

RANCANG APLIKASI ANDROID PENGENDALI MOBIL DAN KAMERA MENGUNAKAN APP INVENTOR

Sandy S. Prayogo

Jurusan Teknik Elektro, Fakultas Teknologi Industri
Universitas Gunadarma
Depok, Indonesia
sunariver@student.gunadarma.ac.id

Trini Saptariani

Jurusan Sistem Komputer, Fakultas Ilmu Komputer &
Teknologi Informasi
Universitas Gunadarma
Depok, Indonesia
trini@staff.gunadarma.ac.id

Nur Sultan Salahuddin

Jurusan Sistem Komputer, Fakultas Ilmu Komputer & Teknologi Informasi
Universitas Gunadarma
Depok, Indonesia
sultan@staff.gunadarma.ac.id

Abstrak— Teknologi perangkat bergerak yang sudah sedemikian cepat perkembangannya dapat dimanfaatkan dalam berbagai aplikasi di bidang kehidupan, diantaranya adalah pada bidang pendidikan, kedokteran, komunikasi, dan multimedia dan sebagian besar perangkat pintar yang digunakan saat ini bersistem operasi android. Aplikasi Android pengendali mobil dan kamera telah dirancang untuk dijalankan pada perangkat pintar Android. Aplikasi ini digunakan untuk mengendalikan mobil beserta kamera yang bertujuan untuk memantau tempat-tempat yang sulit dijangkau oleh manusia. Dari hasil uji coba, aplikasi Android ini dapat dioperasikan dengan baik dari segi fungsi. Dimana Aplikasi ini dapat menampilkan pandangan lingkungan yang ditangkap oleh kamera yang berada pada mobil dan juga dapat mengendalikan pergerakan mobil dan pergerakan kamera.

Kata kunci— *Android, App Inventor, Kendali, Perangkat Mobil, IP Kamera.*

I. PENDAHULUAN

Teknologi perangkat bergerak yang sudah sedemikian cepat perkembangannya dapat dimanfaatkan dalam berbagai bidang kehidupan, diantaranya adalah pada bidang pendidikan, kedokteran, komunikasi, dan multimedia. Penggunaan perangkat bergerak dapat juga digunakan pada media transportasi dan eksplorasi terutama untuk wilayah-wilayah yang tidak aman untuk dijangkau oleh manusia, sebagai contoh daerah pertambangan, daerah bencana alam, dan daerah-daerah terpencil seperti goa dan lorong.

Permasalahan yang timbul adalah bagaimana membuat sistem kendali pada mobil penjelajah yang dibekali dengan kamera sehingga dapat memantau wilayah-wilayah tersebut. Penelitian ini bertujuan untuk merancang dan membuat aplikasi kendali untuk mobil penjelajah, *MControlling*, aplikasi ini diimplementasikan pada perangkat pintar android

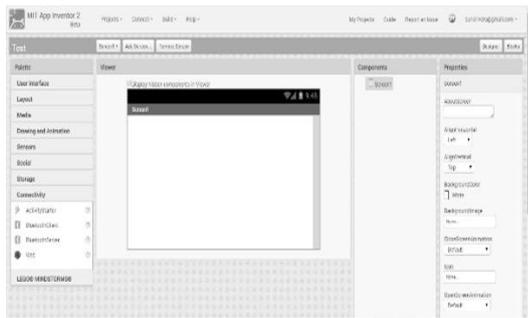
dikarenakan android merupakan *platform* dengan pengguna terbanyak dari tahun 2009 hingga 2014 sebagai sistem operasi pada perangkat bergerak. Secara mobilitas, perangkat android yang merupakan perangkat bergerak akan lebih mudah dibawa dan digunakan dibandingkan menggunakan *desktop atau notebook*. Selain itu aplikasi pada perangkat android juga *opensource* yang bisa digunakan secara bebas [3].

Tujuan dari penelitian ini adalah merancang dan membuat aplikasi android pengendali mobil dan kamera, sehingga memberi manfaat bagi pengguna khususnya untuk daerah-daerah yang sulit dijangkau dan berbahaya bagi manusia, untuk memperoleh informasi tentang keadaan lingkungan daerah tersebut dengan lebih aman dan akurat menggunakan perangkat bergerak berbasis Android.

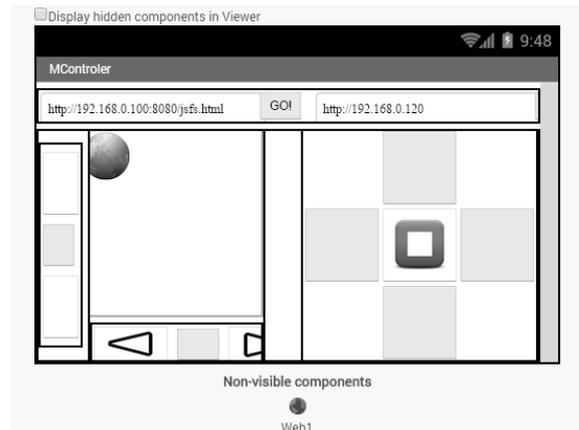
II. PERANCANGAN APLIKASI PENGENDALI MOBIL DAN KAMERA

Perancangan aplikasi *MControlling* yang dibuat pada *platform* Android menggunakan model pemrograman berbasis blok (*Software Block-Base Programming Tool*) *App Inventor*, yaitu sebuah model proses pengembangan perangkat lunak yang sistem pemrogramannya menggunakan blok-blok logika yang terkonfigurasi dengan layar interface yang di buat *App Inventor* juga merupakan aplikasi berbasis web yang langsung diakses melalui web browser yang dikembangkan oleh Massachusetts Institute of Technology (MIT). Proses pembuatan aplikasi dan juga program juga dilakukan langsung di web broser dan dapat dicoba dengan emulator atau ke perangkat bergerak [1].

Gambar 3. Struktur rancangan aplikasi

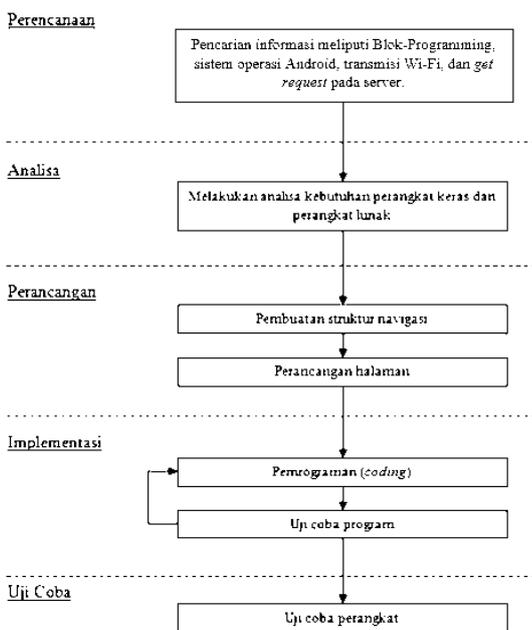


Gambar 1. Interface App Inventor

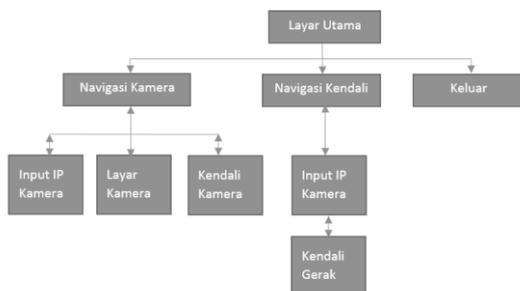


Gambar 4. Rancangan tampilan aplikasi pada interface app inventor

Dengan menggunakan App Inventor, dibuatlah diagram alur perancangan aplikasi sebagai berikut:

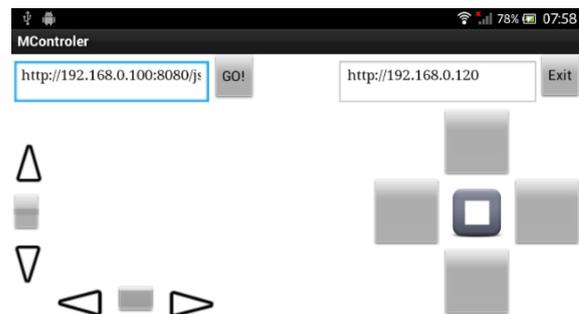


Gambar 2. Diagram perancangan aplikasi lampiran.



Pada saat aplikasi dijalankan, pengguna akan dihadapkan dengan sebuah menu utama aplikasi yang langsung merupakan sistem navigasi dan kendali. Aplikasi terdiri dari satu layer tampilan berbentuk *landscape*, dengan tujuan mempermudah dan mempercepat pengguna dalam melakukan kendali. Tampilan layar aplikasi berbentuk menyerupai sebuah *joystick* yang dibagi antara sisi kanan dan kiri.

Di bagian kiri layar merupakan navigasi layar kamera, untuk mendapatkan akses kamera yang kita tuju, maka user harus menuliskan alamat dari IP kamera tersebut pada *textbox* di atas kiri layar. Kemudian layar di bawahnya akan menampilkan kamera yang sedang melakukan *shot* secara langsung. Terdapat juga tombol yang memungkinkan untuk menggerakkan kamera ke arah yang diinginkan.



Gambar 5. Rancangan tampilan aplikasi pada Android

Pada bagian kanan layar berupa sistem kendali dari mobil penjelajah dengan tombol navigasi di sebelah kanan layar tampilan yang berisi arah mobil yang akan digerakan seperti maju, mundur, kiri, dan kanan, serta tombol stop di tengah apabila akan menghentikan mobil penjelajah secara paksa. Sebelum mengaktif sistem kendali tersebut, user juga harus

memasukan IP yang digunakan pada robot pengendali tersebut yang berbasis wifi. Terdapat juga tombol exit untuk menutup aplikasi.

III. IMPLEMENTASI DAN UJI COBA

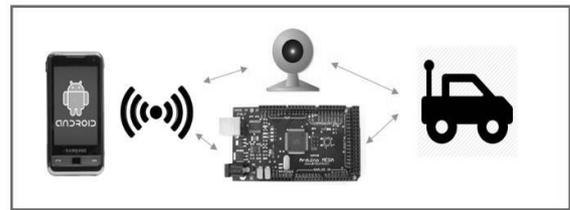
Dalam tahap implementasi ini dilakukan pembuatan *source code* pada *App Inventor 2*. Program dijalankan langsung dari berbagai jenis *web browser* yang sudah terintegrasi dengan java menggunakan *source code* yang berupa blok-blok logika. Program yang telah dibuat akan secara otomatis tersimpan pada akun google yang digunakan untuk menjalankan *App Inventor 2* secara *online*.

Pada rancangan aplikasi ini terdiri dari penulisan *source code* beberapa sub bagian dan juga langkah awal untuk mengintegrasikan aplikasi pada perangkat hardware yang akan dikendalikan. Saluran transmisi menggunakan koneksi nirkabel berupa wifi 2.4GHz yang sudah secara umum digunakan. Perangkat keras memiliki sistem kendali berupa webserver yang memberikan perintah masukan untuk semua sistem kendali pada perangkat mobil penjelajah. Sub program tersebut terdiri dari tiga bagian yaitu:

1. Sinkronisasi antara akses kendali pada hardware dan juga pada aplikasi yang dibuat dengan cara memasukan IP dari hardware yang akan diakses, hardware yang digunakan berupa mobil kendali dan IP Kamera.
2. Pengiriman perintah dari aplikasi ke perangkat yang dituju (mobil penjelajah) saat dilakukan penekanan pada tombol.
3. Pengiriman perintah berupa nul dari aplikasi saat tidak ada aktifitas yang dilakukan pada aplikasi.

Sebagai contoh perangkat hardware mobil penjelajah yang digunakan yaitu berupa *wireless car* dengan mikrokontroler Arduino dan TCP/IP yang digunakan 192.168.0.120. Berikut konfigurasi perintah yang akan dikirim dari aplikasi ke mobil penjelajah dalam bentuk *get request* ke *web server*. Perintah yang ada sebelumnya sudah terdapat pada mobil penjelajah. Sedangkan IP kamera yang memiliki IP sendiri langsung ditampilkan pada layar web pada aplikasi *MController*, IP kamera yang digunakan juga dapat berupa perangkat pintar android dengan aplikasi "*IP Webcam*" yang terhubung pada jaringan *wireless* yang sama.

Pada gambar 6, prototipe mobil sudah dibekali suatu sistem yang terdiri dari mikrokontroler, Wi-Fi, dan juga IP kamera. Mobil akan menerima data atau perintah yang dikirim melalui perangkat android melalui komunikasi *wireless* yang selanjutnya diproses oleh mikrokontroler. Kamera yang terletak pada mobil akan memiliki IP sendiri yang akan diakses oleh perangkat android, sehingga gambar yang ditangkap kamera akan muncul pada aplikasi.



Gambar 6. Sistem rancangan transmisi

TABEL 1. PERINTAH *GET REQUEST* DARI APLIKASI KE PERANGKAT MOBIL PENGENDALI

| No. | Tombol | Perintah | Pergerakan |
|-----|--------|------------------------|------------|
| 1. | Atas | Sending "/?maju1" | Maju |
| 2. | Bawah | Sending "/?mundur1" | Mundur |
| 3. | Kiri | Sending "/?kiri1" | Kiri |
| 4. | Kanan | Sending "/?kanan1" | Kanan |
| 5. | Stop | Sending "/?stop1" | Berhenti |

Setiap tombol memiliki fungsi pengiriman perintah yang berbeda ke perangkat mobil. Digunakan metode *request get* dari aplikasi ke *server* yang ada di perangkat mobil penjelajah. Berikut tampilan *source code* *App Inventor MController* untuk pengiriman *request get* saat dilakukan penekanan dan saat tidak melakukan penekanan tombol.

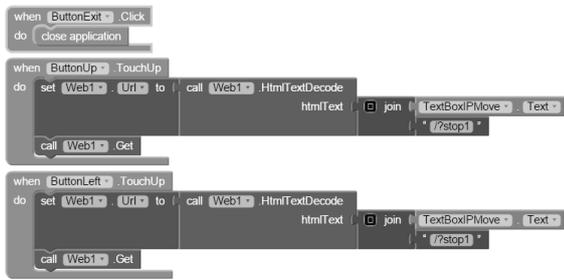
```

when ButtonCam.Click
do
  call WebViewer1.GoToUrl
  url | TextBoxIPWebcam.Text

when ButtonUp.TouchDown
do
  set Web1.Uri to call Web1.HtmlTextDecode
  htmlText | join | TextBoxIPMove.Text
  call Web1.Get
  "/?maju1"

when ButtonStop.TouchDown
do
  set Web1.Uri to call Web1.HtmlTextDecode
  htmlText | join | TextBoxIPMove.Text
  call Web1.Get
  "/?stop1"
  
```

Gambar 7. Block Code perintah pengambilan IP dan juga pengiriman *get request* saat dilakukan penekanan tombol



Gambar 8. Block Code perintah pengambilan IP dan juga pengiriman *get request* saat tidak ada tombol yang ditekan

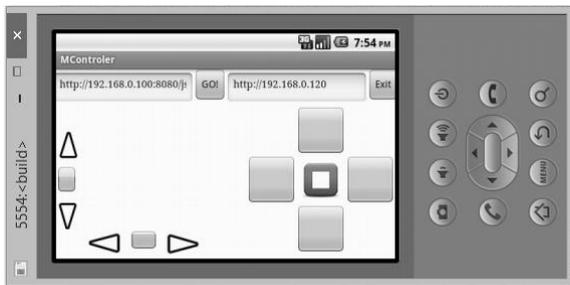
Gambar 11. Layar Kamera saat TCP/IP sesuai



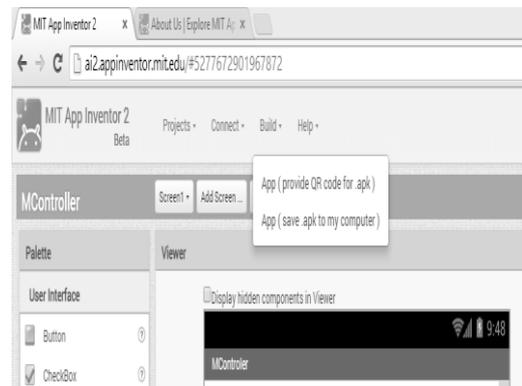
Gambar 12. Layar kamera saat TCP/IP tidak sesuai

Setelah semua *source code* selesai disusun dan disimpan, berikutnya yaitu tahapan uji coba aplikasi. Tahap ini dilakukan pengujian aplikasi yang telah selesai pada *emulator* Android yang terdapat pada *App Inventor*. Pengujian ini dilakukan untuk mengetahui apakah aplikasi layanan berbasis lokasi yang dibuat sudah dapat berfungsi dengan baik. Berikut adalah hasil uji coba tampilan aplikasi yang dijalankan pada *emulator* *Android* :

Setelah aplikasi diuji coba pada emulator *Android* menunjukkan hasil yang diharapkan, langkah selanjut yaitu dilakukan uji coba pada perangkat *Android*. Untuk melakukan uji coba ini, dipergunakan file *.apk* (Gambar.13) untuk melakukan pemasangan pada perangkat bergerak *Android*. File *.apk* tersebut dapat diperoleh langsung dari *web server* pada *App Inventor* dengan cara mengklik tombol “ “ pada tab “*Build*”. Kemudian diinstall ke perangkat bergerak *Android*.



Gambar 9. Tampilan muka *Mcontroller*

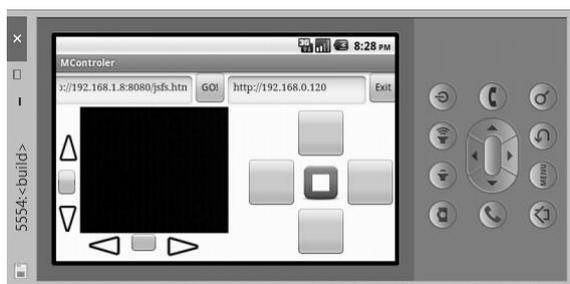


Gambar 13. Penyimpanan file program ke bentuk *.apk*.

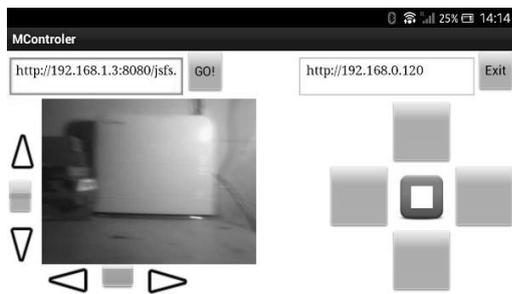


Gambar 10. Input TCP/IP pada *textbox*

Selanjutnya melakukan uji coba pada perangkat *Android*, gambar 14 adalah hasil uji coba tampilan aplikasi yang dijalankan pada perangkat *Sony Xperia SL*. Aplikasi diuji dengan IP kamera ukuran resolusi 640x480 dan 320x240 piksel menggunakan kamera ponsel dengan aplikasi *IP Webcam*.



(a)



(b)

Gambar 14. Hasil ujicoba aplikasi pada perangkat Android Sony Xperia SL, (a) dengan resolusi kamera 640x480 dan (b) dengan resolusi kamera 320x240

Aplikasi ini juga di uji cobakan dengan beberapa perangkat Android, hasilnya terdapat pada tabel 1.

TABEL 1. HASIL UJICOBA APLIKASI PADA BEBERAPA PERANGKAT ANDROID

| Tipe Ponsel | Versi Android | Fungsional Aplikasi | Stabilitas Aplikasi | Tampilan Kamera |
|---------------------|------------------|---------------------|---------------------|-----------------|
| Samsung Galaxy S2 | Jelly Bean 4.1 | Baik | Baik | Baik |
| Samsung Galaxy Mini | Gingerbread 2.3 | Baik | Kurang | Kurang |
| Sony Xperia SL | Jelly Bean 4.1.2 | Baik | Baik | Baik |

Berdasarkan hasil uji coba yang dilakukan pada tiga buah perangkat Android, aplikasi ini berjalan dengan baik di semua tipe perangkat android dengan versi OS yang berbeda. Dari sisi tampilan, aplikasi ini menggunakan layout linear dengan listview, sehingga tampilan akan menyesuaikan dengan resolusi dari perangkat android. Sedangkan dari sisi stabilitas, terjadi lagging pada beberapa perangkat android dikarenakan spesifikasinya dan begitu juga pada tampilan kamera yang membutuhkan akses yang cepat dari *client* (perangkat android).

IV. KESIMPULAN

Berdasarkan hasil uji coba, aplikasi pengendali mobil dapat dioperasikan dengan baik dari segi fungsi pada tiga perangkat berbasis Android. Aplikasi ini dapat menampilkan pandangan lingkungan disekitar yang ditangkap oleh kamera yang berada pada mobil penjelajah serta dapat mengendalikan pergerakan kamera dan mobil. Jadi aplikasi ini siap untuk diimplementasikan dalam sistem mobil penjelajah. Namun aplikasi ini memiliki kekurangan, diantaranya yaitu aplikasi ini hanya mengakses mobil penjelajah yang sebelumnya sudah diketahui segala unsur program dan juga web server dari mobil penjelajah itu sendiri atau bisa juga dengan melakukan konfigurasi ulang pada mobil penjelajah disesuaikan dengan program pada aplikasi MController, dengan begitu aplikasi ini

bisa digunakan secara umum sebagai sistem kendali yang efisien.

DAFTAR PUSTAKA

- [1] Gestwicki, P., & Ahmad, K., "App inventor for Android with studio-based learning", J. Comput. Sci. Coll, 2011.
- [2] Kerimil,"The cheapest and simplest method to control arduino through Ethernet". <http://www.instructables.com/id/The-cheapest-and-simplest-method-to-control-arduino/> (diakses tanggal 15 Desember), 2013.
- [3] Tim Laboratorium Sistem Komputer, "Mobile Computing System". Universitas Gunadarma, 2011.
- [4] Manning. 2014. Manning Early Access Program App Inventor. Manning Publication, 2014.
- [5] MIT App Inventor. "About App Inventor". <http://appinventor.mit.edu/explore/aboutus.html>. (diakses tanggal 20 Desember 2014)..
- [6] Soares, Andrea," Reflections on Teaching App Inventor for Non-Beginner Programmers: Issues, Challenges and Opportunities". Illinois University Carbondale, IL 62901, USA,2013.
- [7] Wolber, D., Abelson, H., Spertus, E., & Looney, L,"App Inventor: Create Your Own Android Apps: O'Reilly Media, 2011.