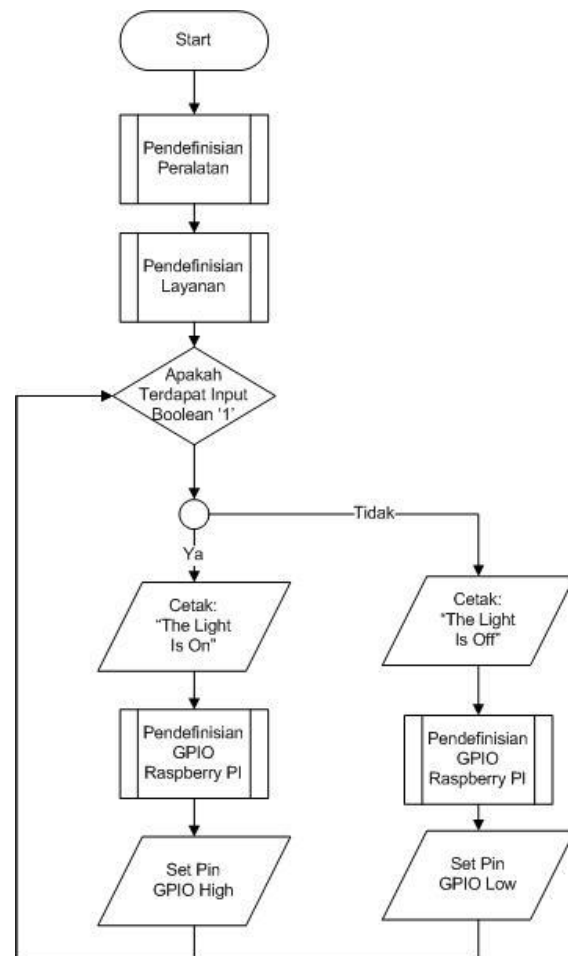
Pada sisi perangkat rumah cerdas diimplementasikan *framework* aplikasi miliki GUPNP (Gnome Developer, 2009). Diagram alir untuk perangkat rumah cerdas ditunjukkan pada Gambar 5.

```
<?xml version="1.0" encoding="utf-8"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
      <name>SetTarget</name>
      <argumentList>
        <argument>
          <name>NewTargetValue</name>
          <relatedStateVariable>Target</relatedStateVariable>
          <direction>in</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetTarget</name>
      <argumentList>
        <argument>
          <name>RetTargetValue</name>
          <relatedStateVariable>Target</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetStatus</name>
      <argumentList>
        <argument>
          <name>ResultStatus</name>
          <relatedStateVariable>Status</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="no">
      <name>Target</name>
      <dataType>boolean</dataType>
      <defaultValue>0</defaultValue>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>Status</name>
      <dataType>boolean</dataType>
      <defaultValue>0</defaultValue>
    </stateVariable>
  </serviceStateTable>
</scpd>
```

**Gambar 4 UPnP Service Description Template**  
Sumber: (<http://openconnectivity.org/upnp>)

. Pada tahap ini dilakukan deskripsi perangkat keras yang menjelaskan tipe dari perangkat, *manufacturer*, nama model, UDN seperti yang sudah

ditetapkan sebagai standar template oleh UPnP seperti yang ditunjukkan pada Gambar 4. Kemudian dilanjutkan dengan mendefinisikan layanan yang berisi versi dari layanan yang diberikan terkait dengan langkah-langkah layanan yang diberikan. Untuk penelitian saat ini ditentukan layanan yang diberikan adalah masukan berupa data *boolean* dimana nilai 1 dinyatakan untuk tanda lampu sedang menyala dan 0 merupakan masukan untuk lampu tidak menyala.



**Gambar 5 Diagram Alir Pada Sisi Perangkat Keras Sistem Rumah Cerdas (Server Side)**

Pengembangan aplikasi dilakukan dengan dua buah file yang pertama ada file UPnPCore.h dan HardwareHandler.h. File UPnPCore.h akan menangani proses pervasive system dengan menggunakan protokol UPnP dengan framework GUPnP. File HardwareHandler.h digunakan untuk melakukan penanganan terhadap perangkat GPIO pada Raspberry Pi untuk menghidupkan atau mematikan lampu pada level perangkat keras. Pada level perangkat keras desain aplikasi ditunjukkan pada Gambar 7. Dijelaskan pada gambar tersebut terdapat beberapa fungsi utama yaitu, *on\_get\_status*, *on\_get\_target*, *on\_set\_target*, dan *on\_query\_target*. Fungsi *on\_get\_target* dan *on\_get\_status* digunakan

untuk mendapatkan status perangkat lampu saat ini dan menghasilkan tampilan berupa nilai 1 (on) atau 0 (off) tergantung dengan status lampu saat ini. Fungsi `On_set_target` digunakan untuk melakukan perubahan nilai status lampu dari 1 menjadi 0 atau sebaliknya, didalam fungsi `on_set_target` terdapat fungsi `switchcontrol()` yang digunakan untuk mengirimkan data perubahan hasil ke header `HardwareHandler.H`. Gambar 6 menunjukkan perancangan aplikasi untuk memanipulasi I/O pada GPIO di Raspberry Pi. Setelah fungsi `switchcontrol()` dieksekusi, maka data akan dikirimkan menuju ke `header HardwareHandler.H` untuk melakukan aktivasi Pin 6 yang dihubungkan pada lampu untuk melakukan aktivasi HIGH untuk menghidupkan lampu, atau LOW untuk mematikan lampu. `HardwareHandler.h` memanfaatkan library `WiringPi.h` (`WiringPi`, 2015).

```
#include "../Header/HardwareHandler.h"
#include "../Header/UpnpCore.h"
#include <wiringPi.h>
const int pin = 6;
pinmode(6, OUTPUT);
void hardwareSetup() {
    wiringPiSetup();
}
void switchControl(bool setStatus) {
    switchStatus = setStatus;
    if (setStatus) {
        digitalWrite(6,HIGH);
    }
    Else{
        digitalWrite(6,LOW);
    }
}
```

**Gambar 6 Perancangan Aplikasi HardwareHandler.H**  
Sumber: (`Wiring Pi`, 2015)

```
G_MODULE_EXPORT
void on_get_status(GUPnPService *service,
GUPnPServiceAction *action, gpointer user_data){
    gupnp_service_action_set(action,
    "ResultStatus", G_TYPE_BOOLEAN, target, NULL);
    gupnp_service_action_return(action);
}

G_MODULE_EXPORT
void on_get_target(GUPnPService *service,
GUPnPServiceAction *action, gpointer user_data){
    gupnp_service_action_set(action,
    "RetTargetValue", G_TYPE_BOOLEAN, target, NULL);
    gupnp_service_action_return(action);
}

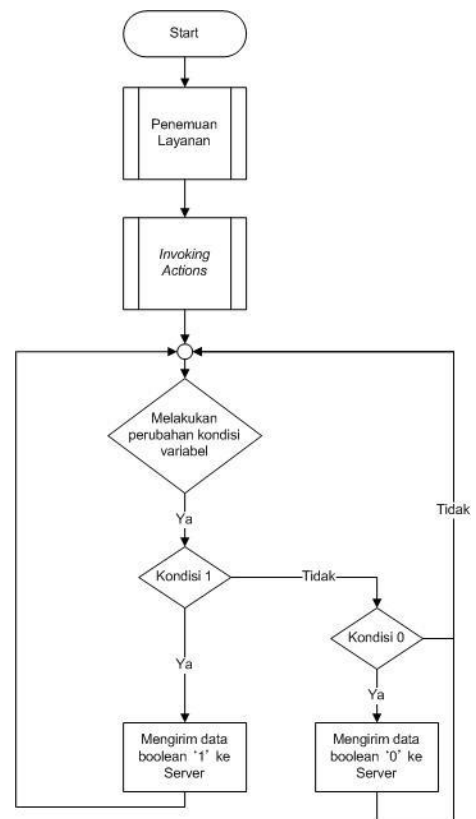
G_MODULE_EXPORT
void on_set_target(GUPnPService *service,
GUPnPServiceAction *action, gpointer user_data){
    gupnp_service_action_get(action,
    "newTargetValue", G_TYPE_BOOLEAN, &target, NULL);
    switchControl(target); //Melakukan Eksekusi
    GPIO
    g_print("status: %d\n", target);
    gupnp_service_notify (service, "Status",
    G_TYPE_BOOLEAN, target, NULL);
    gupnp_service_action_return(action);
}

G_MODULE_EXPORT
void on_query_status (GUPnPService *service,
const char *variable_name, GValue *value,
gpointer user_data){
    g_value_init (value, G_TYPE_BOOLEAN);
    g_value_set_boolean (value, target);
    g_print("query status\n");
}
```

**Gambar 7 Perancangan aplikasi UPnPCore.H**  
Sumber: (`GUPnP Framework`, 2009)

Diagram alir pada aplikasi perangkat lunak ditunjukkan pada Gambar 8. Tahapan dimulai dengan penemuan layanan dimana pada tahap ini klien akan melakukan pencarian terhadap perangkat sistem rumah cerdas dan dilanjutkan dengan *invoking actions*. Pada tahap ini klien akan mencari informasi terkait dengan layanan apa saja yang terdapat pada perangkat keras. Penggunaan aplikasi *client* menggunakan perangkat yang ada pada pengembangan UPnP Android yaitu `cling` (4<sup>th</sup> Line, 2015) dan `Spy UPnP` pada aplikasi klien berbasis PC (`Intel`, 2009).

Setelah tahapan ini selesai maka klien akan dapat mengirimkan data yang digunakan dalam merubah variabel yang terdapat pada server dimana apabila klien mengirimkan data boolean "1" maka sistem perangkat rumah cerdas akan menyalakan lampu dan apabila bernilai "0" dilakukan proses untuk mematikan lampu.



**Gambar 8 Diagram Alir Pada Sisi Perangkat Mobile UPnP**

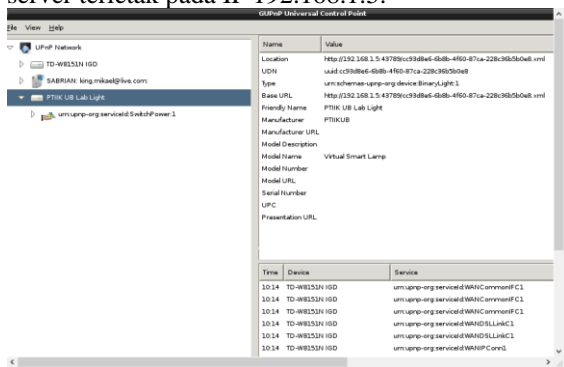
#### 4.4 Pengujian Sistem

Pengujian dilakukan dengan dua skenario yaitu pengujian fungsional dan pengujian non-fungsional pada purwarupa sistem. Pengujian fungsional dilakukan untuk melakukan pengecekan terhadap fungsi yang telah dinyatakan pada analisa kebutuhan sistem. Pengujian non-fungsional lebih ditekankan pada kemampuan Raspberry Pi dalam melakukan pengolahan data status dan respon ketika diberi masukan melalui perangkat berbasis PC dan Android.

#### 4.4.1 Pengujian Fungsional Sistem Skenario Pencarian perangkat secara pervasive pada jaringan lokal)

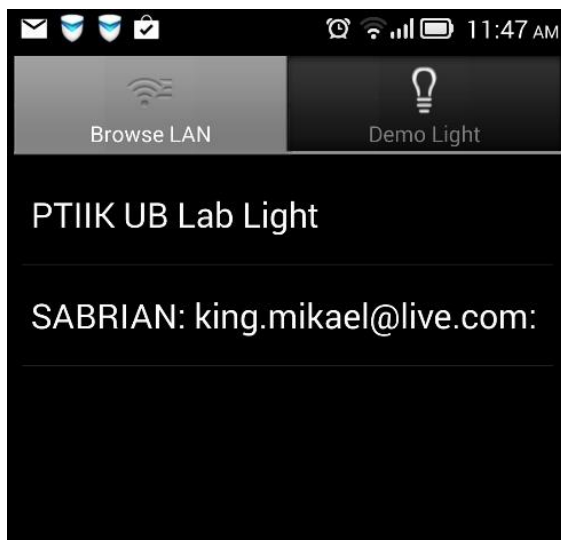
Pengujian fungsional sistem dengan skenario pencarian perangkat secara pervasive pada jaringan lokal dilakukan untuk mendeteksi apakah sebuah server/perangkat dapat di deteksi dengan menggunakan *control point* generik secara pervasif dan tanpa melakukan konfigurasi. Pengujian dilanjutkan dengan pengujian perangkat pada smartphone dengan menggunakan *android* dan Personal Computer.

Gambar 9 menunjukkan bagaimana sebuah *control point* generik dengan menggunakan *tools* UPnP Spy dan dapat dideteksi secara pervasif. Pada gambar tampak PTIIK UB Lab Light yang merupakan sebuah purwarupa lampu virtual dimana server terletak pada IP 192.168.1.5.



Gambar 9 Pengujian Deteksi Pervasif System dengan menggunakan UPnP Spy

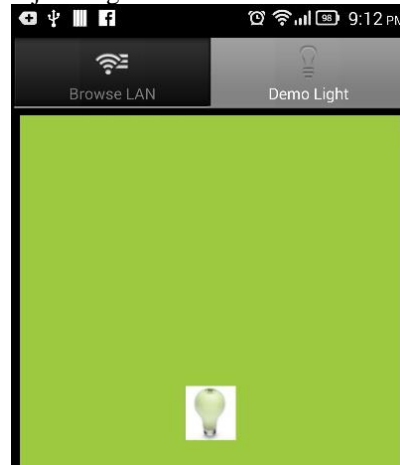
Pengujian untuk deteksi sistem secara pervasif juga harus didapatkan melalui perangkat *mobile*. Gambar 10 adalah hasil perancangan aplikasi pada klien (melalui perangkat android) disana juga tampak bahwa peralatan *virtual light* pada PTIIK UB lab light dapat dideteksi secara pervasif.



Gambar 10 Pengujian Deteksi Perangkat dengan Aplikasi Android (Cling UPnP)

#### 4.4.2 Pengujian Fungsional Sistem Skenario Perubahan Parameter untuk Nyala dan Mati

Pengujian berikutnya dilakukan untuk melakukan kontrol terhadap GPIO Raspberry Pi. Pengujian dilakukan dengan melakukan kontrol nyala lampu di android seperti yang ditunjukkan pada Gambar 11 dan ditunjukkan pada gambar tersebut Aplikasi Cling Android mengirimkan nyala lampu on pada server. Pada saat tombol pada android di tekan maka server akan menerima kode bahwa lampu dalam kondisi on dan merubah status GPIO pada Raspberry Pi dengan Pin 6 menjadi High.



Gambar 11 Lampu On Pada Android Dan Status GPIO Raspberry Pi

Gambar 12 menunjukkan ketika android mengirimkan tanda off dan Raspberry Pi melakukan perubahan status GPIO Raspberry Pi menerima dalam kondisi off dan merubah status Pin 6 pada Raspberry Pi menjadi kondisi Low.

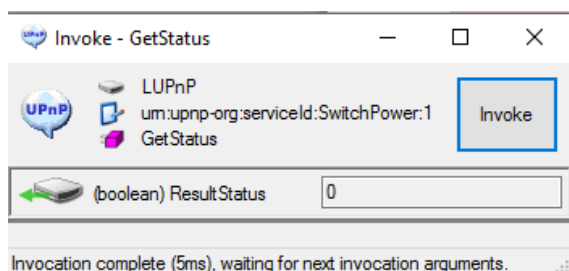


Gambar 12 Lampu Off Pada Android dan Status GPIO Raspberry Pi



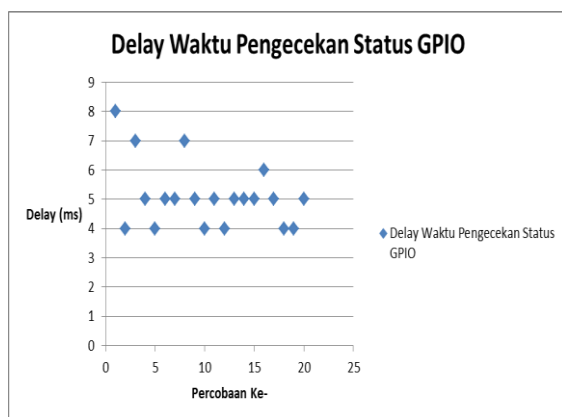
#### 4.4.2 Pengujian Non-Fungsional Sistem Skenario Penghitungan Delay Pada Pengecekan Status GPIO.

Pengujian non-fungsional dilakukan untuk melakukan pengujian performansi delay pada purwarupa sistem pada jaringan area lokal. Pengujian dilakukan dengan menggunakan aplikasi UPnP Spy untuk mendapatkan lama waktu ketika proses pengambilan delay waktu.



Gambar 13 Pengujian Perhitungan Delay Pada GPIO Raspberry Pi

Gambar 13 Menunjukkan cara melakukan control terhadap pengecekan status GPIO pada Raspberry Pi. Jika Status didapatkan bernilai 0 maka kondisi pin 6 GPIO Raspberry Pi sedang dalam kondisi Off. Status *Invocation Complete* menunjukkan waktu 5 ms yang artinya proses pengecekan status GPIO membutuhkan waktu 5 ms jika dilakukan pada jaringan area lokal. Pengujian dilakukan sebanyak 20 kali dan hasilnya ditunjukkan pada Gambar 14.

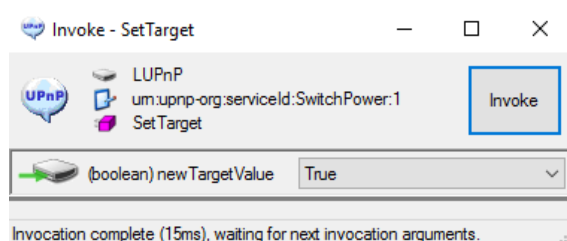


Gambar 14 Hasil Pengecekan Delay Status GPIO

Hasil pengujian menunjukkan delay rata-rata adalah 5.1 ms.

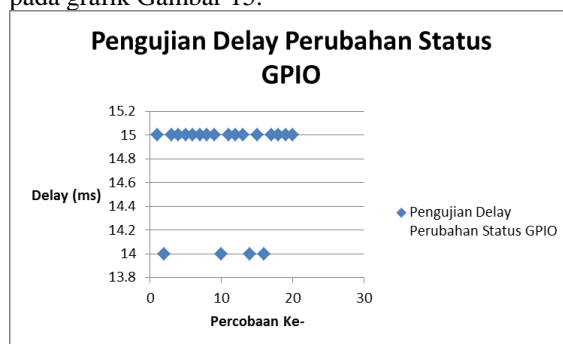
#### 4.4.2 Pengujian Non-Fungsional Sistem Skenario Penghitungan Delay Pada Perubahan Status GPIO.

Pengujian berikutnya dilakukan dengan melakukan pengujian *delay* pada perubahan status GPIO. Proses perubahan dilakukan dengan aplikasi PC UPnP Spy yang ditunjukkan pada Gambar 15. Ditunjukkan pada gambar tersebut proses perubahan status dengan delay 5ms.



Gambar 15 Pengujian Delay Pada Proses Perubahan Status GPIO

Hasil pengujian menunjukkan rata-rata delay yang dialami pada proses perubahan status GPIO adalah 14.8 ms. Hasil tiap pengujian ditunjukkan pada grafik Gambar 15.



Gambar 16 Pengujian Delay Perubahan Status pada GPIO

## 5. KESIMPULAN & SARAN

Kesimpulan yang dapat diambil dalam penelitian ini adalah proses perancangan purwarupa sistem dilakukan dengan integrasi protokol UPnP yang ditanam pada Raspberry Pi dengan menggunakan Framework GUPnP dan Framework WiringPi untuk melakukan proses perubahan status I/O pada Raspberry Pi. Pengujian sistem secara fungsional membuktikan bahwa Raspberry Pi dapat dikenali secara paspasiv melalui jaringan lokal dengan aplikasi kontrol UPnP Spy pada PC dan Cling pada Android. Sistem juga dapat menerima input dengan dua aplikasi tersebut yang akan menentukan proses perubahan status GPIO pada Raspberry Pi. Proses pengambilan status GPIO Raspberry Pi melalui UPnP menghasilkan rata-rata delay 5.1 ms dan proses perubahan status GPIO menghasilkan rata-rata delay 14.8 ms.

Saran untuk penelitian, saat ini penelitian dilakukan pada sebuah *open network* dan belum diteliti untuk mekanisme autentikasi. Autentikasi penting pada mekanisme pengenalan pervasif untuk dapat melakukan pembedaan antara perangkat pemilik satu dengan perangkat pemilik yang lain. Fitur pada smart lamp perlu ditingkatkan dengan tambahan yang lain. Pada penelitian ini hanya dilakukan simulasi untuk GPIO Raspberry Pi dalam kondisi nyala dan mati. Perlu ditambahkan juga layanan seperti *dimmer*, perubahan warna ataupun juga integrasi dengan jaringan sensor nirkabel.



## 6. DAFTAR PUSTAKA

- 4<sup>th</sup> Line, Cling, 2015–*Java/Android UPnP Library and Tools*, dari <http://4thline.org/projects/cling>
- Ahmed, V. dan Ladhake, S.A., 2011, Novel Low Cost Remote Monitoring System for Home Automation using Cell Phone, *International Conference on Computing Intelligence and Communication System*, hal. 569-573.
- Atzori, L., Iera, A. & Morabito, G., 2010. *The Internet of Things: A survey*, s.l.: Elsevier.
- Brisa Project, 2014, *Documentation for Brisa Framework*, dari <http://brisa.garage.maemo.org/documentation.html>
- D.S Kim., Lee, J.M., Kwon W.H., Yuh IK., 2002, *Design and implementation of home network systems using upnp middleware for networked appliances*. IEEE Transactions on Consumer Electronics.
- Element14, 2012, *Quick Start Guide The Raspberry Pi – Single Board Computer*.
- Gnome Developer, 2014, *Documentation for GUPnP Framework*, dari <https://developer.gnome.org/gupnp/unstable>
- Intel, 2009, *Intel Software for UPnP Technology: Technology Overview*, dari <https://software.intel.com/en-us/articles/intel-software-for-upnp-technology-technology-overview>
- Siwen, Luo., Yunhong, Li., *Design and Implementation of Home Automation System*. 2012 Fourth International Symposium on Information Science and Engineering. ISISE.
- Open Connectivity Foundation, 2016, *About UPnP*, dari <http://openconnectivity.org/upnp>
- Setiawan. E., Akbar, S.R. dan Basuki, A., 2013, *Sistem Otomasi Rumah Melalui Media Sosial Berbasis Embedded Sistem*, DIPAPTIK-Universitas Brawijaya.
- Shang, H.L., Xu, R.M. dan Yuan, J.K., 2012, A Smart Home System Based On ZigBee and IOS Software, *IEEE 18th International Conference on Parallel and Distributed System*, hal. 940-944.
- Warriach, E.U., Kaldeli, E., Bresser, J., Lazovik, A., Aiello, M, 2011, *Heterogenous Device Discovery Framework For The Smart Homes*. IEEE GCC Conference and Exhibition (GCC), Dubai, United Arab Emirates.
- WiringPi, 2015, *GPIO Interface library for the Raspberry Pi*, dari <http://wiringpi.com/>
- Reynolds, F., *The Ubiquitous Web, UPnP and Smart Homes*. Pervasive Computing Group, Cambridge.