

APLIKASI MOBILE ROBOT PEMANTAU RUANGAN MELALUI INTERNET DENGAN MOBILE PHONE BERBASIS ANDROID

(Application of Room Monitoring Mobile Robot Through Internet with Android Based Mobile Phone)

Wellina Leonardy*, Lukas**

**Fakultas Teknik Jurusan Teknik Elektro
Universitas Katolik Indonesia Atma Jaya
Jl. Jendral Sudirman 51, Jakarta Selatan 12930
*wellinaleonardy@yahoo.com, **lukas@atmajaya.ac.id**

Abstrak

Sistem pemantau ruangan pada umumnya menggunakan sebuah kamera yang diposisikan secara statis dan dihubungkan dengan layar untuk menampilkan hasil tangkapan kamera. Sistem tersebut memiliki keterbatasan dalam hal sudut penangkapan kamera dan jarak pemantauan. Sistem pemantau ruangan yang memiliki sudut penangkapan kamera dan jarak pemantauan yang dinamis dapat menjadi solusi dalam masalah tersebut. Untuk merealisasikan sistem tersebut, dibutuhkan sebuah *mobile phone* yang memiliki fitur Wi-Fi dan sebuah perangkat (misalnya *mobile phone* atau komputer) yang dapat terhubung ke internet. *Mobile phone* digunakan sebagai kamera dan berinteraksi dengan *mobile robot*. Sedangkan perangkat lain digunakan untuk menampilkan gambar hasil tangkapan kamera dan berinteraksi langsung dengan pengguna. Dari hasil pengujian diperoleh bahwa dengan adanya koneksi internet Wi-Fi, terjadi interaksi antara *mobile phone* dengan perangkat sehingga sistem pemantau ruangan jarak jauh dapat tercapai.

Kata Kunci: pemantau ruangan, *mobile robot*, *mobile phone*, internet

Abstract

Room monitoring system generally uses a static camera that is positioned and connected to a screen to display the camera catches. However, this system has its limitations in terms of catching the camera angle and distance monitoring. The room monitoring system with a dynamic camera angle and distance monitoring is expected to be a solution to the problem. Actualizing such a system takes a mobile phone that has Wi-Fi and a device such as a mobile phone or a computer connected to the internet. The mobile phone is used as a camera that interacts with a mobile robot. Other devices are used to display the video screen and interact directly with the user. The test results show that the Wi-Fi internet connection allows a mobile phone and a device to interact to achieve remote monitoring system.

Keywords: room monitoring, *mobile robot*, *mobile phone*, internet

Tanggal Terima Naskah : 18 September 2013
Tanggal Persetujuan Naskah : 04 Oktober 2013

1. PENDAHULUAN

1.1 Latar Belakang

Untuk memantau ruangan sering digunakan kamera yang dihubungkan dengan layar untuk menampilkan hasil tangkapan kamera. Pada sistem tersebut, koneksi antara kamera dan layar menggunakan kabel sehingga jarak pemantauan menjadi terbatas. Selain itu, kamera diposisikan secara statis pada sudut-sudut tertentu dengan tujuan agar dapat melihat seluruh isi ruangan. Namun demikian, sistem ini mempunyai kelemahan pada hasil tangkapan kamera yang terbatas.

Teknologi saat ini memungkinkan koneksi nirkabel dengan kecepatan tinggi sebagai media untuk menghubungkan kamera dengan layar dimana gambar hasil tangkapan kamera dapat dikirimkan secara *streaming* ke layar. Untuk memperluas sudut penangkapan kamera, dapat digunakan sebuah *mobile robot* untuk membawa kamera yang pergerakannya dapat dikendalikan.

Untuk merealisasikan sistem pemantau ruangan tersebut, dibutuhkan sebuah *mobile phone* yang memiliki fitur *Wireless Fidelity* (Wi-Fi) dan sebuah perangkat (misalnya *mobile phone* atau komputer) yang dapat terhubung ke internet. *Mobile phone* digunakan sebagai kamera dan berinteraksi dengan *mobile robot* sedangkan perangkat digunakan untuk menampilkan layar video dan berinteraksi langsung dengan pengguna.

1.2 Tujuan

Tujuan yang hendak dicapai adalah merancang dan membangun suatu perangkat lunak *mobile phone* yang dapat mengirim gambar ke perangkat lain maupun menerima data perintah dari perangkat lain melalui internet untuk diaplikasikan pada sistem pemantau ruangan.

1.3 Batasan Masalah

Pembatasan masalah untuk penelitian ini adalah:

- Koneksi internet menggunakan *Wireless Fidelity* (Wi-Fi)
- Diberikan tiga pilihan resolusi gambar, yaitu (176 x 144) piksel, (240 x 160) piksel, (320 x 240) piksel
- Perangkat lunak yang digunakan adalah *Eclipse Classic 3.6.2* yang sudah ter-*install Android Development Tools (ADT) Plugin*.

2. KONSEP DASAR

2.1 Mobile Robot

Mobile robot adalah jenis robot yang memiliki kemampuan bergerak bebas/berpindah secara otomatis atau dikendalikan oleh manusia. Dari segi manfaat, *mobile robot* dapat membantu manusia dalam melakukan otomasi dalam transportasi, *platform* bergerak untuk robot industri, eksplorasi tanpa pengemudi, dan masih banyak lagi manfaat lainnya.

2.2 Android

Android merupakan sebuah sistem operasi perangkat *mobile* berbasis *Linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Keunggulan utama *Android* adalah gratis dan *open source* bagi pengembang (*programmer*) untuk pembuatan aplikasi. Setiap *release*-nya diberi kode nama berdasarkan nama hidangan makanan [1].

Pada rancangan ini digunakan *Android* versi *Gingerbread* (2.3). Struktur aplikasi *Android* ditulis dalam bahasa pemrograman *Java*. Kode *Java* dikompilasi bersama dengan *file resource* yang dibutuhkan oleh aplikasi. Prosesnya dikemas oleh *tools* yang dinamakan “*apt tools*” ke dalam paket *Android*, sehingga menghasilkan *file* dengan ekstensi *apk*. *File* *apk* ini yang disebut dengan aplikasi dan nantinya dapat dijalankan pada peralatan *mobile*.

2.3 Bahasa Pemrograman Web

Bahasa pemrograman *web* merupakan suatu teknik instruksi standar untuk memerintah komputer melalui *website*. Dalam rancangan berikut merupakan bahasa pemrograman yang berbasis *web*.

2.3.1 HyperText Markup Language

HyperText Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi di dalam sebuah *browser* internet [2].

2.3.2 Hypertext Preprocessor

Hypertext Preprocessor (PHP) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML [3]. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada saat itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web*. PHP banyak digunakan untuk membuat situs yang dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

2.3.3 eXtensible Markup Language

eXtensible Markup Language (XML) adalah bahasa *markup* yang dirancang khusus untuk penyampaian informasi melalui *World Wide Web* (WWW), yang merupakan pengembangan dari HTML. XML mampu mendefinisikan sebuah dokumen berstruktur hirarki dengan elemen-elemen yang tersarang (*nested*) sepenuhnya dalam elemen lain dan dengan sebuah elemen tingkat teratas yang dikenal sebagai elemen *root*.

Untuk memproses dokumen XML dibutuhkan XML *parser*, yakni sebuah program yang mampu menguraikan dokumen XML menjadi elemen individual. Salah satu metode XML *parser* adalah *Document Object Model* (DOM). DOM merupakan suatu metode untuk mengakses dan memodifikasi dokumen yang bercabang-cabang (*tree base*). Beberapa prosedur untuk melakukan manipulasi terhadap struktur pohon DOM meliputi penambahan (*adding*), penghapusan (*deleting*), dan pemutakhiran (*updating*) [4].

2.3.4 JavaScript

Javascript adalah bahasa *script* yang ditempelkan pada kode HTML dan diproses di sisi klien. Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi semakin luas. Secara fungsional, *JavaScript* digunakan untuk menyediakan akses *script* pada objek yang ditanamkan (*embedded*) [5].

2.3.5 Cascading Style Sheets

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. CSS

biasanya digunakan untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan halaman *web* [6].

2.4 Web Server

Web server adalah perangkat lunak *server* yang menjadi tulang punggung dari *World Wide Web* (WWW). *Web server* menunggu permintaan dari *client* yang menggunakan *browser*, seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla*, atau program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memroses permintaan itu, kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser* [7]. Data ini mempunyai format standar yang disebut format *Standard General Markup Language* (SGML). Data yang berupa *hypertext* ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* tersebut. Untuk berkomunikasi dengan *client* (*web browser*), *web server* mempunyai protokol sendiri, yaitu *HyperText Transfer Protocol* (HTTP).

2.5 Wireless Fidelity (Wi-Fi)

Wireless Fidelity (Wi-Fi) merupakan teknologi jaringan lokal nirkabel (*Wireless Local Area Networks* - WLAN) yang mempergunakan teknologi radio sehingga pemakainya dapat mengirim data dengan cepat dan aman [8]. Awalnya Wi-Fi ditujukan untuk penggunaan perangkat nirkabel dan Jaringan Area Lokal (LAN), namun saat ini lebih banyak digunakan untuk mengakses internet. Hal ini memungkinkan seseorang dengan kartu nirkabel (*wireless card*) atau *Personal Digital Assistant* (PDA) terhubung ke internet menggunakan titik akses (atau dikenal dengan *hotspot*) terdekat.

3. PERANCANGAN DAN REALISASI SISTEM

3.1 Arsitektur Sistem Pemantauan Ruangan melalui Internet

Sistem pemantau ruangan melalui internet (*Internet Monitoring System/iMon*) terdiri dari perangkat keras (*Eye Robot/iBot*) dan perangkat lunak. *iBot* terdiri dari enam modul dan dua mode operasi. Enam modul pada *iBot* adalah modul sensor, modul mikrokontroler, modul pengendali motor, modul regulator tegangan, modul tampilan LED, dan modul *bluetooth*. Mode operasi pada *iBot* adalah *mode* mandiri dan *mode* manual. Pada *mode* mandiri, *iBot* memiliki kemampuan untuk bergerak patroli mengikuti garis berdasarkan jalur yang telah dibuat (*line follower*). Pada *mode* manual, pengguna dapat mengontrol pergerakan *iBot*, yaitu pengguna dapat menghentikan pergerakan *iBot* dan mengarahkan *iBot* menuju ke suatu ruangan. Rancangan perangkat lunak terdiri dari aplikasi *web* (program *client*) dan aplikasi *Android* berbasis *Java* (program *iBot*). Diagram blok *iMon* dapat dilihat pada Gambar 1.

3.2 Spesifikasi *iBot* dan Perangkat untuk Pengguna

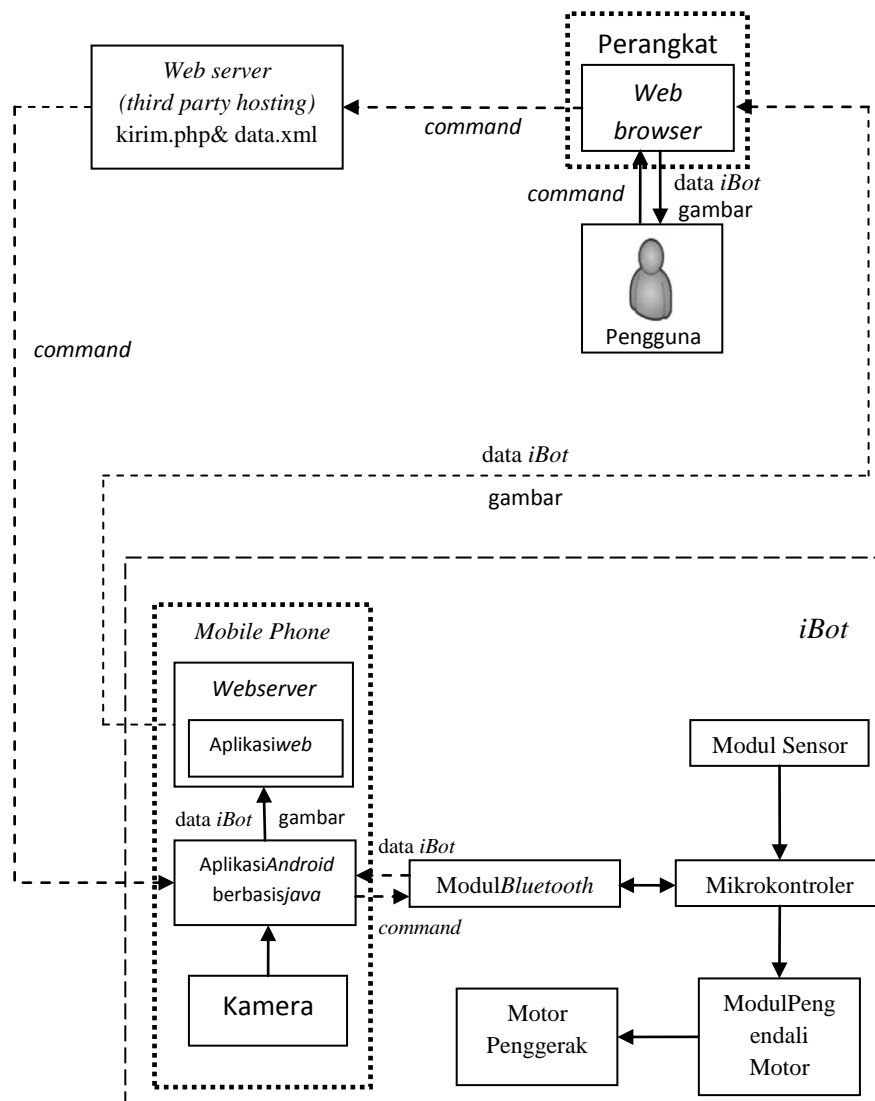
iBot menggunakan sebuah *mobile phone* yang berfungsi sebagai *server* yang akan ditempatkan pada *robot*. Spesifikasi *mobile phone* untuk *server*, yaitu:

- menggunakan sistem operasi *Android 2.3* (*Gingerbread*)
- mendukung jaringan Wi-Fi dan *bluetooth*
- memiliki kamera dengan resolusi minimal 0.3 Megapiksel.

iBot memiliki spesifikasi sebagai berikut:

- 1) Satu buah *gearbox* tamiya yang terhubung dengan dua buah roda
- 2) Dua buah roda *caster*

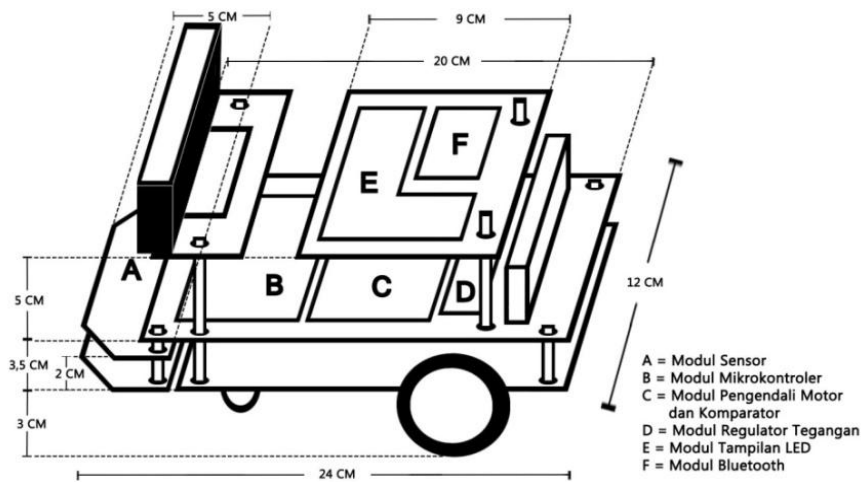
- 3) Dua buah motor DC untuk penggerak *mobile robot* dengan metode *differential drive*
- 4) Satu buah motor DC untuk mengatur arah tangkapan kamera *mobile phone*
- 5) Dua buah motor *driver IC L293D*
- 6) Modul sensor, yang terdiri dari lima buah LED infra merah, lima buah fototransistor, lima buah LED untuk *display*, dan dua buah IC komparator LM339.
- 7) Pemroses menggunakan mikrokontroler AT89S51
- 8) Antarmuka mikrokontroler dan *mobile phone* menggunakan modul *Bluetooth* jenis MB-C04
- 9) Aki kering 10 Volt 850 mAh dan rangkaian regulator tegangan 5 Volt sebagai catu daya mikrokontroler, modul *bluetooth*, dan modul sensor
- 10) Baterai NiCD 7,2 Volt 1500 mAh sebagai catu daya modul pengendali motor.



Gambar 1. Diagram blok iMon

Program *client* dapat diakses pengguna melalui berbagai perangkat seperti *mobile phone*, komputer, *laptop*, *Personal Digital Assistant (PDA)*, dan sebagainya. Spesifikasi perangkat untuk pengguna, yaitu:

- Memiliki fitur koneksi ke internet
- Memiliki *browser*, seperti *Google Chrome*, *Mozilla*, *Opera Mini*, atau program *browser* lainnya.

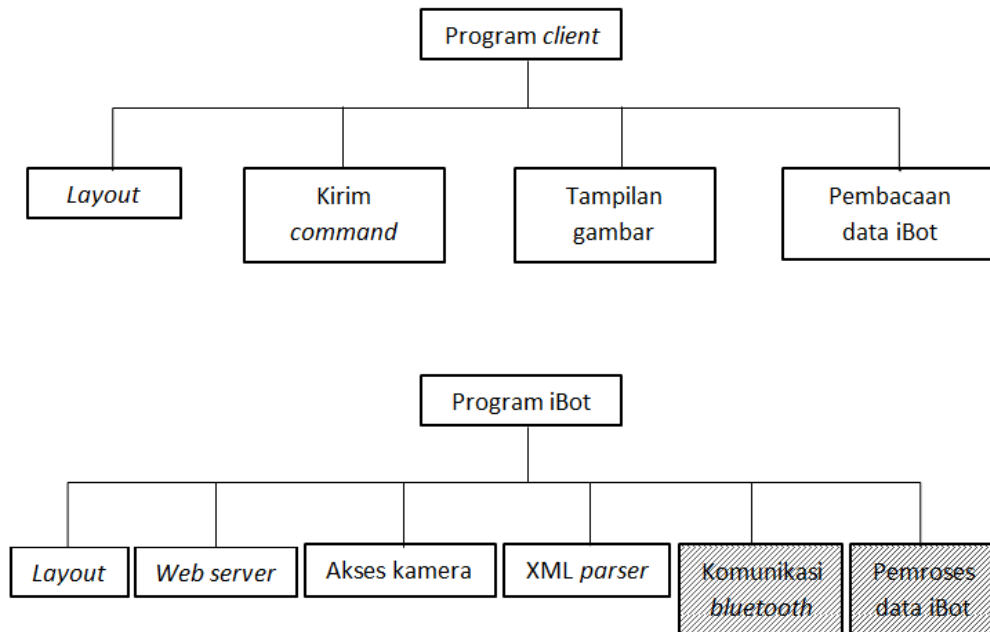


Gambar 2. Sketsa dan dimensi *iBot*

3.3 Perancangan dan Realisasi *iMon*

Pada penelitian ini dibuat rancangan dan realisasi *iMon* yang terdiri dari perancangan dan realisasi *iBot* serta pembuatan perangkat lunak *iMon*. *iBot* terdiri dari modul regulator tegangan, modul mikrokontroler, modul sensor, modul pengendali motor, modul tampilan LED, dan modul *bluetooth* sedangkan perangkat lunak *iMon* terdiri dari program *client* dan program *iBot*. Pembagian modul perangkat lunak *iMon* dapat dilihat pada Gambar 3.

Program *client* terdiri dari *layout*, kirim *command*, tampilan gambar, dan pembacaan data *iBot* sedangkan program *iBot* terdiri dari *layout*, *web server*, akses kamera, XML *parser*, komunikasi *bluetooth*, dan pemroses data *iBot*.



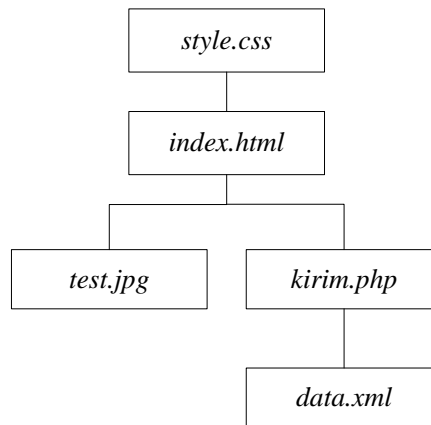
Gambar 3. Pembagian modul perangkat lunak *iMon*

3.3.1 Perancangan dan Realisasi Program Client

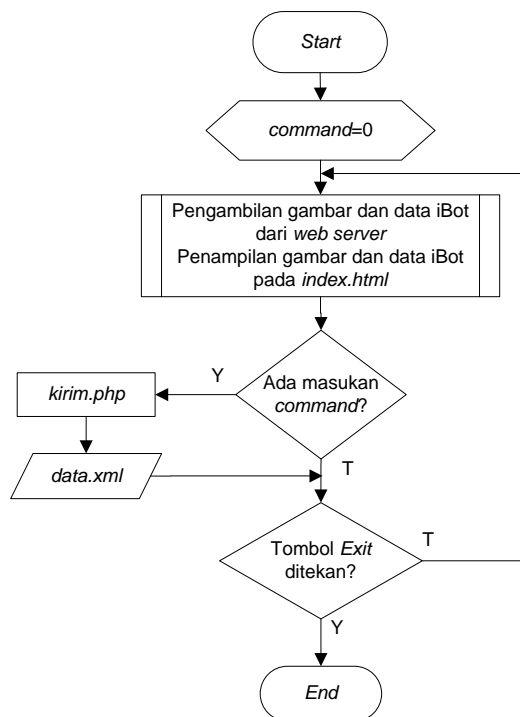
Program *client* diletakkan pada *web server* dan dapat diakses oleh pengguna melalui sebuah *browser*. Tugas utama dari program *client* adalah:

- 1) Mengambil gambar dari *web server*, kemudian menampilkannya melalui *browser* pada layar perangkat pengguna.
- 2) Membaca data *iBot* dalam file *data.txt* pada *web server*, kemudian menampilkan hasilnya melalui *browser* pada layar perangkat pengguna.
- 3) Mengirimkan *command* ke dalam file XML yang berada di *third party hosting web server* (<http://www.tokofunia.com>).

Program *client* direalisasikan dengan pembuatan *style.css*, *index.html*, *test.jpg*, *kirim.php*, dan *data.xml*. Herarki file dari program *client* dapat dilihat pada Gambar 4 dan diagram alir program *client* dapat dilihat pada Gambar 5.



Gambar 4. Herarki file program *client*



Gambar 5. Diagram alir program *client*

Berdasarkan fungsinya, program *client* terbagi menjadi empat modul, yaitu modul *layout*, modul kirim *command*, modul tampilan gambar, dan modul pembacaan data *iBot*. Berikut adalah deskripsi masing-masing modul.

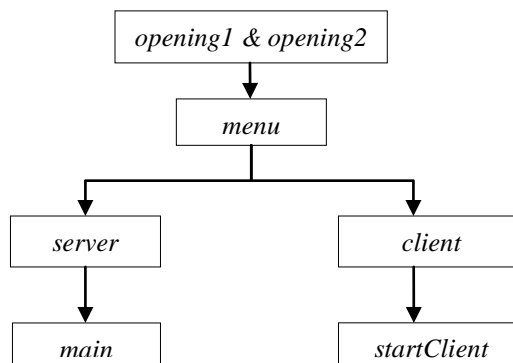
- 1) Modul *Layout*
 Modul *layout* merupakan modul pengatur tampilan program *client* dengan menggunakan *CSS* dan fungsi *table* pada *HTML*. Pada modul *layout* ditampilkan gambar yang diambil dari *web server*. Selain itu, juga ditampilkan pembacaan *Speed* dan *Sensor* pada *iBot*. Pada modul ini terdapat beberapa tombol, yaitu *Power (on/off)*, *Mode Operasi (manual/mandiri)*, *Speed*, *Camera Motor*, dan arah (atas, bawah, kiri, kanan). Apabila pengguna menekan salah satu tombol, maka *file kirim.php* akan mengubah nilai yang terdapat pada *data.xml*.
- 2) Modul Tampilan Gambar
 Pada *index.html* terdapat *tag image* yang berfungsi untuk menampilkan *test.jpg* yang diambil dari *web server*.
- 3) Modul Kirim *Command*
 Pada *index.html* terdapat tombol-tombol *command* untuk memberi perintah pada *iBot*. Apabila tombol *command* ditekan, maka dengan *JavaScript file kirim.php* di *third party hosting* akan diakses untuk mengubah isi *tag <c> data.xml* di *third party hosting*.
- 4) Modul Pembacaan Data *iBot*
 Dengan menggunakan metode *XML DOM parser*, file *data.txt* pada *web server* akan dibaca dan hasilnya akan ditampilkan pada *layout* berupa gambar LED. Dengan fungsi *myTimer1* pada *JavaScript*, pembacaan *file data.txt* akan dilakukan setiap 0,5 detik sehingga gambar LED pada *layout* juga akan berubah setiap 0,5 detik.

3.3.2 Perancangan dan Realisasi Program *iBot*

Program *iBot* di-install pada *mobile phone* yang berfungsi sebagai *server*. Tugas utama dari program *iBot* adalah:

- 1) Mengaktifkan *web server* dan mengarahkan *folder "/sdcard/iBot/"* sebagai *folder resource*.
- 2) Mengambil gambar hasil tangkapan kamera secara periodik, yang kemudian disimpan ke dalam *folder "/sdcard/iBot/"* pada *mobile phone*.
- 3) Membaca *file XML* yang berada di *third party hosting web server*, kemudian diterjemahkan menjadi data *command* dan dikirimkan ke modul *bluetooth*.

Perancangan program *iBot* dimulai dengan menambahkan *permission* pada *Android Manifest.xml*, yaitu *camera*, *bluetooth*, *write external data storage*, dan *internet*. Program dilanjutkan dengan membuat *activity* secara hirarki yang dapat dilihat pada Gambar 6 dan diagram alir program *iBot* dapat dilihat pada Gambar 7.



Gambar 6. Hirarki *activity* program *iBot*

Berdasarkan fungsinya, program *iBot* terbagi menjadi empat modul, yaitu modul *layout*, modul *web server*, modul akses kamera, modul *XML parser*.

1) Modul *Layout*

Modul *layout* merupakan modul pengatur tampilan setiap *activity* pada program *iBot*. Setiap *activity* diatur tampilannya melalui *file XML* yang berada di dalam *folder layout*.

2) Modul *Web Server*

Modul *web server* merupakan modul program *iBot* yang berhubungan dengan *web server*. Tugas utama modul *web server* meliputi:

- Menetapkan *folder "/sdcard/iBot"* sebagai *folder resource web server*
- Menyiapkan objek untuk menangani *client*, dalam program ditetapkan *server* dapat melayani maksimal tiga *client* dalam satu waktu. Setiap objek berjalan pada masing-masing *thread*-nya
- Membuat objek *server*
- Membuka koneksi *server* untuk menerima *request* dari *client*
- Mengizinkan *client* untuk melakukan *streaming* gambar atau mengakses *file* pada *folder resource web server*

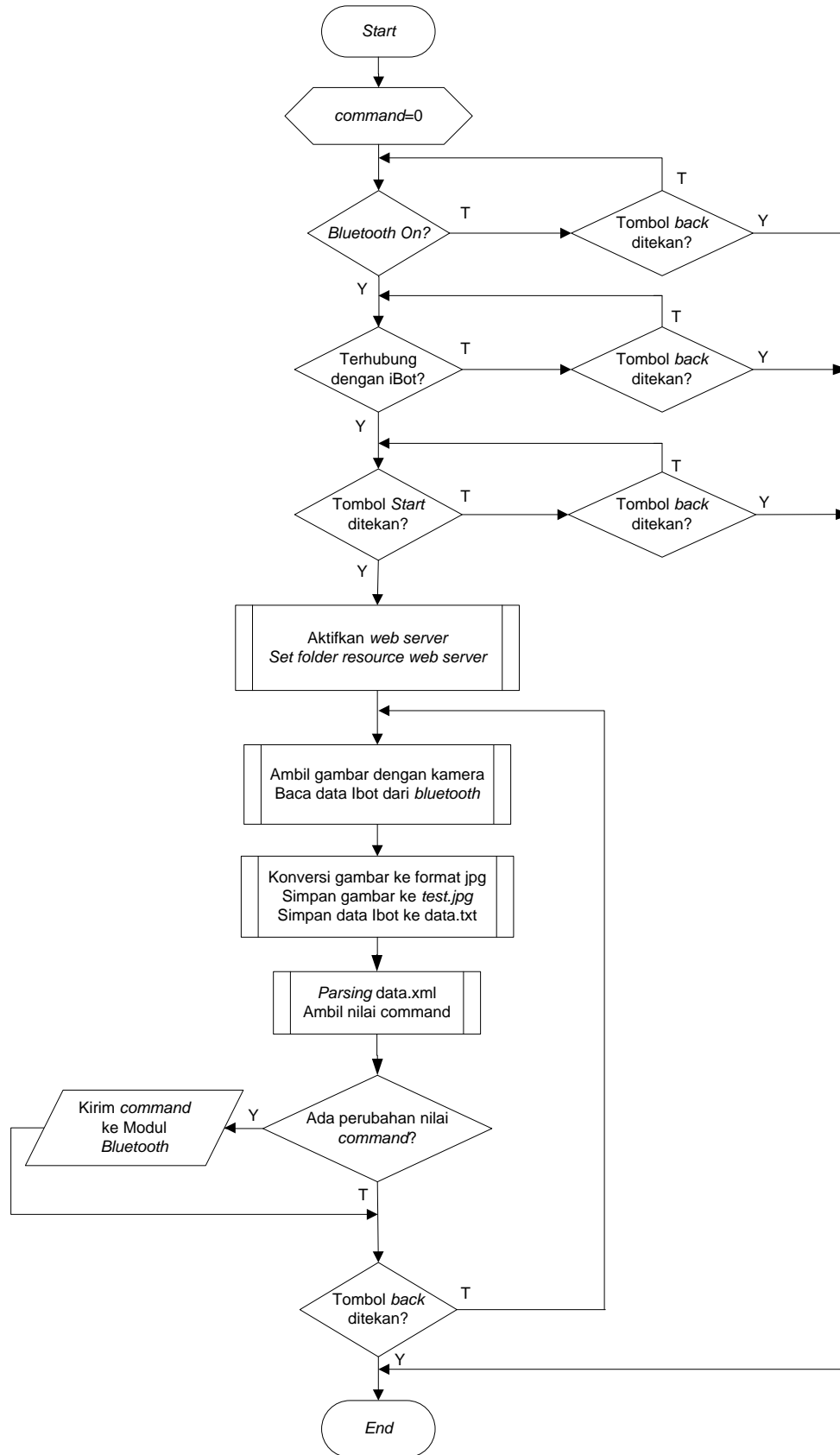
3) Modul Akses Kamera

Modul akses kamera merupakan modul program *iBot* yang berhubungan dengan kamera. Tugas utama modul akses kamera meliputi:

- Mengakses kamera *mobile phone* dengan cara mendapatkan *instance* dari kelas *Camera*
- Mendapatkan parameter kamera dengan memanggil fungsi *getParameters()*
- Membuat objek pada kelas Kamera yang mewarisi kelas *SurfaceView* sebagai wadah untuk menampilkan gambar tangkapan kamera
- Masukkan *cameraFrame* pada *layout activity* yang aktif sehingga gambar tangkapan kamera dapat dilihat di layar
- Menggunakan parameter kamera untuk menentukan resolusi *Surface*
- Memanggil fungsi *startPreview()* untuk menampilkan gambar tangkapan kamera pada layar dan meng-*update*-nya terus-menerus
- Menangkap gambar dan menghasilkan *image* dalam bentuk *byte array*
- Melakukan kompresi pada *image* yang masih dalam bentuk *byte array* menjadi *jpg*
- Menyimpan *image* ke dalam *web server*

4) Modul *XML parser*

Modul *XML parser* merupakan modul program *iBot* untuk mem-*parsing file data.xml* yang terdapat pada *third party hosting web server*. Dari hasil *parsing* akan didapatkan data *command* yang kemudian akan dikirim melalui *bluetooth* untuk pengendalian *iBot*.



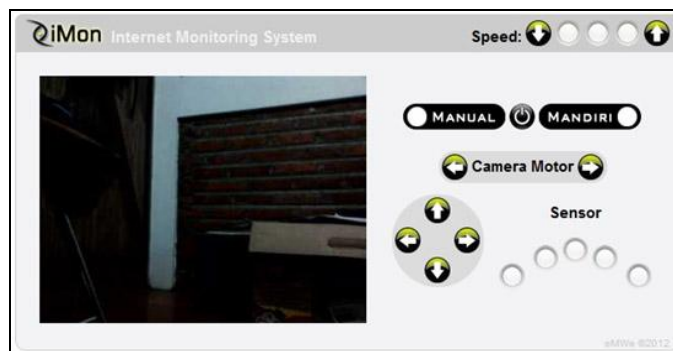
Gambar 7. Diagram alir program *iBot*

4. PENGUJIAN SISTEM

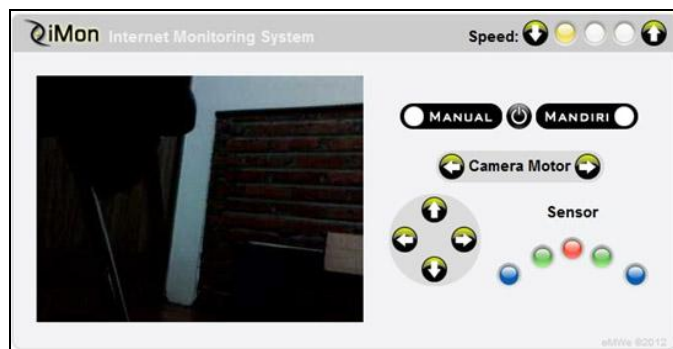
4.1 Pengujian Program *Client*

Tujuan dari pengujian program *client* adalah untuk mengetahui apakah program *client* sudah dapat beroperasi dengan baik. Pengujian dilakukan dengan menggunakan komputer dan *mobile phone* untuk mengakses program *client*. Komputer memiliki spesifikasi prosesor *Intel Pentium Core 2 Duo E7500 2,93GHz* dan memori *DDR2 2048 MB*. Untuk *mobile phone* digunakan *Samsung Galaxy Gio S5660* dengan spesifikasi prosesor *Qualcomm QCT MSM7227-1 Turbo 800 MHz* dengan sistem operasi *Android 2.3 (Gingerbread)* dan menggunakan browser *Dolphin HD*.

Pada awalnya, program *web server* diaktifkan dengan cara menyalakan program *iBot* yang sudah ter-*install* pada *mobile phone* (*web server* langsung pada *mobile phone*). Apabila *web server* sudah aktif, maka program *client* siap dijalankan melalui *browser*. Tampilan program *client* sebelum *iBot* diaktifkan dapat dilihat pada Gambar 8 dan tampilan program *client* setelah *iBot* diaktifkan dapat dilihat pada Gambar 9.



Gambar 8. Tampilan program *client* sebelum *iBot* diaktifkan



Gambar 9. Tampilan program *client* setelah *iBot* diaktifkan

Pada pengujian pertama dilakukan lima kali percobaan untuk mengambil gambar dari *web server*. Dari pengujian pengambilan gambar dari *web server*, diperoleh hasil seperti tampak pada Tabel 1.

Pengujian kedua dilakukan dengan lima kali percobaan untuk mengirim *command* ke program *iBot*. Pada tahap ini akan diuji lamanya waktu yang dibutuhkan untuk mengirimkan *command* ke *data.xml* sampai *data.xml* dibaca oleh program *iBot*. Hasil pengujian tersebut dapat dibaca pada Tabel 2.

Tabel 1. Waktu pengambilan gambar dari *web server*

| Percobaan ke- | Waktu pengambilan gambar (detik) | | | |
|------------------|----------------------------------|-------------------------------|-------------------------------------|----------------------------|
| | Komputer (Fastnet 2 Mbps) | | Samsung Gio (Browser Dolphin HD) | |
| | Browser Google Chrome | Browser Mozilla Firefox | Menggunakan Wi-Fi | Menggunakan Jaringan 3G |
| 1 | 0,94 | 1,11 | 1,81 | 4,52 |
| 2 | 1,10 | 0,94 | 1,54 | 9,33 |
| 3 | 1,02 | 0,95 | 1,75 | 4,38 |
| 4 | 0,94 | 1,12 | 1,64 | 7,55 |
| 5 | 0,97 | 1,01 | 1,75 | 3,98 |
| Rata-rata | 0,99 | 1,02 | 1,69 | 5,95 |

Tabel 2. Waktu pengiriman *command* ke program *iBot*

| Percobaan ke- | Waktu pengiriman <i>command</i> (detik) | | | |
|------------------|---|-------------------------------|-------------------------------------|----------------------------|
| | Komputer (Fastnet 2 Mbps) | | Samsung Gio (Browser Dolphin HD) | |
| | Browser Google Chrome | Browser Mozilla Firefox | Menggunakan Wi-Fi | Menggunakan Jaringan 3G |
| 1 | 1,22 | 1,56 | 1,96 | 5,54 |
| 2 | 1,18 | 1,65 | 1,48 | 6,78 |
| 3 | 1,15 | 1,45 | 1,44 | 9,26 |
| 4 | 1,24 | 1,59 | 1,36 | 4,45 |
| 5 | 1,19 | 1,33 | 1,61 | 9,56 |
| Rata-rata | 1,19 | 1,51 | 1,57 | 7,12 |

Berdasarkan Tabel 1, diperoleh nilai rata-rata waktu pengambilan gambar dari *web server* pada komputer antara *browser Google Chrome* dengan *Mozilla Firefox* memiliki nilai yang hampir sama (sekitar 1 detik). Pada *Samsung Gio* membutuhkan waktu pengambilan gambar yang lebih lama, yaitu sebesar 1,69 detik dengan menggunakan Wi-Fi dan 5,95 detik dengan menggunakan jaringan 3G. Pada pengujian dengan menggunakan jaringan 3G diperoleh nilai yang tidak stabil.

Berdasarkan Tabel 2, diperoleh nilai rata-rata waktu pengiriman *command* ke program *iBot* pada komputer melalui *browser Google Chrome* sebesar 1,19 detik, sedangkan dengan *Mozilla Firefox* sebesar 1,51 detik. Pada *Samsung Gio* membutuhkan waktu pengiriman *command* yang lebih lama, yaitu sebesar 1,57 detik dengan menggunakan Wi-Fi dan 7,12 detik dengan menggunakan jaringan 3G. Pada pengujian dengan menggunakan jaringan 3G diperoleh nilai yang tidak stabil.

Dari pengujian diperoleh hasil bahwa *delay* yang terjadi tidak dipengaruhi oleh jarak antara pengguna dan *iBot*, melainkan dipengaruhi oleh *bandwidth* internet.

4.2 Pengujian Program *iBot*

Tujuan dari pengujian program *iBot* adalah untuk mengetahui apakah program *iBot* yang akan digabungkan dengan *mobile robot* sudah dapat beroperasi dengan baik. Pengujian dilakukan pada *mobile phone Samsung Galaxy Gio S5660* dengan spesifikasi prosesor *Qualcomm QCT MSM7227-1 Turbo 800 MHz* dengan sistem operasi *Android 2.3 (Gingerbread)*. Pengujian dilakukan dengan menggunakan koneksi Wi-Fi dengan

kecepatan 2 Mbps. Pengujian tersebut meliputi pengujian pengambilan gambar secara periodik dari kamera *mobile phone* dan pengujian penghitungan lama waktu pembacaan *command* dari *data.xml* oleh program *iBot*. Tampilan halaman utama program *iBot* dapat dilihat pada Gambar 10.



Gambar 10. Tampilan halaman utama program *iBot*

Pada pengujian pertama dilakukan percobaan untuk mengambil gambar secara periodik dari kamera *mobile phone*. Dari hasil percobaan diperoleh waktu untuk pengambilan gambar dari kamera *mobile phone* berjalan secara *streaming*. Pada pengujian kedua dilakukan percobaan untuk menghitung lama waktu pembacaan *command* pada *data.xml* oleh program *iBot*. Dari pengujian, diperoleh hasil seperti pada Tabel 3.

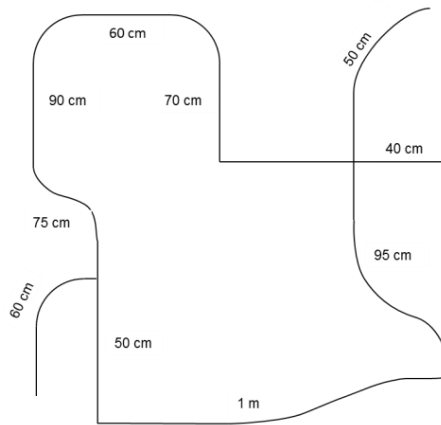
Tabel 3. Waktu pembacaan *command* pada *data.xml* oleh program *iBot*

| Percobaan ke- | Waktu pengiriman dengan Wi-Fi (detik) |
|------------------|---------------------------------------|
| 1 | 0,75 |
| 2 | 0,74 |
| 3 | 0,75 |
| 4 | 0,75 |
| 5 | 0,74 |
| Rata-rata | 0,75 |

4.3 Pengujian *iMon*

Pengujian *iMon*, yaitu pengujian sistem secara keseluruhan bersama dengan peneliti lain dalam tim. Pengujian dilakukan baik untuk *mode* manual maupun *mode* mandiri. Pada pengujian *mode* manual, diukur berapa besar *delay* yang dibutuhkan untuk mengirimkan *command* ke *iBot*. Dalam hal ini *command* merupakan perintah untuk menggerakkan *iBot*.

Pengujian pada *mode* mandiri dilakukan dengan menghitung lama *iBot* berpatroli menelusuri jalur pada setiap tingkat kecepatan. Sketsa jalur *iBot* yang digunakan dalam pengujian dapat dilihat pada Gambar 11.



Gambar 11. Sketsa jalur *iBot*

Hasil pengujian waktu tempuh *iBot* menelusuri jalur dapat dilihat pada Tabel 3.

Tabel 3. Waktu tempuh *iBot* menelusuri jalur

| Putaran ke- | Waktu tempuh (detik) | | |
|------------------|---------------------------|---------------------------|---------------------------|
| | <i>Speed 1</i> PWM 50% | <i>Speed 2</i> PWM 60% | <i>Speed 3</i> PWM 70% |
| 1 | 95,99 | 70,17 | 55,15 |
| 2 | 100,27 | 75,49 | 60,10 |
| 3 | 102,45 | 77,87 | 54,35 |
| 4 | 97,29 | 78,01 | 63,44 |
| 5 | 105,67 | 76,24 | 58,35 |
| Rata-rata | 103,30 | 75,56 | 58,28 |

Berdasarkan Tabel 3 didapatkan bahwa *iBot* paling stabil bergerak pada kecepatan 2.

5. KESIMPULAN

Berdasarkan perancangan dan hasil pengujian program *client* dan program *iBot*, dapat disimpulkan bahwa:

- 1) Pengiriman/penerimaan data berupa gambar dan data perintah pengendalian *mobile robot* ke/dari perangkat lain dapat dilakukan secara mengalir (*streaming*) melalui internet.
- 2) Gambar hasil tangkapan kamera yang ditampilkan program *client* semakin kabur pada kecepatan *iBot* yang semakin tinggi.
- 3) *Browser* terbaik untuk program *client* dari segi pengambilan gambar dan pengiriman *command* adalah *Google Chrome*.

REFERENSI

- [1]. Mulyadi, ST. 2010. Membuat Aplikasi untuk Android. Yogyakarta: Multimedia Center Publishing.
- [2]. Gottleber, T. T. and Trainor, T. N. 2003. Even More Excellent: HTML with XML, XHTML, and JavaScript. New York: McGraw-Hill Companies.
- [3]. Maxfield, W. 2001. MYSQL & PHP from Scratch. Indianapolis: Que Publishing.

- [4]. McLaughlin, B. 2000. *Java and XML*. Sebastopol: O'Reilly & Associates.
- [5]. Flanagan, D. 1998. *JavaScript: The Definitive Guide*, 3rd ed. Sebastopol: O'Reilly & Associates.
- [6]. Schmitt, C. et al. 2008. *Professional CSS: Cascading Style Sheets for Web Design*, 2nd ed. Indianapolis: Wiley Publishing.
- [7]. Azmi, Fauzan. "*Definisi Web Server*" [Online] <http://www.stmikpontianak.ac.id/admin/upload/web-server.pdf> (diakses 9 Mei 2009).
- [8]. Davis, H. 2004. *Absolute Beginner's Guide to Wi-Fi Wireless Networking*. Indianapolis: Que Publishing.