

# PENGEMBANGAN APLIKASI *DOMOTIC* BERBASIS ARSITEKTUR BERORIENTASI LAYANAN DENGAN ANDROID SDK SEBAGAI NILAI TAMBAH LAYANAN BERBASIS INTERNET

Widy Agung Priasmoro, Widyawan, Bimo Sunarfri Hantono

Jurusan Teknik Elektro dan Teknologi Informasi, Fakultas Teknik  
Universitas Gadjah Mada Yogyakarta

[widy.agung@gmail.com](mailto:widy.agung@gmail.com)

## Abstrak

*Kebutuhan akses ke peralatan di rumah sangat dibutuhkan, terutama bagi mereka penghuni rumah yang menginginkan akses cepat mengetahui informasi keadaan rumahnya atau melakukan aktifitas mengatur beberapa peralatan di rumah. Namun, hal tersebut akan mempersulit penghuni rumah apabila sedang berpergian jauh. Untuk itu, diperlukan sebuah kendali jarak-jauh yang dapat memudahkan penghuni rumah agar dapat dengan mudah mematikan atau menyalakan lampu, mengatur suhu udara ruangan dan memantau keadaan rumah melalui kamera. Di dalam penelitian ini dipaparkan mengenai pengembangan Domestic Robotic dengan arsitektur berbasis layanan. Penelitian ini akan dilakukan pada pengembangan dan implementasi aplikasi client berbasis mobile Android sebagai remote control home automation dan web-service RESTful yang terpasang di home-gateway. Pemilihan smartphone Android sebagai remote control dikarenakan biaya yang terjangkau untuk berbagai kalangan konsumen, selain itu bagi para developer, pengembangan aplikasinya tidak membutuhkan banyak biaya. Penelitian ini dilakukan di tim JTETI UGM, dengan hasil keluaran berupa sistem rumah pintar (smart-home) yang dapat dikendalikan dan dipantau dengan menggunakan perangkat Android.*

**Kata Kunci:** *domotic, home automation, web-service, home-gateway, RESTful, Android.*

## 1. Pendahuluan

Pengelolaan rumah yang cepat dan praktis adalah impian semua kalangan masyarakat. Hal-hal yang masyarakat inginkan dalam pengelolaan rumah adalah meningkatkan kualitas hidup (dari segi kenyamanan, keamanan dan hemat energi) (Gonzalez, et al., 2001).

*Domotic* atau *Domestic Robotic* menjadi solusi pengelolaan rumah secara pintar (*smart home*). Teknologi ini dapat diaplikasikan dengan berbagai peralatan rumah, di mana teknis pengelolaan rumah dibuat menjadi otomatis melalui sebuah *remote control*, salah satunya *smartphone* Android. Akses *Domotic* melalui *device* atau perangkat digital yang berbeda, diketahui mampu melakukan kontrol dengan cepat dan lebih personal (Greaves, 2002).

Beberapa aplikasi komersial untuk *Domotic* sudah tersedia. Akan tetapi, sistem yang sudah ada merupakan sistem yang tertutup (*closed system*) dengan interoperabilitas yang terbatas.

Penelitian ini bervisi bahwa arsitektur berbasis layanan atau *Service-Oriented Architecture* (SOA) merupakan solusi untuk mengatasi permasalahan heterogenitas dan interoperabilitas *Domotic*. Dalam konsep interoperabilitas berbasis SOA, setiap entitas *Domotic* merupakan penyedia layanan (*service provider*). Dan pihak lain, sebagai contoh: *server* terpusat, menggunakan layanan tersebut dan dapat mengintegrasikannya sebagai bentuk layanan baru (Delgado, et al., 2006). Salah satu contoh layanan yang dapat ditawarkan adalah *video surveillance* jarak-jauh untuk perumahan dan kantor. Dengan berbekal perangkat *mobile* Android yang terhubung koneksi internet ataupun jaringan intranet WAN (*Wide Area Network*), pengguna mampu memantau ruangan melalui *IP-cam* dan mengetahui status perangkat elektronik seperti lampu ataupun komputer yang berada di rumah atau kantornya.

Penelitian ini berusaha mencoba menerapkan prototipe *remote control Domotic* dengan menggunakan perangkat *mobile* Android dan Android SDK, di mana memanfaatkan *web-service* RESTful dengan PHP sebagai media untuk berkomunikasi dengan *gateway* yang berada di rumah melalui koneksi internet dengan port HTTP.

## 2. Studi Literatur

### 2.1 *Domotic*

*Domotic* adalah suatu sistem di mana rumah (*domos*) bertemu dengan teknologi informasi dalam berbagai bentuk (*informatic*, namun dapat juga *mechanism*, *ergonomic* dan *communication*) (Aiello & Dustdar, 2008). *Domotic* digunakan untuk memantau dan atau mengatur sebuah bangunan dengan tujuan akhir berupa peningkatan kenyamanan, keamanan, komunikasi dan penghematan energi (Caytiles & Park, 2012).

### 2.2 *Web Service*

*Web service* adalah keluarga basis standar XML yang didesain untuk dapat berkomunikasi secara bebas, dinamis terhadap aplikasi. *Web service* dapat dilihat sebagai evolusi dari perkembangan *web*. Dalam hal ini, manusia dapat berinteraksi dengan aplikasi via HTML dan suatu aplikasi secara independen dari sistem operasi, protokol, dan bahasa pemrograman mampu berinteraksi dengan aplikasi lain (Breitman, et al., 2007).

### 2.3 Arsitektur Berbasis Layanan

Secara konsep, Arsitektur berbasis layanan (SOA) adalah gaya arsitektur yang menggunakan komponen sistem di dalam perangkat lunak untuk

menyediakan akses ke jaringan. Layanan ini menggunakan pesan-pesan untuk pertukaran permohonan yaitu meminta atau merespon informasi dalam bentuk dokumen yang jelas yang membuat asumsi sangat sedikit tentang teknologi yang berkemampuan sebagai penerima.

Secara teknologi, arsitektur berbasis layanan adalah bahasa pemrograman XML yang mendefinisikan arsitektur pesan dan format pesan, sehingga memberikan protokol pengolahan dasar. Dokumen XML ini berisi *header* dan *body*. *Header* digunakan untuk *routing* (contoh: pengalamatan) dan konfigurasi *Quality of Service* (QoS) (contoh: transaksi, keamanan, dan kehandalan). *Body* berisi pesan yang bermuatan (Pautasso, et al., 2008).

### 3. Metodologi Penelitian

Penelitian dilakukan dengan beberapa tahap, yaitu :

- a. Studi Pustaka: mengidentifikasi permasalahan pada objek penelitian, kemudian dilanjutkan dengan mencari informasi seputar teknik penyelesaian masalah tersebut.
- b. Perumusan masalah dan menentukan tujuan penelitian: langkah ini melakukan perumusan masalah berdasarkan latar belakang dan informasi dari berbagai sumber kemudian menentukan tujuan dari penelitian yang akan dilakukan.
- c. Pembuatan model perancangan perangkat lunak: melakukan perancangan perangkat lunak yang sesuai dengan tujuan dan pertanyaan penelitian. Perancangan tersebut berupa gambaran dari model yang akan dibuat ke dalam sebuah aplikasi.
- d. Membangun arsitektur berbasis layanan dengan RESTful: di dalam tahap ini, pengembangan *web service* RESTful menggunakan bahasa pemrograman PHP. *Web service* ini akan dipasang di *home gateway* yang berada di rumah, di mana *home gateway* tersebut terhubung dengan perangkat elektronik yaitu lampu dan *IP-cam*.
- e. Membangun aplikasi *client mobile* Android dengan SDK: di dalam tahap ini, aplikasi Android dikembangkan dengan memanfaatkan *web service* RESTful yang dikembangkan sebelumnya. Aplikasi *client android* ini dikembangkan dengan menggunakan Android SDK dan bahasa pemrograman Java. Fungsi dari aplikasi *client* ini adalah sebagai *remote control* yang mampu memantau status perangkat elektronik di rumah.

- f. Evaluasi Sistem: pada proses ini dilakukan penganalisisan *web service* dan aplikasi *client* yang telah dibuat untuk mengetahui apakah prototipe sistem tersebut sudah sesuai dengan harapan. Salah satu cara mengevaluasi prototipe sistem ini adalah dengan melakukan analisis percobaan terhadap sistem. Dari analisis percobaan ini akan diketahui apakah penyelesaian kasus dalam aplikasi ini hasilnya akan sesuai yang diharapkan setelah dilakukan pengujian. Jadi dari hal tersebut, akan diketahui apakah sistem ini sudah sesuai yang diharapkan atau belum.

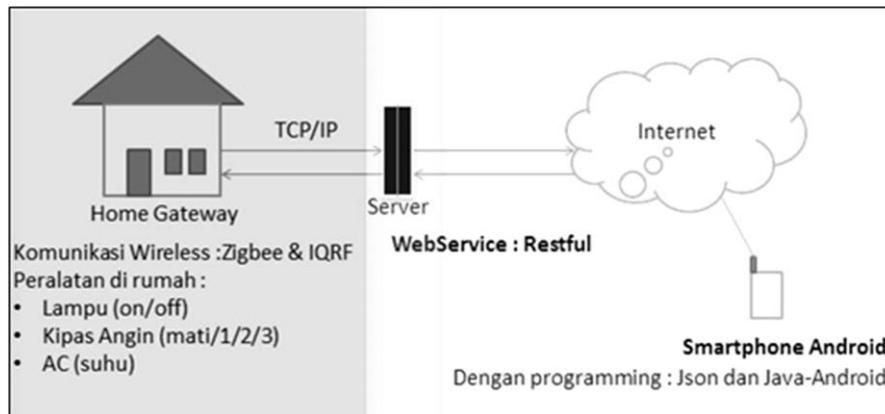
### 3.1 Analisis Kebutuhan Sistem

Analisis kebutuhan perangkat lunak merupakan analisis terhadap semua kebutuhan yang diperlukan dalam lingkup perangkat lunak yang akan dikembangkan. Analisis kebutuhan yang dilakukan terhadap perangkat lunak akan menghasilkan spesifikasi kebutuhan perangkat lunak tersebut. Analisis kebutuhan perangkat lunak merupakan hal yang sangat penting sekali karena apabila dalam melakukan analisis kebutuhan sistem terjadi kesalahan, maka hal ini akan menyebabkan kesalahan pada tahap berikutnya.

Salah satu tujuan dari analisis kebutuhan perangkat lunak adalah dapat menentukan apakah perangkat lunak yang akan dibangun dapat mencapai tujuan yang diinginkan. Analisis kebutuhan perangkat lunak ini dipakai untuk melakukan analisis seluruh kebutuhan perangkat lunak sehingga dapat membantu dalam proses pengambilan keputusan yang optimal.

## 4. Cara Kerja Sistem

Perancangan sistem *Domotic* ini melibatkan komunikasi *wireless* menggunakan perangkat lunak Zigbee dan IQRF yang menghubungkan peralatan rumah dengan *home-gateway*. Adapun peralatan rumah yang terhubung ke *home-gateway*, yaitu lampu dengan indikator *on/off*, kipas angin dengan indikator mati, *speed 1*, *speed 2*, dan *speed 3* dan *Air Conditioner (AC)* dengan indikator suhu. Agar sistem *Domotic* ini dapat di-remote dengan menggunakan *smartphone* Android, dibutuhkan sebuah *web-service*, yaitu RESTful. Dari sisi *server* yang terhubung ke internet dipasang API REST dengan bahasa pemrograman PHP yang dapat mengirim *response* atau menerima *request* dari aplikasi *client Domotic* di *smartphone* Android. Gambar rancangan *smart home system* JTETI UGM dapat dilihat pada Gambar 1.



**Gambar 1** Rancangan sistem rumah pintar

*Remote control* berupa perangkat *smartphone* Android melakukan *request* ke *server* untuk satu aktifitas yang memberi kondisi ke *node* yang berada di rumah. Setiap *node* yang terhubung dengan *home gateway* memiliki *ID device* yang tersimpan di *server*. Apabila *ID* tersebut sama dengan *ID request*, maka *node* diminta mengubah kondisinya dengan cara meneruskan *request* dari *server* ke *home gateway* yang selanjutnya mengubah *state* melalui komponen *microcontroller* yang terpasang di peralatan rumah. Ketika *state node* tersebut sudah diubah, *server* menyimpan *state* tersebut dengan cara *update* tabel *device*, dan memberikan *response* ke perangkat *remote control*.

#### 4.1 RESTful

Pemilihan *web service* RESTful sebagai media komunikasi data, karena RESTful memiliki struktur *web service* yang sederhana. Menurut Pautasso, dkk. RESTful *Web Service* dianggap sederhana karena RESTful telah mencakup standar W3C/IETF (HTTP, XML, URI, MIME) dan infrastrukturnya telah dikenal luas (*pervasive*). *Client* dan *server* HTTP telah tersedia untuk semua jenis bahasa pemrograman dan sistem operasi maupun *platform* serta *port* default HTTP, yaitu *port* 80 umumnya dibiarkan terbuka oleh konfigurasi *firewall*. Keunggulan lainnya adalah, RESTful tergolong infrastruktur yang ringan, di mana layanan dapat dibangun dengan peralatan yang minimum. Dampaknya adalah *developer* dapat mencoba layanan RESTful ini langsung melalui *web browser* tanpa harus membangun aplikasi khusus dari sisi *client* (Pautasso, et al., 2008).

## 5. Perancangan dan Implementasi Aplikasi *Remote Control*

### 5.1 Deskripsi Umum Aplikasi

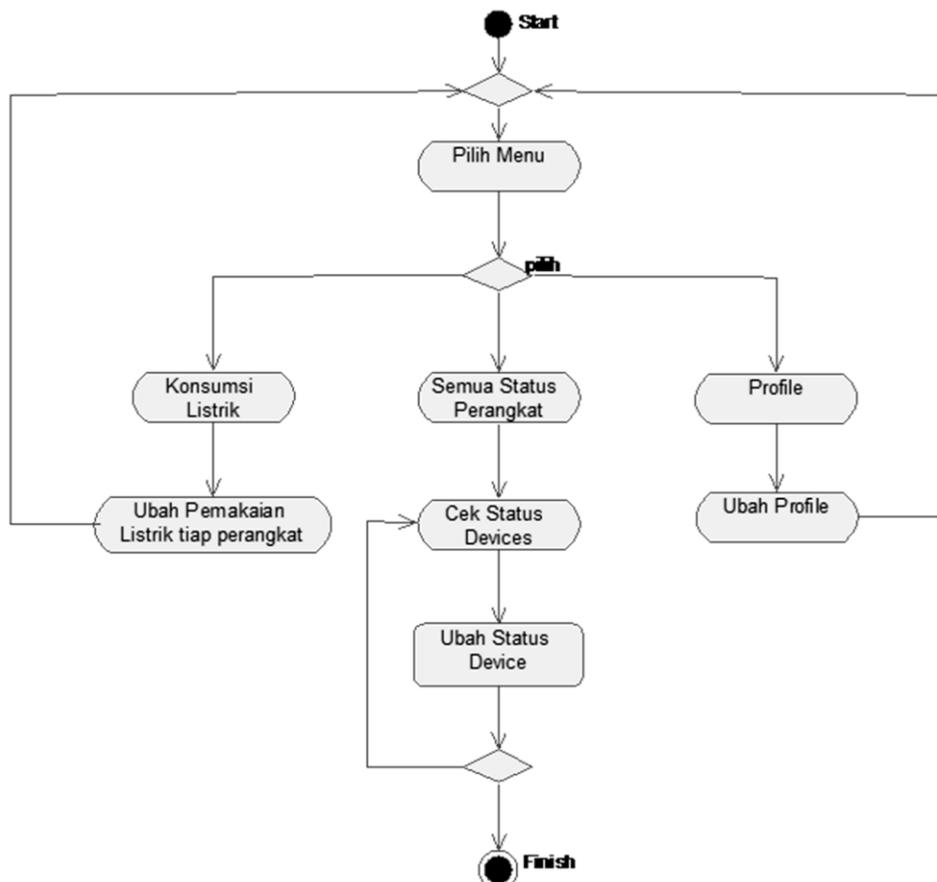
Aplikasi yang akan dikembangkan merupakan sebuah aplikasi berbasis *mobile* Android, pada intinya memiliki dua fungsi, yaitu melakukan *request* ke

server melalui *web service* agar dapat memberi perintah ke *home-gateway* serta menerima *response* dari *home-gateway* yang disajikan dalam bentuk informasi status setiap perangkat yang terhubung ke *home-gateway*.

## 5.2 Hasil Perancangan

### 5.2.1 Activity Diagram

*Activity diagram* aplikasi *remote control* berbasis Android ini menggambarkan semua aktivitas global yang terjadi dalam sebuah perangkat lunak. Pengguna diberi beberapa pilihan menu: *profile*, semua status perangkat rumah dan konsumsi listrik. Gambar 2 menunjukkan *activity diagram* perangkat lunak *remote control Domotic* ini.



**Gambar 2** Activity Diagram Remote Control Domotic

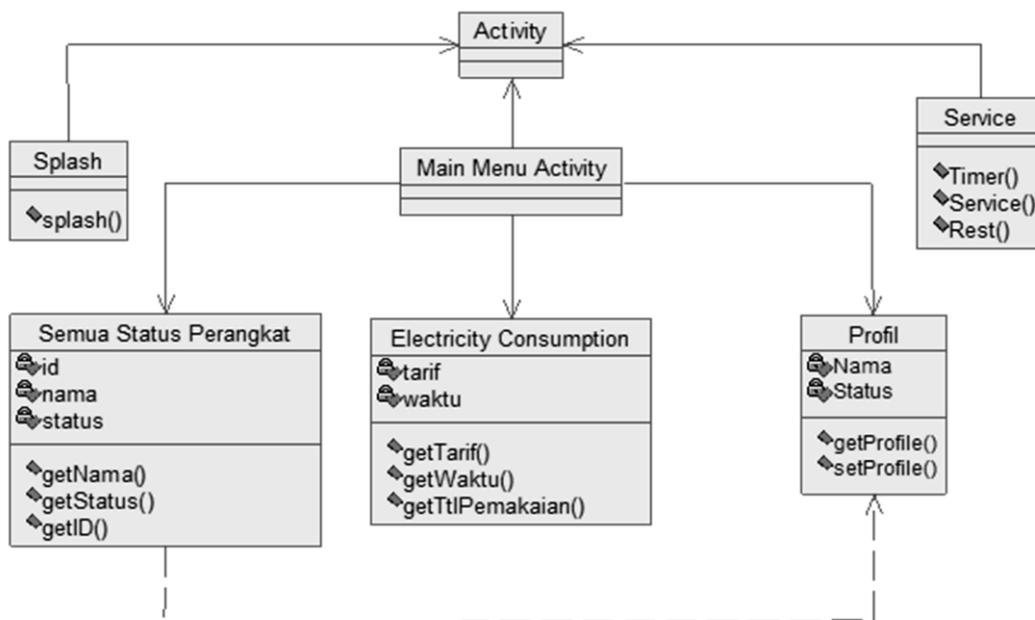
Menu “Konsumsi Listrik” digunakan untuk menampilkan informasi pemakaian listrik setiap peralatan rumah yang terhubung ke *home-gateway*. Di aplikasi tersebut terdapat *service* yang berjalan secara *background* untuk melakukan pengecekan setiap detik untuk peralatan apa saja yang dalam *state: on*, kemudian setiap jamnya akan dikalkulasi dengan tarif listrik per jam untuk diketahui tarif listrik sesuai pemakaian.

Menu “Semua Status Perangkat” digunakan untuk menampilkan status setiap perangkat yang terhubung ke *home-gateway*. Pengguna dapat mengubah *state* setiap perangkat dengan menekan tombol “ubah status *device*”, ini terjadi *realtime*, dan akan langsung terlihat informasinya ketika *server* sudah memberikan *response*.

Menu *Profile* digunakan untuk mengaktifkan satu dari 4 *profile*, yaitu *profile Energy Saving* bertujuan mengubah semua kondisi peralatan rumah dalam mode *energy saving*, *profile cool* bertujuan mengubah semua kondisi peralatan rumah yang berparameter suhu dalam mode mengubah suhunya menjadi dingin, *profile warm* bertujuan mengubah semua kondisi peralatan rumah yang berparameter suhu dalam mode *warm*.

### 5.2.2 Class Diagram

*Class diagram* dari prototipe aplikasi *remote control domotic* berbasis Android dengan menggunakan Android SDK dan *web service* RESTful dapat dilihat pada Gambar 3.

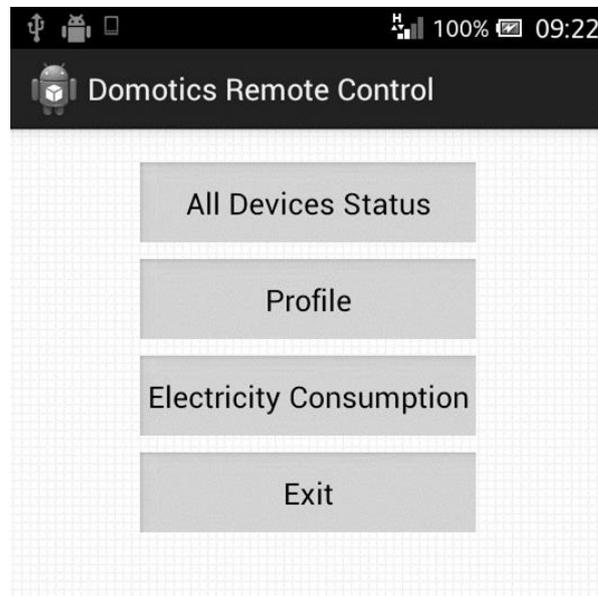


**Gambar 3** Class Diagram Remote Control Domotic

Pada saat aplikasi prototipe *remote control Domotic* ini dijalankan, halaman *Splash* akan tampil ke dalam *Activity*, kemudian masuk ke menu utama (*main menu*) *activity* yang berisi 3 menu: Semua Status Perangkat, *Electricity Consumption*, dan Profil. Selain itu, terdapat *background activity* yang berjalan untuk mengecek waktu dan melakukan *sync* ke *server* melalui RESTful *web service*.

### 5.3 Implementasi Perangkat Lunak

Perangkat lunak yang dibangun memiliki tiga menu utama, yaitu *All Devices Status*, *Profile*, *Electricity Consumption* dan *Exit*. Tampilan menu utama terlihat pada Gambar 4.



Gambar 4 Menu utama

### 5.4 Pengujian

Pada pengujian ini diharapkan dapat menemukan kesimpulan apakah aplikasi prototipe *remote control Domotic* ini *feasible* untuk diterapkan di dunia nyata. Secara rinci, tujuan pengujian aplikasi prototipe ini antara lain:

1. Mengukur fungsionalitas dari aplikasi prototipe. Pengujian ini mengacu pada analisis kebutuhan sistem yang telah dijabarkan sebelumnya.
2. Menganalisis interoperabilitas *web service RESTful* yang dimanfaatkan sebagai penyedia komunikasi antara *home gateway* dengan aplikasi *domotic* berbasis *mobile Android*.
3. Mensimulasikan kontrol jarak-jauh *Domotic* melalui aplikasi *mobile Android* terhadap perangkat elektronik di rumah yang terhubung dengan *home gateway*.

#### 5.4.1 Hasil Pengujian

Hasil pengujian fungsional dan pengujian performansi dapat dilihat pada Tabel 1 dan Tabel 2. Dari hasil pengujian nampak terlihat bahwa secara fungsional dan performansi *web service RESTful* mampu berjalan baik dan cepat. Namun, hal ini juga dipengaruhi oleh kualitas jaringan internet yang terpasang di *smartphone Android*.

**Tabel 1** Hasil pengujian fungsional

Pengujian	Kriteria Keberhasilan	Status
Mengubah status <i>device</i>	<i>Device</i> yang dipilih dapat mengubah <i>state</i> sesuai masukan pengguna.	Berhasil
Menghitung konsumsi Listrik <i>device</i>	Keterangan tagihan listrik terlihat di halaman " <i>Electricity Consumption</i> " sesuai pemakaian.	Berhasil
Mengubah status semua <i>device</i> secara bersamaan melalui menu <i>profile</i>	Semua <i>device</i> dapat mengubah <i>state</i> sesuai masukan pengguna	Berhasil

**Tabel 2** Hasil pengujian *web service* RESTful

<i>Request</i>	<i>Method</i>	<i>Time</i>
Cek status semua <i>device</i> [URL: http://.../rest/devices]	HTTP: GET	Pengujian ke: 1. 529 ms 2. 584 ms 3. 370 ms
Cek status salah satu <i>device</i> [URL: http://.../rest/devices/(:num)]	HTTP: GET	Pengujian ke: 1. 2366 ms 2. 2305 ms 3. 2757 ms
Mengubah status salah satu <i>device</i> [URL: .../rest/device/lampu/(:num) ]	HTTP: POST	Pengujian ke: 1. 568 ms 2. 493 ms 3. 526 ms

## 6. Penutup

Prototipe aplikasi *remote-control Domotic* yang dibuat belum memenuhi spesifikasi untuk diterapkan di lingkungan sesungguhnya. Hal ini disebabkan karena belum diimplementasikannya prototipe ini untuk dapat berkomunikasi dengan *home-gateway*. Dalam hal komunikasi data ke *server* masih diperlukan penelitian lebih lanjut mengenai perancangan, implementasi sampai dengan pengujian performansi *response* dari *home-gateway* ke *remote control*. Di samping itu, perlu juga dipertimbangkan untuk alternatif koneksi selain HTTP untuk tujuan penghematan biaya, misalnya dengan konektivitas *bluetooth* atau WiFi apabila *remote-control* pada jarak terjangkau dari peralatan di rumah yang terhubung *home-gateway*.

## Daftar Pustaka

- Aiello, M. & Dustdar, S., 2008. Are Our Homes Ready for Services? A Domotic Infrastructure Based On The Web Service Stack. *Pervasive and Mobile Computing*, 4(4), pp. 506-525.
- Bolzani, C. A. M., Montagnoli, C. & Netto, M. L., 2006. Domotics Over IEEE 802.15.4 – A Spread Spectrum Home Automation Application. Proceeding. *The 2006 IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications*, pp. 396-400.
- Breitman, K. K. K., Casanova, M. A. & Truszkowski, W., 2007). *Semantic Web: Concepts, Technologies and Applications*. London: Springer.
- Caytiles, R. D. & Park, B., 2012. Mobile IP-Based Architecture for Smart Homes. *International Journal of Smart Home*, 6(1), pp. 29-36.
- Delgado, A. R., Picking, R. & Grout, V., 2006. Remote-Controlled Home Automation Systems with Different Network Technologies. Proceeding. *The 6th International Network Conference (INC 2006)*, University of Plymouth, pp. 357-366.
- Gonzalez, V. M., Mateos, F., Lopez, A. M., Enguita, J. M., Garcia, M. & Olaiz, R., 2001. Visir, A Simulation Software for Domotics Installations to Improve Laboratory Training. Proceeding. *31st Annual IEEE Frontiers in Education Conference*, 3, pp. F4C-6.
- Greaves, D., 2002. Control Software for Home Automation, Design Aspects and Position Paper. Proceeding. *The 22nd International Conference on Distributed Computing System Workshop (ICDCSW '02)*, pp. 757-764.
- Meier, R., 2009. *Professional Android Application Development*. Canada: Wiley Publishing, Inc.
- Pautasso, C., Zimmermann, O. & Leymann, F., 2008. RESTful Web Services vs. “Big” Web Services: Making the Right Architectural Decision. Proceeding. *The 17th International Conference on World Wide Web*, pp. 805-814.
- Robles, R. J. & Kim, T. H., 2010. A Review on Security in Smart-Home Development. *International Journal of Advanced Science and Technology*, 15, pp. 13-22.
- Sandu, F. & Iolu, D., 2007. Embedded Web-Servers for Remote Control in Domotics. *ACTA TECHNICA NAPOCENSIS*, 48(3), pp. 41-44.