

**APLIKASI PENGENDALIAN DAN MONITORING ROBOT
YANG DI LENGKAPI IP CAMERA BERBASIS ANDROID**

SKRIPSI



**Disusun Oleh :
SU'AADA NURLAILA
09.12.509**

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama : Su'aada Nurlaila
NIM : 0912509
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer S-1

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 19 September 2013

METERAI
TEMPEL
D286BABF607300025
6000
DJP
Suat Pernyataan,


Su'aada Nurlaila
NIM :0912509

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala limpahan rahmat-Nya sehingga penelitian berjudul Aplikasi Pengendalian Dan Monitoring Robot Yang Di Lengkapi IP Camera Berbasis Android.

Penelitian ini dibuat untuk memenuhi salah satu syarat dalam memperoleh gelar sarjana teknik. Ucapan terima kasih yang sebesar-besarnya kami sampaikan pada :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ibrahim Ashari, ST, MT selaku Ketua Jurusan Elektro.
3. Bapak Ir. H. Anang Subardi, MT, selaku Dekan Fakultas Teknologi Industri.
4. Bapak Aryunto Soetedjo, Dr. Eng., ST., MT Selaku Sekretaris Jurusan Teknik Elektro S-1 IIN Malang dan juga selaku Dosen pembimbing 1.
5. Bapak Ir. Yusuf Ismail Nakhoda, MT Selaku Dosen pembimbing 2.
6. Kedua orang tua yang selalu mendukung dalam keadaan apapun.
7. Dan semua Pihak yang telah membantu dalam penulisan dan penyusunan penelitian ini.

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna, untuk itu kritik dan saran dari pembaca sangat penulis harapkan untuk perbaikan penelitian ini.

Malang, Agustus 2013

Penulis

Daftar Isi

Halaman Judul	i
Lembar Persetujuan	ii
Abstrak	iii
Kata Pengantar	iv
Daftar Isi	v
Daftar Gambar	viii
Daftar Tabel	x

BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Penulisan.....	4

BAB II LANDASAN TEORI

2.1 Robot.....	5
2.1.1 Pengertian Robot.....	5
2.1.2 Perkembangan Robot.....	5
2.2 Pengertian Android.....	8
2.2.1 Sejarah Perkembangan Android OS.....	9
2.3 Eclips.....	11
2.4 Borland Delphi.....	12
2.4.1 IDE Delphi.....	13
2.5 Bluetooth.....	14
2.5.1 Pengertian Bluetooth.....	14
2.5.2 Perkembangan Sejarah Bluetooth.....	15
2.5.3 Cara Kerja Bluetooth.....	16

2.6	Wifi.....	17
2.6.1	Pengertian Wifi	17
2.6.2	Perkembangan Sejarah Wifi.....	18
2.6.3	Cara Kerja Wifi	19
2.7	Sensor Accelerometer	20
2.7.1	Pengertian Sensor Accelerometer	20
2.7.2	Perinsip Kerja Sensor Accelerometer	21
2.8	IP Camera.....	22
2.8.1	Pengertian IP Camera.....	22
2.9	Sistem Minimum Mikrokontroler	23
2.10	Motor DC	23

BAB III ANALISA DAN PERANCANGAN SISTEM

3.1	Analisa Sistem.....	26
3.1.1	Deskripsi Umum Sistem	26
3.1.2	Analisa Kebutuhan	27
3.1.2.1	Smartphone Yang Digunakan	28
3.1.2.2	Minimum Sistem Yang Digunakan	29
3.1.2.3	Bluetooth Slave Yang Digunakan.....	29
3.1.2.4	Prangkat Lunak Yang Digunakan.....	30
3.1.2.5	Robot Yang Digunakan	32
3.2	Pemodelan Aplikasi.....	32
3.2.1	Diagram Alir Monitoring Robot	33
3.2.2	Diagram Alir Kendali Robot.....	34
3.3	Perencanaan Dan Desain Aplikasi	35
3.3.1	Perencanaan Software	35
3.3.2	Perencanaan Antaramuka Aplikasi	40
3.3.2.1	Rancangan Desain Tampilan Utama.....	40

Daftar Gambar

Gambar 2.1 Logo Android.....	9
Gambar 2.2 Jendela Utama Delphi 7.0 Dan Bagian-Bagiannya.....	13
Gambar 2.3 Logo Bluetooth	15
Gambar 2.4 Blok Fungsional Sistem Bluetooth	16
Gambar 2.5 Layer-layer Pada Sistem Bluetooth.....	17
Gambar 2.6 Cara Kerja Wifi.....	20
Gambar 2.7 Accelerometer	21
Gambar 2.8 IP Camera.....	22
Gambar 2.9 Minimum Sistem.....	23
Gambar 2.10 Konstruksi Motor DC	24
Gambar 2.11 PenentuanArah Gaya Pada Berarus Listrik Dalam Medan Magnet.....	25
Gambar 2.12 Contoh Motor DC	26
Gambar 3.1 Diagram Umum Sistem.....	27
Gambar 3.2 Samsung Galaxy Tab 2.7.0	28
Gambar 3.3 Bluetooth Slave.....	30
Gambar 3.4 IP Camera TL-SC3130G.....	32
Gambar 3.5 Flowchart monitoring.....	33
Gambar 3.6 Flowchart kontrol robot	34
Gambar 3.7 Konfigurasi java JDK	36
Gambar 3.8 Add Repository ADT plugin.....	36
Gambar 3.9 Instal NDK Plugins dan Developer Tool	37

Daftar Tabel

Tabel 4.1 Spesifikasi kebutuhan untuk pengujian	45
Tabel 4.2 Hasil pengujian jarak kontroler dengan robot	49
Tabel 4.3 Pengujian arah gerak robot	52
Tabel 4.4 Pengujian pada smartphone berbeda.....	53



BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Pada masa sekarang *mobile phone* sudah banyak sekali dijumpai dimana saja, berbagai macam jenis dan *type*. Hal ini yang menyebabkan *developer software mobile phone* untuk membuat perangkat lunak yang dapat memberikan kemudahan, juga dapat memenuhi kebutuhan manusia sehari-hari serta mampu menjalankan jenis aplikasi yang beragam pada berbagai sistem operasi, salah satunya adalah sistem operasi berbasis android dalam memanfaatkan teknologi yang ada pada saat ini. Android ini merupakan sistem operasi yang berbasis Linux seperti *smartphone* dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang agar dapat menciptakan aplikasi yang digunakan oleh bermacam peranti bergerak.

Seiring dengan perkembangan teknologi yang semakin maju, teknologi robot pun digunakan sebagai alat bantu manusia yang memiliki beberapa kelebihan dan akan terus berkembang. Salah satu pengembangan dalam teknologi robot adalah pengendalian serta monitoring robot menggunakan *smartphone* berbasis android atau tablet android yang tadinya hanya menggunakan *joystick* yang terhubung dengan kabel sekarang dapat di gantikan dengan memanfaatkan bluetooth dan accelerometer yang ada pada *smartphone* android. Aplikasi ini merupakan pengembangan dari aplikasi pengontrol robot dengan bluetooth dan sensor accelerometer pada *smartphone* android. Pengembangan yang dilakukan adalah menambahkan aplikasi agar mempunyai kemampuan memonitoring area di sekitar robot dari jarak jauh setelah keluar dari pemantauan pengendali.

Untuk itu pembuatan aplikasi ini tidak hanya sekedar untuk mengendalikan melainkan mempunyai kemampuan untuk memonitoring area disekitar robot yang menggunakan dua buah komunikasi yaitu komunikasi Bluetooth dan komunikasi *wireless*. Aplikasi kendali robot ini akan memanfaatkan sensor accelerometer yang terdapat pada *smartphone* android dan menggunakan komunikasi bluetooth untuk mengirimkan data pergerakan robot. Aplikasi monitoring robot ini juga di lengkapi dengan sebuah IP kamera *wireless*, yang digunakan sebagai indera robot dan menggunakan komunikasi *wireless* untuk mengirim *streaming video* ke *smartphone*

4. Menggunakan IP kamera yang sudah di lengkapi dengan wifi (include)
5. Tidak membahas module komunikasi robot dengan smartphone android.

1.6 Metodologi Penelitian

Metode penelitian dalam penulisan skripsi ini adalah sebagai berikut:

1. Studi Literatur

Pengumpulan data dan informasi seperti dilakukan dengan mencari bahan-bahan kepustakaan dan referensi mulai dari perangkat keras dan perangkat lunak yang akan dibangun dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan pada perancangan program aplikasi kendali dan monitoring robot.

2. Analisa Kebutuhan Aplikasi

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kategori-kategori yang harus di tampilkan pada program yang akan dibuat dalam pembuatan aplikasi kendali dan monitoring.

3. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan program, akan dijadikan acuan dalam merancang kerangka secara global yang menggambarkan mekanisme dari program yang akan dibuat dan di implementasikan ke dalam sistem.

4. Eksperimen dan Evaluasi

Setelah melalui beberapa tahap mulai dari pengumpulan data, pada tahap ini program yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan penyempurnaan program jika diperlukan.

5. Pembahasan

Dalam pembahaas ini akan di jelaskan hasil yang di dapatkan setelah melakukan pengujian program aplikasi kendali dan monitoring robot, dan mengetahui sejauh mana kesulitan dalam proses pembuatan, apakah aplikasi yang di buat sudah berhasil atau tidak.

6. Kesimpulan

Berisi tentang kesimpulan dari hasil pengujian aplikasi yang telah dibuat.

1.7 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan karya skripsi ini, sistematika penulisan disusun sebagai berikut:

1. BAB I : Pendahuluan
Berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.
 2. BAB II : Tinjauan Pustaka
Membahas tentang dasar teori mengenai permasalahan yang berhubungan dengan penelitian.
 3. BAB III : Perancangan dan Pengujian Program
Membahas tentang kebutuhan media yang digunakan dalam membangun program dan menjelaskan kerangka program secara global.
 4. BAB IV : Pembuatan dan Pengujian Program
Berisi tentang pembahasan mengenai langkah-langkah pembuatan program serta pengujian terhadap program tersebut.
 5. BAB V : Penutup
Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan program selanjutnya.
-



BAB II LANDASAN TEORI

2.1 Robot

2.1.1 Pengertian Robot

Dalam pembuatan sistem pengontrolan robot menggunakan Bluetooth dan sensor Accelometer, serta monitoring robot yang dilengkapi IP Camera menggunakan komunikasi Wifi pada Smartphone Android ini, mengacu pada beberapa dasar teori yang mendukung sistem kerja dari pengontrol dan monitoring tersebut. Adapun dasar teori yang didalam perancangan pengontrol dan monitoring ini adalah sebagai berikut.

Robot berasal dari bahasa Czech, *robota*, yang berarti pekerja. Pada dasarnya robot dibuat untuk mendukung dan membantu pekerjaan manusia, seperti yang banyak terlihat dibidang industry dimana robot dapat meningkatkan hasil produksi industri. Kata robot diperkenalkan dalam bahasa Inggris pada tahun 1921 oleh Wright Karel Capek dalam salah satu drama satiristiknya, *R.U.R (Rossum's Universal Robots)*.

Sedangkan pengertian robot adalah system atau alat yang dapat berperilaku atau meniru perilaku manusia dengan tujuan untuk menggantikan dan mempermudah kerja/aktifitas manusia. Awal muncul robot dapat diketahui dari bangsa Yunani kuno yang membuat patung yang dapat dipindah-pindahkan. Sekitar 270 BC, seorang insinyur Yunani membuat dengan komponen yang dapat dipindahkan. Tahun 1941, barulah istilah robotics digunakan dalam teknologi robot oleh penulis fiksi ilmiah Isaac Asimov. Dia juga memprediksi akan munculnya robot-robot industri canggih dimasa datang. Jika kita lihat hari ini, maka apa yang dibayangkan olehnya terbukti dimana begitu pesatnya perkembangan robot-robot industri saat ini. Istilah revolusi robot, robot age atau era robot sudah menjadi hal biasa untuk menjelaskan perkembangan itu. Robotics diterima sebagai istilah atau kata untuk mendeskripsikan semua kemajuan teknologi yang berhubungan dengan robot.

2.1.2 Perkembangan Robot

Ketika Georde Devil dan Joseph Engelberger membentuk perusahaan robot pertama kali tahun 1956. Devil memprediksi robot akan menjadi bagian penting di industri sebagai operator pabrik dan membantu pekerja dalam menjalankan mesin-mesin pabrik. Beberapa tahun kemudian atau tepatnya 1961, General Motor pertama

robot. Terakhir pada tahun 2000 Honda memamerkan robot yang dibangun bertahun-tahun lamanya bernama ASIMO, serta disusul oleh Sony yaitu robot anjing AIBO.

Robot banyak dibuat oleh industry Riset, universitas, Departemen Pertahanan, serta institusi besar lainnya seperti NASA dan SONY. Saat ini hamper semua industry manufaktur menggunakan robot, karena biaya per jam untuk mengoperasikan robot jauh lebih murah dibandingkan menggunakan manusia. Robot pada awalnya digunakan untuk melakukan fungsi spesifik, misalnya pengecoran, penyoderan, dan lain-lain. Namun saat ini sudah banyak robot yang melakukan banyak fungsi.

Bentuk robot seperti manusia tidak lagi diperhatikan meski perkembangan robot android atau humanoid tetap berlangsung dan mengalami penyempurnaan. Kini robot adalah pekerja industri atau berupa tangan dan lengan yang dikontrol oleh computer dan dapat dirubah fungsinya dengan mengedit program robot. Bentuk robot industri ini lebih dikenal sekarang dibanding robot menyerupai manusia.

Kemampuan robot untuk melakukan gerakan manusia sangat membantu dunia industri seperti industri mobil, proses pengelasan, perakitan, pemindahan dan banyak lagi. Gerakan berulang yang presisi adalah salah satu keunggulan robot daripada manusia sehingga didapat hasil produksi yang konstan dan standard.

Robot industri harus diprogram untuk melakukan semua step gerakan atau kerja sebelum ia digunakan. Tahap awal ini bisa disebut merangkai atau membangun pola berfikirnya robot. Benda kerja harus ditempatnya ditempat yang pasti dan tidak berubah-ubah selama proses (meski sekarang kemajuan object recognition sudah maju namun dalam prakteknya benda kerja masih harus diposisikan ditempat yang tetap). Jika benda kerja meleset dari posisinya maka proses akan salah dan robot tidak bisa mengoreksinya. Robot tidak bisa melihat dan mendengar. Dia tidak bisa merasakan objek dan memprediksi adanya kesalahan dan robot tidak memiliki kemampuan mengadopsi situasi baru yang terjadi disekitarnya.

Robot memberikan keuntungan tersendiri bagi pekerja industri dan suatu negara dimana ia bisa memperbaiki kualitas hidup manusia karena bebas dari pekerjaan yang menjenuhkan, kotor dan penuh resiko atau dalam istilahnya 3D= Dull, Dirty and Dangerous. Benar bahwa robot akan menimbulkan pengangguran tapi jangan lupa robot juga menciptakan lapangan pekerjaan; Insinyur robot, Teknisi, Sales, Programmer dan

Pengawas/supervisor. Robot memberikan keuntungan bagi industri karena adanya peningkatan output dan perbaikan kualitas. Industri robot tidak mengenal lelah dan keluhan, ia bisa bekerja tanpa lelah siang malam dengan performance yang sama. Akibatnya, biaya produk per unit akan turun, menaikkan keuntungan dan memberi dampak positif terhadap pasar serta ekonomi dunia secara keseluruhan. Namun secara garis besar robot dapat diklasifikasikan menjadi beberapa jenis antara lain:

- 1) Robot industri
- 2) Robot antariksa
- 3) Robot transportasi
- 4) Robot perang
- 5) Robot kendali jarak jauh
- 6) Robot kedokteran
- 7) Robot riset
- 8) Robot bermain, dll

2.2 Pengertian Android

Android adalah sistem operasi berbasis Linux yang dirancang terutama untuk perangkat layar sentuh (touchscreen) mobile seperti smartphone dan komputer tablet. Dengan sistem distribusi open sources yang digunakan memungkinkan para pengembang untuk menciptakan beragam aplikasi menarik yang dapat dinikmati oleh para penggunanya, seperti game, aplikasi dan lain-lain. Hal itulah yang membuat smartphone berbasis Android ini lebih murah dibanding gadget yang sejenis. *Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.*

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS), seperti yang terdapat pada smartphone NEXUS keluaran google yang bekerja sama dengan vendor smartphone HTC, dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD), jenis yang kedua ini jenis-jenis yang banyak terdapat pada smartphone android di pasaran.



Gambar 2.1 Logo Android

2.2.1 Sejarah Pengembangan Android OS

Android, Inc didirikan di Palo Alto, California , Amerika Serikat pada bulan oktober 2003 oleh Andy Rubin, Rich Miner , Nick Sears, dan Chris White. Awalnya dikembangkan oleh Android Inc. Setelah itu kemudian di akuisisi oleh Google pada pertengahan tahun 2005 dan mengubah nama penyedia aplikasi Android dari Android market menjadi Google play. Android diresmikan pada tahun 2007 dan ponsel Android pertama kali dijual pada Oktober 2008, berikut beberapa versi android:

1. Android Versi 1.1, (Tanpa Nama)

Dirilis pada 9 maret 2009. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android Versi 1.5 Cupcake

Dirilis pada pertengahan mei 2009. Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3. Android Versi 1.6 Donut,

Dirilis pada september 2009 dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur

lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4. Android versi 2.0/2.1 Eclair

Diluncurkan pada 3 desember 2009, perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

5. Android versi 2.2 froyo.

Diperkenalkan pada 20 mei 2010, Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

6. Android d versi 2.3 Gingerbread.

Diluncurkan pada 6 desember 2010. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0 honeycomb.

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.

8. Android versi 4.0 ice cream sandwich.

Dirilis pada 19 oktober 2011, membawa fitur Honeycomb yang dikhususkan penggunaannya pada tablet untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

9. Android versi 4.1 Jelly Bean

Jelly bean sendiri merupakan penerus dari sistem operasi android 4.0 ice cream sandwich yang diharapkan nantinya akan mampu menyaingi sistem - sistem operasi terbaru dari apple (IOS6), microsof (Windows phone 8/WP8) dan terakhir dari RIM (BlackBerry 10/BB10). Fitur dari OS android 4.1 jelly bean ini adalah pada bagian area notifikasi yang bisa dikostumasi dan tentu saja kini kita bisa melakukan action atau tindakan terhadap hal-hal yang muncul di notifikasi. Misalnya saat Ada sebuah email masuk, maka cukup dengan membuka area notifikasi kita bisa mendapatkan pilihan mau diapakan email tersebut, tanpa harus membuka aplikasi email terlebih dahulu dan fitur lainnya pada jelly bean ini terdapat pada home screen yang membuat user lebih leluasa untuk dipersonalisasi, Accessibility seperti gesture dan speech feedback, serta tampilan menu kamera yang lebih diperbaharui dengan hadirnya fitur Film Strip View dan Faster Deletion dan tentunya fitur Undo.

2.3 Eclips

Menurut Nasruddin Sifaat h (Pemrograman aplikasi mobeli smartphone dan tablet PC berbasis android 2012:16) Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse:

- a. **Multi-platform:** Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- b. **Mult-language:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- c. **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in. Eclipse dibuat dari kerja sama antara perusahaan-perusahaan anggota 'Eclipse Foundation' (beserta individu-individu lain). Banyak nama besar yang ikut dalam 'Eclipse 23 Foundation', termasuk IBM, BEA, Intel, Nokia, Borland. Eclipse bersaing langsung dengan Netbeans IDE. Plugin tambahan pada Eclipse jauh lebih banyak dan bervariasi dibandingkan IDE lainnya.

1. IntelliJ IDEA (commercial, free 30-day trial)
2. Oracle JDeveloper (free)
3. Xinox JCreator (ada versi berbayar maupun free)

2.4 Borland Delphi

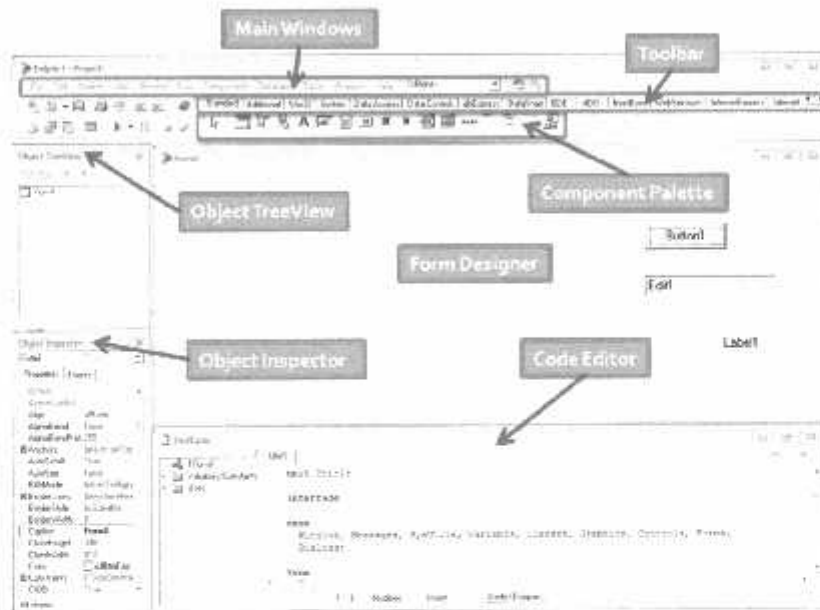
Borland Delphi merupakan salah satu *software* aplikasi pemrograman yang menduduki peringkat 50 besar dunia. Bahasa pemrograman yang digunakan adalah Pascal. Delphi 7.0 memiliki fitur dan fasilitas yang cukup lengkap, kinerja yang bagus pada spesifikasi *hardware* yang minimum, selain itu juga cukup populer dan banyak digunakan para pemrogram.

Beberapa keunggulan yang dimiliki bahasa pemrograman ini antara lain mudah dipelajari dan digunakan, memiliki fasilitas *source completion* yang digunakan untuk melengkapi kode yang dituliskan pada kode editor secara otomatis, pemrograman yang terstruktur, kecepatan kompilasi, serta pola desain yang menarik.

Delphi merupakan sebuah bahasa pemrograman untuk membuat program atau aplikasi komputer berbasis Windows. Bahasa pemrograman di Delphi merupakan

bahasa prosedural, berbasis objek, dan termasuk keluarga visual. Bahasa prosedural artinya bahasa atau sintaknya mengikuti urutan tertentu atau prosedur. Perintah – perintah untuk membuat objek dilakukan secara visual merupakan maksud dari bahasa pemrograman keluarga visual. Sedangkan berbasis objek maksudnya semua komponen yang ada merupakan objek – objek yang memiliki nama, properti, dan *procedure*. Komponen yang ada tidak cuma berupa teks yang sebenarnya program kecil, tetapi muncul berupa gambar – gambar sehingga Delphi disebut juga *visual programming*.

2.4.1 IDE Delphi



Gambar 2.2

Jendela Utama Delphi 7.0 Dan Bagian - Bagianya

Pada jendela utama Borland Delphi terdapat beberapa menu seperti yang terdapat pada aplikasi berbasis Windows pada umumnya. Selain itu terdapat beberapa *toolbar* yang berfungsi sebagai *shortcut* menu, dan terdapat juga *component palette* yaitu sebagai tempat terdapatnya komponen.

Bagian bagian pada IDE Delphi :

1. Main Windows

Berisi menu - menu perintah pada aplikasi Windows pada umumnya antara lain Menu, Edit, *Search*, dan lainnya.

2. Toolbar

Toolbar merupakan kumpulan komponen - komponen yang dikompakkan sesuai dengan fungsinya.

3. Component Palette

Component Palette berisi kumpulan ikon yang melambangkan komponen-komponen pada VCL.

4. Object Tree View

Object Treeview berisi daftar komponen yang telah diletakkan di *form designer*.

5. Form Designer

Form Designer merupakan tempat merancang jendela aplikasi. Perencanaan form dapat dilakukan dengan menempatkan komponen-komponen yang diambil dari *component palette*.

6. Object Inspector

Object Inspector digunakan untuk merubah karakteristik dari sebuah komponen. Pada *Object Inspector* ini terdapat 2 tab yaitu tab properti dan events. Tab properti digunakan untuk mengubah properti dari komponen seperti mengganti warna, nama, judul form dan sebagainya. Sedangkan pada Tab Events digunakan untuk menempatkan kode program dalam menangani suatu kejadian seperti OnClick , OnClose.

7. Code Editor

Code Editor adalah tempat dimana kode program dituliskan. Pada *code editor* ini pernyataan-pernyataan bahasa object pascal ditempatkan seperti procedure, function dan sebagainya.

2.5 Bluetooth

2.5.1 Pengertian Bluetooth

Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frequency hopping tranceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas.

Bluetooth sendiri dapat berupa card yang bentuk dan fungsinya hampir sama dengan card yang digunakan untuk wireless local area network (WLAN) dimana menggunakan frekuensi radio standar IEEE 802.11, hanya saja pada bluetooth

mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah.



Gambar 2.3 Logo Bluetooth

Pada dasarnya bluetooth diciptakan bukan hanya menggantikan atau menghilangkan penggunaan kabel didalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi mobile wireless dengan biaya yang relatif rendah, konsumsi daya yang rendah, interoperability yang menjanjikan, mudah dalam pengoperasian dan mampu menyediakan layanan yang bermacam-macam.

2.5.2 Perkembangan Sejarah Bluetooth

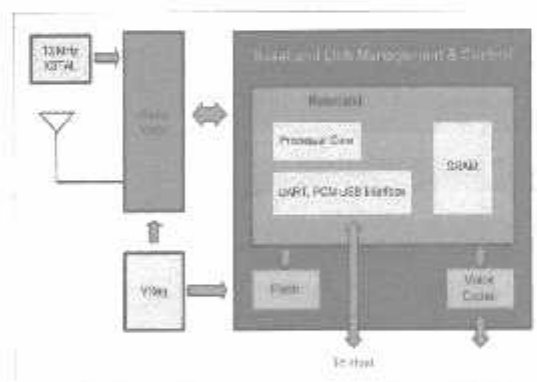
Nama bluetooth berawal dari proyek prestisius yang dipromotori oleh perusahaan-perusahaan raksasa internasional yang bergerak di bidang telekomunikasi dan komputer, di antaranya Ericsson, IBM, Intel, Nokia, dan Toshiba.

Proyek ini di awal tahun 1998 dengan kode nama bluetooth, karena terinspirasi oleh seorang raja Viking (Denmark) yang bernama Harald Blatand. Raja Harald Blatand ini berkuasa pada abad ke-10 dengan menguasai sebagian besar daerah Denmark dan daerah Skandinavia pada masa itu. Dikarenakan daerah kekuasaannya yang luas, raja Harald Blatand ini membiayai para ilmuwan dan insinyur untuk membangun sebuah proyek berteknologi metamorfosis yang bertujuan untuk mengontrol pasukan dari suku-suku di daerah Skandinavia tersebut dari jarak jauh. Maka untuk menghormati ide raja Viking tersebut, yaitu Blatand yang berarti bluetooth (dalam bahasa Inggris) proyek ini diberi nama.

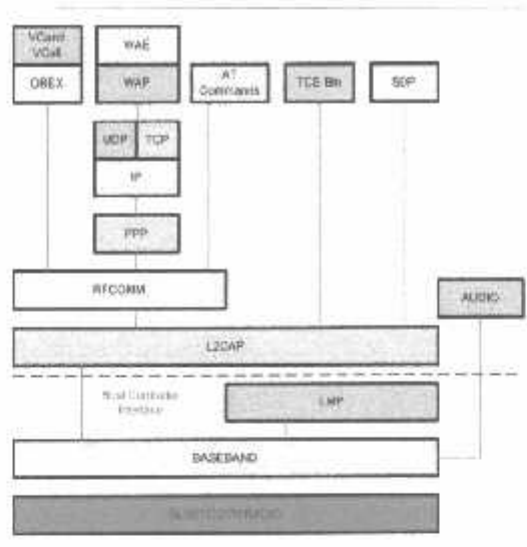
2.5.3 Cara Kerja Bluetooth

Protokol bluetooth menggunakan sebuah kombinasi antara circuit switching dan packet switching. Bluetooth dapat mendukung sebuah kanal data asinkron, tiga kanal suara sinkron simultan atau sebuah kanal dimana secara bersamaan mendukung layanan data asinkron dan suara sinkron. Setiap kanal suara mendukung sebuah kanal suara sinkron 64 kb/s. Kanal asinkron dapat mendukung kecepatan maksimal 723,2 kb/s asimetris, dimana untuk arah sebaliknya dapat mendukung sampai dengan kecepatan 57,6 kb/s. Sedangkan untuk mode simetris dapat mendukung sampai dengan kecepatan 433,9 kb/s.

Sebuah perangkat yang memiliki teknologi wireless bluetooth akan mempunyai kemampuan untuk melakukan pertukaran informasi dengan jarak jangkauan sampai dengan 10 meter (~30 feet), bahkan untuk daya kelas 1 bisa sampai pada jarak 100 meter. Sistem bluetooth terdiri dari sebuah radio transceiver, baseband link Management dan Control, Baseband (processor core, SRAM, UART, PCM USB Interface), flash dan voice code. sebuah link manager. Baseband link controller menghubungkan perangkat keras radio ke baseband processing dan layer protokol fisik. Link manager melakukan aktivitas-aktivitas protokol tingkat tinggi seperti melakukan link setup, autentikasi dan konfigurasi. Secara umum blok fungsional pada sistem bluetooth secara umum dapat dilihat pada Gambar dibawah ini.



Gambar 2.4. Blok fungsional sistem bluetooth.



Gambar 2.5. Layer-layer pada sistem bluetooth.

Tiga buah lapisan fisik yang sangat penting dalam protokol arsitektur Bluetooth ini adalah :

1. Bluetooth radio, adalah lapis terendah dari spesifikasi Bluetooth. Lapis ini mendefinisikan persyaratan yang harus dipenuhi oleh perangkat tranceiver yang beroperasi pada frekuensi 2,4 GHz ISM.
2. Baseband, lapis yang memungkinkan hubungan RF terjadi antara beberapa unit Bluetooth membentuk piconet. Sistem RF dari bluetooth ini menggunakan frekuensi-hopping-spread spectrum yang mengirimkan data dalam bentuk paket pada time slot dan frekuensi yang telah ditentukan, lapis ini melakukan prosedur pemeriksaan dan paging untuk sinkronisasi transmisi frekuensi hopping dan clock dari perangkat bluetooth yang berbeda.
3. LMP, Link Manager Protocol, bertanggung jawab terhadap link set-up antar perangkat Bluetooth. Hal ini termasuk aspek securiti seperti autentifikasi dan enkripsi dengan pembangkitan, penukaran dan pemeriksaan ukuran paket dari lapis baseband.

2.6 Wi-fi

2.6.1 Pengertian Wi-fi (Wireless Fidelity)

Wi-Fi merupakan singkatan dari *Wireless Fidelity*, Wi-fi adalah teknologi jaringan tanpa kabel yang menggunakan frekuensi tinggi, frekuensi yang di gunakan

oleh teknologi Wi-Fi berada pada spektrum 2,4 Ghz sehingga digunakan untuk Jaringan Lokal Nirkabel (Wireless Local Area Networks - WLAN) yang didasari pada spesifikasi IEEE 802.11. Standar terbaru dari spesifikasi 802.11a atau b, seperti 802.11g, saat ini sedang dalam penyusunan, spesifikasi terbaru tersebut menawarkan banyak peningkatan mulai dari luas cakupan yang lebih jauh hingga kecepatan transfernya.

Awalnya Wi-Fi ditujukan untuk penggunaan perangkat nirkabel dan Jaringan Lokal (LAN), namun saat ini lebih banyak digunakan untuk mengakses internet. Hal ini memungkinkan seseorang dengan komputer dengan kartu nirkabel (wireless card) atau personal digital assistant (PDA) untuk terhubung dengan internet dengan menggunakan access point (atau dikenal dengan hotspot) terdekat. Wi-Fi (Wireless Fidelity) adalah koneksi tanpa kabel seperti handphone dengan mempergunakan teknologi radio sehingga pemakainya dapat mentransfer data dengan cepat dan aman. Wi-Fi tidak hanya dapat digunakan untuk mengakses internet, Wi-Fi juga dapat digunakan untuk membuat jaringan tanpa kabel di perusahaan. Karena itu banyak orang mengasosiasikan Wi-Fi dengan kebebasan, karena teknologi Wi-Fi memberikan kebebasan kepada pemakainya untuk mengakses internet atau mentransfer data dari ruang meeting, kamar hotel, kampus, dan café-café yang bertanda Wi-Fi Hot Spot.

2.6.2 Perkembangan Sejarah Wi-fi

Wireless merupakan teknologi yang bertujuan untuk menggantikan kabel yang menghubungkan terminal komputer dengan jaringan, dengan begitu komputer dapat berpindah dengan bebas dan tetap dapat berkomunikasi dalam jaringan dengan kecepatan transmisi yang memadai.

Sebuah jaringan wireless dibangun dengan standar 802.11 yg dirilis Institute of Electrical and Electronics Engineers (IEEE). Tahun 1991, Wi-Fi dikembangkan oleh CNR Corporation/AT&T yang kemudian namanya berubah menjadi Lucent * Agere System, sebuah perusahaan yang berdiri di Niewegien, Belanda. Dulu kecepatan transfer datanya hanya 1-2 Mbps. Vic Hayes, si penemu Wi-Fi, yang akhirnya disebut sebagai "Bapak Wi-Fi" merancang beberapa standar Wi-Fi, yaitu IEEE 802.11a, 802.11b dan 802.11g. Tahun 2003, Vic pensiun dari Agere Systems yang akhirnya kalah berkompetisi. Chipset all in one 802.11abg buatan Agere Systems yang diberi nama kode WARP tidak pernah menembus pasaran.

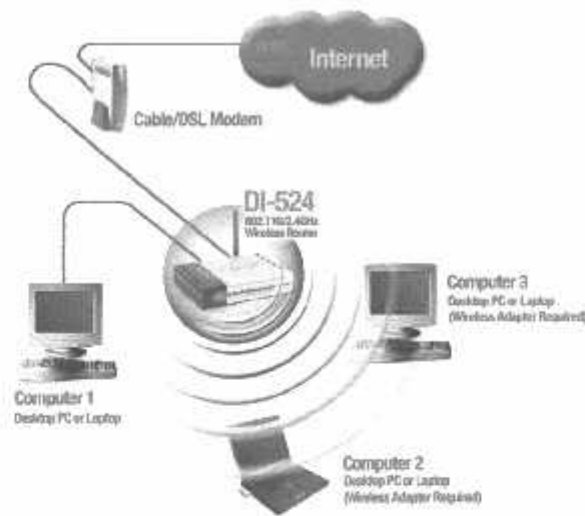
Akhirnya tahun 2004, Agere Systems mundur dari pasar Wi-Fi. RAGAM STANDAR WI-FI Di tahun 1997, IEEE (Institute of Electrical and Electronics Engineers) yang anggotanya terdiri dari insinyur dan perusahaan-perusahaan komputer dan jaringan besar - di antaranya Cisco, Microsoft, dan Apple - yang berkompeten dalam mengembangkan berbagai standar dalam industri komputer dan elektronik, mengadopsi sebuah standar baru untuk jaringan nirkabel, namanya standar 802.11. Standar tersebut dibuat untuk mendukung akses LAN (Local Area Network) tanpa kabel. Ada beragam standar 802.11 yang diterapkan oleh IEEE. Untuk koneksi Wi-Fi, antar perangkat, ada tiga jenis jaringan nirkabel yang ditetapkan oleh IEEE. Mereka adalah standar 802.11a, 802.11b, dan 802.11g. Yang membedakan masing-masing standar adalah jangkauan frekuensi dan kecepatan transfernya.

Kebanyakan perangkat yang dijual saat ini bernama standar 802.11b, meskipun kecepatan transfer datanya lebih lambat ketimbang kecepatan transfer data pada jaringan 802.11a. Pasalnya, kebanyakan instalasi Wi-Fi saat ini dibangun menggunakan standar 802.11b. Sebagai informasi, standar IEEE 802.11a beroperasi pada frekuensi 5 GHz. Kecepatan transfer yang ditawarkannya adalah maksimum 54 MBps (Mega Bit per second). Standar IEEE 802.11b beroperasi pada frekuensi 2,4 GHz dan menawarkan kecepatan transfer data maksimum hingga 11 MBps. Sedang yang terakhir, standar 802.11g, beroperasi pada frekuensi 2,4 GHz dan memiliki kecepatan transfer maksimum hingga 54 MBps. Jika diperhatikan, standar 802.11g merupakan gabungan dari standar 802.11a dan 802.11b. Dengan begitu, pengguna Wi-Fi yang memiliki perangkat yang mendukung standar jaringan 802.11b bisa terhubung pada access point 802.11g, tapi tentunya kecepatan transfer yang bisa dinikmatinya adalah maksimum 11 MBps.

2.6.3 Cara Kerja Wi-fi

Jaringan nirkabel / wireless menggunakan gelombang frekuensi radio yang ditransmisikan dalam spektrum GHz 2,4 GHz sampai 5 yang sangat mirip dengan transmisi sinyal televisi atau ponsel. Ketika komputer terhubung ke jaringan nirkabel, adaptor nirkabel akan mengkodekan informasi menjadi sinyal radio dan kemudian mengirimkan data menggunakan antena wireless yang biasanya terintegrasi dengan laptop. Sinyal nirkabel juga dapat diterima melalui antena eksternal dipasang ke slot

USB komputer. Router WiFi untuk jaringan kemudian menerima informasi ini, decode, dan mengirimkannya ke Internet melalui koneksi, kabel berkecepatan tinggi.



Gambar 2.6. cara kerja Wifi

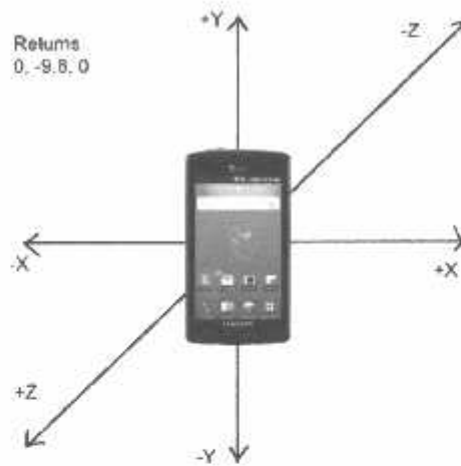
2.7 Sensor Accelometer

2.7.1 Pengertian Sensor Accelometer

Munculnya Android Smartphone yang telah dilengkapi dengan berbagai macam sensor memungkinkan kita untuk membuat sendiri sebuah Handled Instrument yang multi fungsi, ditambah lagi dengan kemampuannya dalam akses internet memungkinkan kita untuk menciptakan sendiri sebuah sistem alat ukur dan kontrol yang dapat di monitor ataupun di kendalikan dari jarak jauh melalui jaringan internet ataupun bluetooth.

Sensor Accelerometer adalah sebuah sensor yang dapat mengukur percepatan, mendeteksi dan mengukur getaran (vibrasi), dan mengukur percepatan akibat gravitasi. Contoh pemanfaatan aplikasi di android sendiri saat ini yang banyak dikembangkan adalah untuk mendeteksi posisi suatu smartphone landscape atau vertical dengan otomatis perubahan pada layar, dan juga gerakan tangan kita sebagai alat konsol untuk bermain game dan masih banyak kegunaan untuk pemanfaatan yang berkaitan dengan kontrol.

Berikut contoh accelerometer sensor aplikasi sederhana di android, yaitu aplikasi untuk menampilkan kordinat sumbu 3 axis accelerometer, yaitu sumbu X , Y, dan Z, untuk ilustrasi 3 axis sumbu di handset android seperti gambar dibawah Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu.



Gambar 2.7 Accelometer

Bertambahnya suatu kecepatan dalam suatu rentang waktu disebut juga percepatan (acceleration). Jika kecepatan semakin berkurang daripada kecepatan sebelumnya, disebut deceleration. Percepatan juga bergantung pada arah/orientasi karena merupakan penurunan kecepatan yang merupakan besaran vektor. Berubahnya arah pergerakan suatu benda akan menimbulkan percepatan pula.

2.7.2 Prinsip kerja Sensor Acelometer

Prinsip kerja dari transduser ini berdasarkan hukum fisika bahwa apabila suatu konduktor digerakkan melalui suatu medan magnet, atau jika suatu medan magnet digerakkan melalui suatu konduktor, maka akan timbul suatu tegangan induksi pada konduktor tersebut. Accelerometer yang diletakan di permukaan bumi dapat mendeteksi percepatan $1g$ (ukuran gravitasi bumi) pada titik vertikalnya, untuk percepatan yang dikarenakan oleh pergerakan horizontal maka accelerometer akan mengukur percepatannya secara langsung ketika bergerak secara horizontal. Hal ini sesuai dengan tipe dan jenis sensor Accelerometer yang digunakan karena setiap jenis sensor berbeda-beda sesuai dengan spesifikasi yang dikeluarkan oleh perusahaan pembuatnya. Saat ini hamper semua sensor/transduser accelerometer sudah dalam bentuk digital (bukan

dengan sistem mekanik) sehingga cara kerjanya hanya berdasarkan temperatur yang diolah secara digital dalam satu chip.

2.8 IP Camera

2.8.1 Pengertian IP Camera

IP CAMERA (Internet Protocol Camera) adalah kamera video digital memiliki kemampuan untuk mengirim dan menerima data via jaringan baik lokal maupun internet. Pada umumnya IP Camera yang di gunakan saat ini dapat menghasilkan dua jenis data, yaitu gambar dalam bentuk JPEG (*Joint Photographic Experts Group*) adalah format gambar yang dapat menyimpan gambar-gambar dengan ukuran lebih kecil dan video dalam bentuk MJPEG (*Motion Joint Photographic Experts Group*). Menurut Wikipedia secara umum, ada dua jenis IP Camera :

1. Centralized IP camera IP camera jenis ini memerlukan Network Video Recorder (NVR) untuk meng-handle perekaman, video dan manajemen alarm.
2. Decentralized IP camera jenis IP camera ini telah memiliki fungsi built-in yang dapat merekam langsung ke media penyimpanan digital seperti flashdisk, harddisk atau media penyimpanan yang terpasang di jaringan



Gambar 2.8 IP Camera

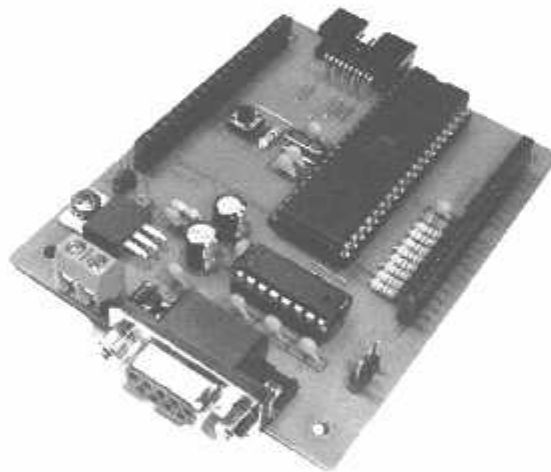
Bagian - bagian pada IP Camera:

1. Lensa, fungsinya untuk memfokuskan gambar.
 2. Sensor gambar (CCD atau CMOS) : digunakan untuk merubah cahaya ke sinyal listrik.
-

3. Processor pengolah gambar dan kompresi gambar, agar data tidak terlalu besar maka perlu di kompresi.
4. Microcomputer dan ethernet , mengontrol sistem dan menghubungkan ke jaringan komputer.
5. Input Output port, berfungsi untuk mengontrol lensa (fokus, zoom), menggerakkan arah kamera, menggerakkan relay dsb.
6. Input Audio/ suara.

2.9 Sistem Minimum Mikrokontroler

Sistem minimum mikrokontroler adalah sistem elektronika yang terdiri dari komponen-komponen dasar yang dibutuhkan oleh suatu mikrokontroler atau rangkaian minimal dimana chip mikrokontroler dapat bekerja dan berfungsi dengan baik. Pada umumnya, suatu mikrokontroler membutuhkan dua elemen (selain power supply) yang berfungsi: Kristal Oscillator (XTAL) sebagai pemompa data, dan Rangkaian RESET untuk membuat mikrokontroler memulai kembali pembacaan program, hal tersebut dibutuhkan pada saat mikrokontroler mengalami gangguan dalam meng-eksekusi program.



Gambar 2.9 Minimum Sistem

2.10 Motor DC

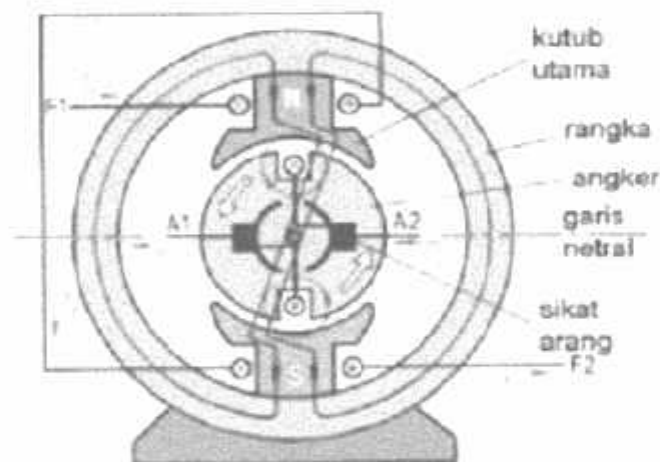
Motor DC merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya. Dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik

maka arah putaran motor akan terbalik pula. Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor sedangkan besar dari beda tegangan pada kedua terminal menentukan kecepatan motor.

Motor DC memiliki 2 bagian dasar :

1. Bagian yang tetap/stasioner yang disebut stator. Stator ini menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektro magnet) ataupun magnet permanen.
2. Bagian yang berputar disebut rotor. Rotor ini berupa sebuah koil dimana arus listrik mengalir.

Gaya elektromagnet pada motor DC timbul saat ada arus yang mengalir pada penghantar yang berada dalam medan magnet. Medan magnet itu sendiri ditimbulkan oleh magnet permanen. Garis-garis gaya magnet mengalir diantara dua kutub magnet dari kutub utara ke kutub selatan. Menurut hukum gaya Lorentz, arus yang mengalir pada penghantar yang terletak dalam medan magnet akan menimbulkan gaya. Gaya F , timbul tergantung pada arah arus I , dan arah medan magnet B .

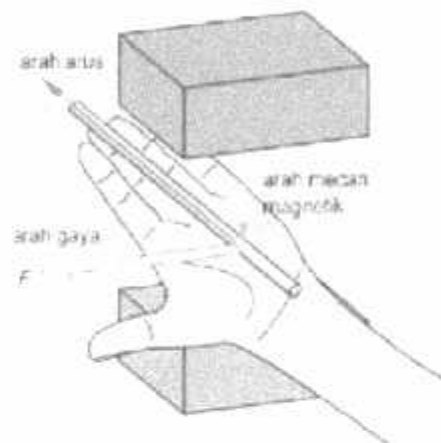


Gambar 2.10 Konstruksi Motor DC

Belitan stator merupakan elektromagnet, dengan penguat magnet terpisah F1-F2. Belitan jangkar ditopang oleh poros dengan ujung-ujungnya terhubung ke komutator dan sikat arang A1-A2. Arus listrik DC pada penguat magnet mengalir dari F1 menuju F2 menghasilkan medan magnet yang memotong belitan jangkar. Belitan jangkar

diberikan listrik DC dari A2 menuju ke A1. Sesuai kaidah tangan kiri jangkar akan berputar berlawanan jarum jam.

Gaya elektromagnet pada motor DC timbul saat ada arus yang mengalir pada penghantar yang berada dalam medan magnet. Medan magnet itu sendiri ditimbulkan oleh magnet permanen. Garis-garis gaya magnet mengalir diantara dua kutub magnet dari kutub utara ke kutub selatan. Menurut hukum gaya Lorentz, arus yang mengalir pada penghantar yang terletak dalam medan magnet akan menimbulkan gaya. Gaya F , timbul tergantung pada arah arus I , dan arah medan magnet B . Arah gaya F dapat ditentukan dengan aturan tangan kiri seperti pada gambar berikut.



Gambar 2.11 Penentuan arah gaya pada kawat berarus listrik dalam medan magnet



Gambar 2.12 Contoh Motor DC



BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Sistem

Aplikasi yang dibuat diharapkan memiliki kemampuan untuk mengontrol dan memonitoring sebuah robot beroda dari jarak jauh yang dilengkapi dengan IP Camera. Dengan memanfaatkan sensor accelerometer dan komunikasi Bluetooth pada *smartphone android* sehingga robot beroda yang dipakai diharapkan dapat bekerja dan bergerak sesuai dengan perintah.

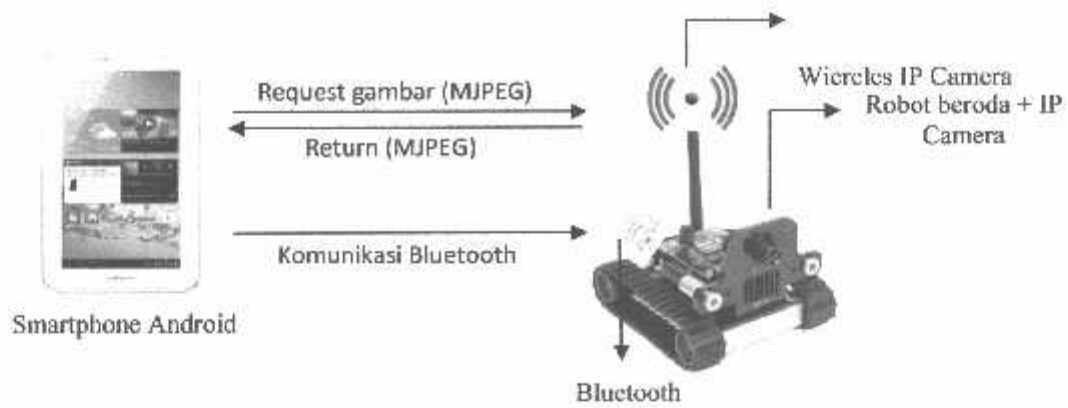
3.1.1 Deskripsi Umum Sistem

Sistem aplikasi kontrol dan monitoring robot berbasis android adalah sebuah aplikasi untuk memudahkan pengontrolan dan monitoring sebuah robot beroda dari jarak jauh dengan memanfaatkan sensor accelerometer dan komunikasi Bluetooth yang terdapat pada *smartphone android*.

Sensor accelerometer yang terdapat pada *smartphone android* di manfaatkan sebagai pengontrolan robot, dimana data yang dihasilkan sensor accelerometer berupa koordinat 3D yang nantinya di konversikan menjadi arah gerak dalam bentuk karakter oleh android. Karakter inilah yang akan di kirim ke mikrokontroler yang terdapat pada robot melalui komunikasi Bluetooth. Setelah data di terima oleh mikrokontroler dan diterjemahkan sesuai dengan library yang terdapat pada mikrokontroler. Ketika data yang diterima sesuai dengan library yang ada, mikrokontroler akan langsung melakukan eksekusi sesuai dengan perintah..

Aplikasi monitoring robot beroda yang di lengkapi IP Camera ini juga menggunakan komunikasi wiereless, bahasa pemerograman yang di pakai dalam pembuatan aplikasi ini adalah bahasa pemerograman java android dan XML. Pada aplikasi monitoring ini *smartphone* akan membaca wifi melalui IP Camera, ketika android sudah dapat membaca wifi pada IP Camera, *smartphone* akan langsung mengirimkan request gambar/*video streaming* (MJPEG). Kemudian setelah IP Camera menerima request (MJPEG) maka akan langsung mengirimkan permintaan gambar/*streaming video* menuju *smartphone android* untuk ditampilkan.

Berikut adalah desain gambaran umum system aplikasi kendali dan monitoring robot berbasis android, yang ditunjukkan pada gambar 3.1 berikut ini.



Gambar 3.1 Diagram Umum Sistem

Dari gambar 3.1 dapat dijelaskan secara umum dan fungsi dari masing-masing blok sebagai berikut :

1. **Smartphone Android** berfungsi sebagai pengontrol dan monitoring robot yang sudah mempunyai bluetooth, wireless dan sensor accelerometer yang sudah dibutuhkan dalam kontrol dan monitoring robot.
2. **Minimum system Mikrokontroler ATmega32** berfungsi untuk program mikrokontroler dan mengkonversi nilai dari sensor accelerometer menjadi nilai sebuah sudut, kemudian menentukan arah robot sesuai dengan perintah.
3. **Sensor Accelometer** berfungsi untuk mengukur percepatan gerakan robot.
4. **Bluetooth** berfungsi untuk mengirimkan data pergerakan sensor accelerometer dari Smartphone Android ke Mikrokontroler ATmega32 yang sudah terpasang Bluetooth slave.
5. **Wireless** berfungsi sebagai komunikasi antara IP Camera dengan Smartphone Android dan mengirimkan streaming video.
6. **Motor DC** berfungsi berfungsi sebagai mekanik robot, untuk menggerakkan maju, mundur, belok kiri, belok kanan dan menentukan suatu posisi atau arah robot.

3.1.2 Analisa Kebutuhan

Dalam pembuatan aplikasi sistem kendalai dan monitoring, akan di jelaskan mengenai alat dan kebutuhan yang menunjang pembuatan aplikasi baik secara perangkat lunak maupun perangkat keras.

3.1.2.1 Smartphone yang digunakan

Smartphone yang di gunakan dalam pembuatan aplikasi kendali dan monitoring robot yaitu menggunakan Samsung Galaxy Tab 2 7.0. Dilengkapi prosesor dan system oprasi Android versi 4.0.3 (Ice Cream Sandwich), dapat di upgrade dengan versi 4.1.3 (Jelly Bean) dengan tampilan dan kinerja menjadi lebih baik. Saat mengujinya dengan menyambungkan ke jaringan Hotspot Wifi dan 3G proses transfer data berlangsung cepat. Proses pengiriman pesan singkat dan email dapat dilakukan dengan lancar..Berikut spesifikasi Samsung Galaxy Tab 2 7.0 :

- Produsen : Samsung, Korea
- Sistemoperasi : Android 4.0 (Ice Cream Sandwich) upgrade ke 4.1(Jelly Bean)
- Prosesor : Dual core, 1200 MHz
- Jaringan/ Data : GPRS, EDGE, 3G
- Dimensi dan berat : 193.7 x 122.4 x 10.5 mm ; 344g
- Layar
 - Tipe : PLS LCD capacitive touchscreen, 16M colors
 - Ukuran : 600 x 1024 pixels, 7.0 inches
- Memori
 - Internal : 16 GB storage, 1GB ram
 - Eksternal : microSD, up to 64 GB
- Kamera : 3,15 MP, 2048x1536 pixels, autofocus/VGA Front
- Baterai : Standard battery, Li-Ion 4000mAh



Gambar 3.2 Samsung Galaxy Tab 2 7.0

3.1.2.2 Minimum Sistem yang digunakan

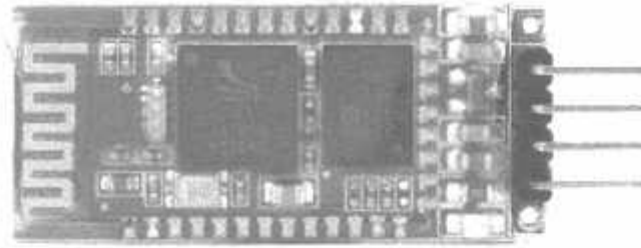
Minimum system yang di pilih untuk modul elektronika yang digunakan untuk pemrograman mikrokontroler pada robot beroda yaitu system minimum mikrokontroler ATMEGA32 AVR, berikut spesifikasi :

- Menggunakan XTAL 11.0592MHz.
- Dilengkapi rangkaian regulator 5V dan dioda pengaman dengan konektor DC yang mudah dihubungkan dengan Adaptor-DC.
- Dilengkapi rangkaian antarmuka (*interface*) Max232 sehingga dapat langsung dihubungkan pada PORT SERIAL / COM PORT komputer.
- Koneksi ADC sudah disiapkan (AVCC, AGND, dan AREF) sehingga sistem sudah siap untuk menerima input ANALOG pada PORTA.
- Tersedia Array LED pada PORTC, dan Push-ON pada PORTD.2 dan PORTD.3 sehingga cocok untuk latihan pemrograman atau pengecekan program (*DEBUG*).
- ISP Flash : 32 KB
- SRAM : 2 KB
- EEPROM : 1024 bytes
- ADC : 8 channel 10-bit

3.1.2.3 Bluetooth slave yang digunakan

Bluetooth slave yang digunakan adalah tipe HC-05 merupakan produk JY-MCU. Berikut spesifikasi Bluetooth slave :

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
 - Frequency: 2.4GHz ISM band
 - Modulation: GFSK(Gaussian Frequency Shift Keying)
 - Security: Authentication and encryption
 - Profiles: Bluetooth serial port
 - Power supply: +3.3VDC 50mA
 - Working temperature: -20°C ~ +75°C
 - Dimension: 26.9mm x 13mm x 2.2 mm
 - Easy adaption on SeeeduinoUartSB v2.1
-



Gambar 3.3 Bluetooth Slave

3.1.2.4 Perangkat Lunak yang digunakan

a. Software IDE Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent).

Berikut ini adalah sifat dari Eclipse:

1. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. Multi-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan open source. Open source berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut plug-in.

b. Java JDK

Java Development Kit (JDK) adalah Sun Microsystems produk ditujukan untuk pengembang Java. Sejak diperkenalkannya Java, telah jauh SDK Java yang paling banyak digunakan. Pada tanggal 17 November 2006, Sun mengumumkan bahwa akan dirilis di bawah GNU General Public License (GPL), sehingga membuat perangkat

lunak bebas. Hal ini terjadi sebagian besar pada tanggal 8 Mei 2007 [3]; Sun kontribusi kode sumber untuk JDKOpen.

c. Android SDK

Android SDK adalah development kit yang ditargetkan pada perangkat mobile yang menjalankan sistem operasi Android. Dengan Android SDK, Anda dapat mengembangkan, melakukan debug, dan uji aplikasi menggunakan emulator built-in yang mengemulasi OS Android di banyak platform (Linux, Windows, dll).

d. IP Camera

Dalam pembuatan aplikasi monitoring robot berbasis Smartphone Android ini menggunakan IP Camera TL-SC3130G. TP-Link TL-SC3130 ini dilengkapi built-in microphone, speaker eksternal, dan kemampuan audio yang sangat baik, TL-SC3130 juga mampu membuat transmisi real-time audio komunikasi dua arah. Jadi, selain pemantauan video, dapat mendengar suara yang terjadi dalam area yang tercover oleh jangkauan microphone kamera. TL-SC3130 dapat memberikan hasil MPEG-4 dan MJPEG stream secara bersamaan, dengan TL-SC3130 dapat memonitor ruangan dalam jaringan local maupun ruangan yang terhubung dengan jaringan internet melalui video streaming. berikut spesifikasinya :

- Sensor gambar : 1/4 " Progressive scan CMOS sensor
 - Lensa : F: 2.0, f: 4.0mm
 - Max. Resolusi : Up to 30(NTSC) / 25(PAL) fps at 640x480, 320x240, 160x120
 - Viewing Angle : Diagonal 67°, Horizontal 53°, Vertical 40°
 - Digital Zoom : 10x Digital
 - Kompresi Video : Motion JPEG;MPEG-4
 - Kompresi Audio : G.711 PCM, 8 kHz, 64 kbit/s
 - Kesesuaian protocol : TCP, TCP/IP, HTTP
 - Kesesuaian Sistem Operasi : Windows 98/ME/2000/2003/XP/Vista/7, Mac OS Leopard 10.5
 - Dimensi : 3.7 x 2.7 x 1.2 in. (96 x 58 x 31mm)
 - Video Streaming : Simultaneous Motion JPEG and MPEG-4 (Dual streaming).
-



Gambar 3.4 IP Camera TL-SC3130G

3.1.2.5 Robot Yang Digunakan

Pada pengujian aplikasi kendali dan monitoring berbasis smartphone android ini, juga membutuhkan robot yang telah di pasang IP Camera, dapat di jelaskan spesifikasi robot yang digunakan pada saat pengujian yaitu:

A. Spesifikasi Robot

- Baterai Ly-poly 2100 mAH 11.1 V
- Mikrokontroler Atmega 32
- Board Minimum sistem yang kompatibel dengan Mega 8535,16 dan 32
- Motor DC dengan gearbox (Geared)
- Driver Motor kit EMS 2A Dual H-Bridge
- rangka mekanik Alumunium 90 %
- sistem mekanik penggerak dengan 2 Roda Aktuator Aktif dan 1 roda pasif.
- kecepatan motor diatur dengan metode PWM (Pulse Width Modulation)
- LCD 16x2

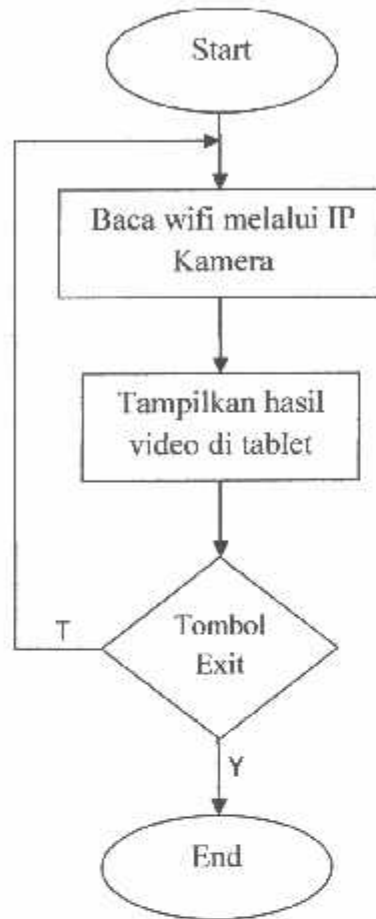
3.2 Pemodelan Aplikasi

Sebelum pembuatan aplikasi, terlebih dahulu harus membuat gambaran alur kerja aplikasi sesuai dengan fungsinya. Alur kerja aplikasi dapat digambarkan dalam bentuk diagram alir.

Dalam perancangan aplikasi ini dibutuhkan 2 buah aplikasi yaitu aplikasi kendali robot dan monitoring robot berbasis android. Aplikasi yang digunakan untuk kendali dan monitoring robot dibuat menggunakan bahasa pemrograman java android. Kedua aplikasi ini menggunakan dua komunikasi, untuk aplikasi kendali robot ini menggunakan komunikasi bluetooth dan untuk monitoring robot menggunakan

komunikasi wiereles. Berikut adalah diagram alir program yang dibutuhkan:

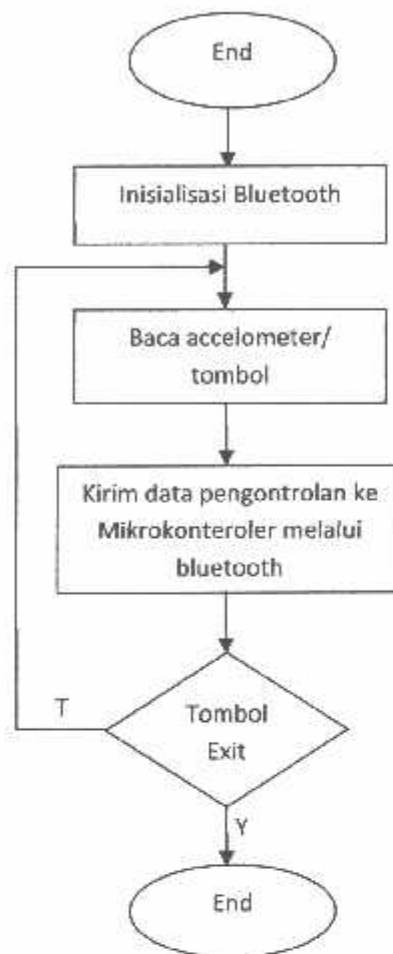
3.2.1 Diagram Alir Monitoring Robot



Gambar 3.5 *Flowchart* Monitoring.

Gambar 3.6 merupakan diagram alir program, untuk aplikasi monitoring robot, bahasa pemrograman yang di pakai dalam pembuatan aplikasi adalah bahasa pemrograman java android. Alur perogram pertama kali pada aplikasi monitoring robot adalah Smartphone android akan membaca wifi melalui IP Camera, ketika smartphone android tersebut telah berhasil membaca wifi pada IP Camera, maka smartphone akan langsung mengirimkan request gambar/video streaming (MJPEG). Kemudian setelah IP Camera menerima request (MJPEG), akan langsung mengirimkan permintaan gambar/streaming video menuju smartphone android untuk di tampilkan sebagai monitoring robot. Setelah itu proses pengiriman streaming video akan terhenti jika aplikasi tidak lagi dijalankan.

3.2.2 Diagram Alir Kendali Robot



Gambar 3.6 *Flowchart* Kontrol Robot

Gambar 3.7 merupakan diagram alir program aplikasi kendali robot, bahasa pemrograman yang digunakan dalam pembuatan aplikasi ini adalah java android. Alur perogram pertama kali pada aplikasi kendali robot adalah melakukan penginisialisasian Bluetooth, setelah inisialisasi bluetooth tersebut berhasil selanjutnya sensor accelerometer yang terdapat pada smartphone android di manfaatkan sebagai kendali robot , dimana data yang dihasilkan sensor accelerometer berupa koordinat 3D (x,y,z) yang nantinya di konversikan menjadi arah gerak. Hasil dari konversi koordinat accelerometer tersebut langsung di kirim ke mikrokontroler melalui komunikasi Bluetooth, setelah data di terima oleh mikrokontroler dan diterjemahkan sesuai dengan library yang terdapat pada mikrokontroler, maka mikrokontroler akan langsung melakukan eksekusi sesuai dengan perintah atau data yang diterima.

3.3 Perancangan Dan Desain Aplikasi

Sistem yang akan dibuat dalam pengontrol robot ini adalah smartphone android sebagai kontrol utama , komunikasi *bluetooth* sebagai penghubung antara smartphone berbasis Android dan *mobile robot*, serta *sensor accelerometer* sebagai pengontrol gerakan *mobile robot* secara *real time*.

Pengontrolan gerakan *mobile robot* dilakukan dengan cara mengirimkan perubahan nilai *sensor accelerometer* kepada *mobile robot* melalui komunikasi serial. Untuk memonitoring pada robot, komunikasi *wiereless* sebagai penghubung antara smartphone android dengan IP kamera yang terpasang pada robot beroda.

Dengan menggunakan aplikasi berbasis android ini dapat dilakukan monitoring dan kendali dari jarak jauh hanya dengan menggunakan dua komunikasi yakni *bluetooth* dan *wifi*.

Beberapa fasilitas yang dimiliki oleh aplikasi ini antara lain :

1. Aplikasi ini dapat digunakan untuk memonitoring dan kendali dari jarak jauh melalui smartphone android
2. Aplikasi ini dapat digunakan untuk memantau keberadaan robot walaupun tidak terlihat langsung oleh pengguna.
3. Aplikasi ini dapat digunakan sebagai kamera pemantau yang bisa mengawasi ruangan dari jarak jauh.

Aplikasi Pengendalian Dan Monitoring Robot Yang Di Lengkapi Ip Camera Berbasis Android ini dibagi dalam tiga tahap, yaitu perancangan software, perancangan antarmuka aplikasi, dan konfigurasi IP Camera. Aplikasi ini dirancang untuk bisa memantau dan kendali robot dari jarak jauh.

3.3.1 Perancangan Software

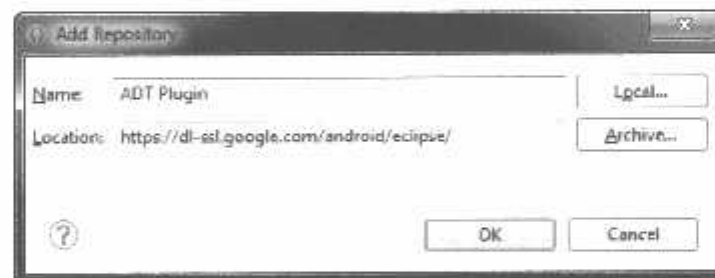
Dalam pembuatan aplikasi ini menggunakan program eclipse, Java JDK, ADT plugin untuk eclipse dan emulator android yaitu SDK. Langkah pertama adalah menginstal java JDK.



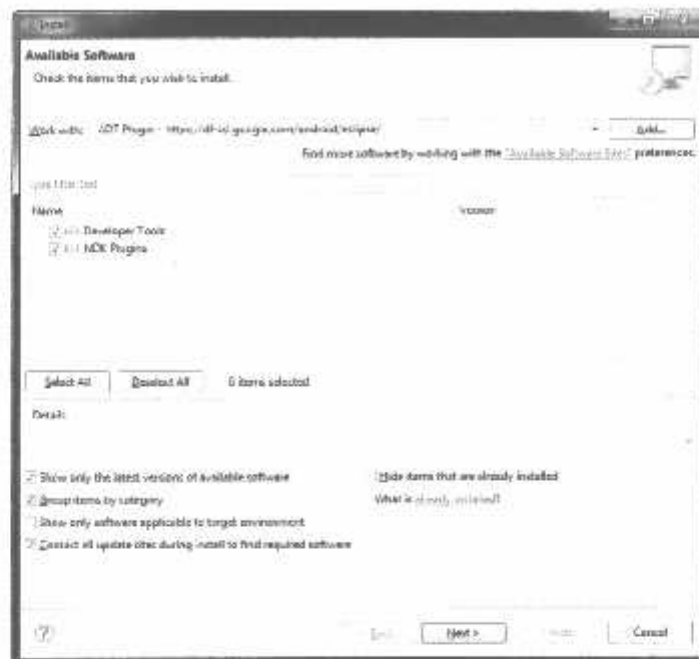
Gambar 3.7 Konfigurasi java JDK

Pada gambar 3.4 konfigurasi java JDK yang sebelumnya sudah di instal pada komputer. Selanjutnya klik kanan Computer > Propertise > Advance system setting > Advance > Environment Variable, pada menu User variables klik New pada Variable name isi dengan PATH dan pada variable value isikan alamat java JDK yang telah terinstal.

Selanjutnya menambahkan ADT plug in pada Eclipse, Program Eclipse bisa di unduh langsung pada situs resminya. Software Eclipse bersifat portable sehingga bisa langsung dijalankan. Extract dan jalankan IDE Eclipse pilih menu Help > Instal New Software, pada windows instalasi klik Add, pada dialog Add Repository isikan Name : ADT Plugin, Location : <https://dl-ssl.google.com/android/eclipse/> kemudian OK, seperti yang terlihat pada gambar 3.5



Gambar 3.8 Add Repository ADT Plugin



Gambar 3.9 Instal NDK Plugins dan Developer tool

Pada gambar 3.6 window **Install – Available Software**, centang **Developer Tool** dan **NDK Plugins**, atau bisa juga dengan mengklik tombol **Select All**. Selanjutnya klik **Next**.

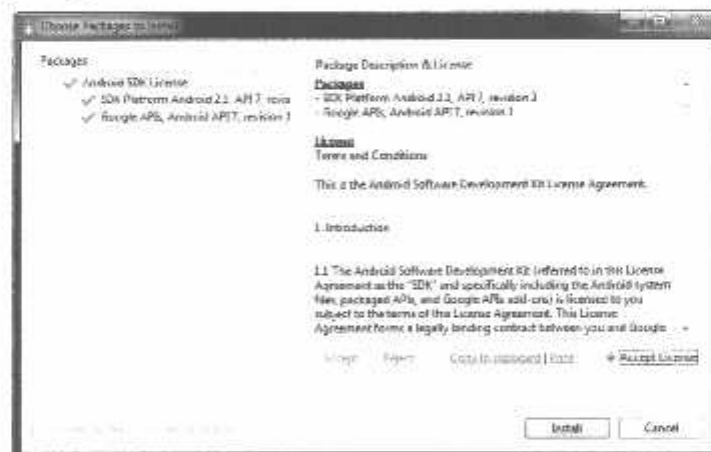
Pada tampilan window berikutnya, akan muncul daftar tool yang akan di-download. Pilih **Next** untuk melanjutkan proses instalasi. Selanjutnya akan muncul **license Agreement**, pilih **Accept** kemudian klik **Finish**. Setelah selesai instalasi, restart IDE Eclipse.

Selanjutnya adalah instalasi paket pada android SDK. Jalankan SDK Manager, maka akan muncul window **Android SDK Manager** seperti gambar 3.7 Di bawah ini, pilih versi android yang ingin di instal,



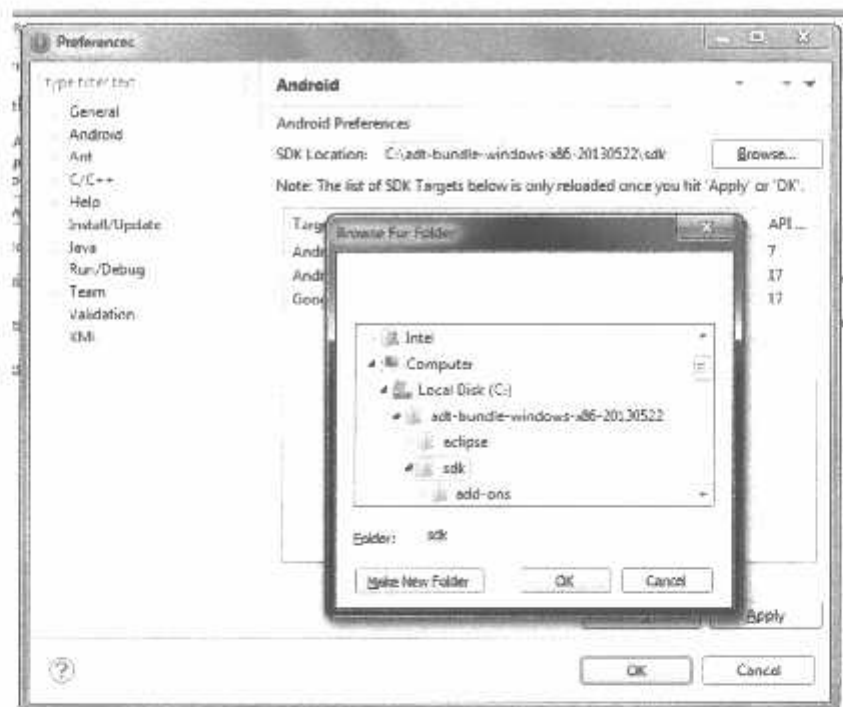
Gambar 3.10 Instal paket android version

Setelah memilih versi android yang akan di instal, klik tombol instal packages, selanjutnya muncul jendela Choose Packages to instal, centang Accept all dan klik tombol Instal seperti gambar 38. Di bawah ini.



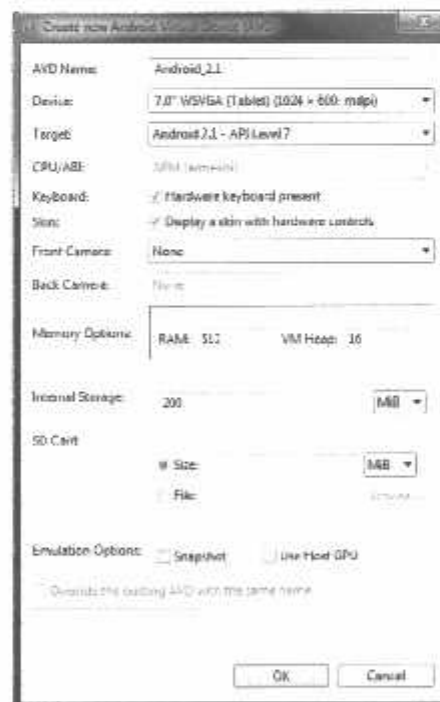
Gambar 3.11 Packages to instal

Langkah berikutnya adalah konfigurasi android SDK pada eclipse. Jalankan program eclipse pilih meni Windows > Preferences, pilih list android. Klik tombol Browse, kemudian pilih directory SDK Manager (Folder SDK) kemudian klik OK.



Gambar 3.12 Konfigurasi Android SDK pada eclipse

Selanjutnya konfigurasi Android Virtual Device Manager. Klik menu Window > Android Virtual Device Manager > New, maka tampil window Create New Android Virtual Device (AVD). Lalu konfigurasi sesuai dengan gambar 3.10 Di bawah ini



Gambar 3.13 konfigurasi Android Virtual Device Manager

Setelah konfigurasi selesai klik OK. Dan pada windows AVD Manager, akan muncul AVD Name yang di buat tadi lalu klik start untuk menjalankan emulator, seperti

yang terlihat pada gambar 3. Di bawah ini

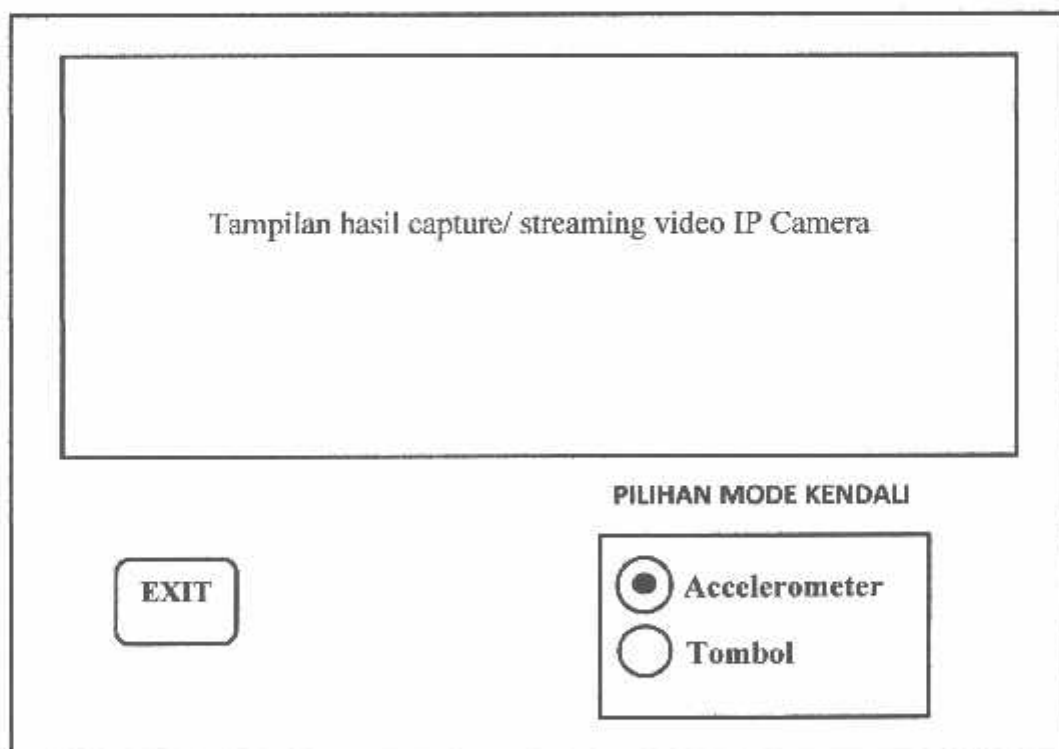


Gambar 3.14 emulator android

3.3.2 Perancangan Antarmuka Aplikasi

3.3.2.1 Rancangan Desain Tampilan Utama

Pada tampilan awal aplikasi kendali dan monitoring berbasis android ini, terdapat dua pilihan mode kendali yaitu pilihan pertama menggunakan mode kendali menggunakan accelerometer dan pilihan kedua mode kendali menggunakan tombol, berikut adalah desain tampilannya.

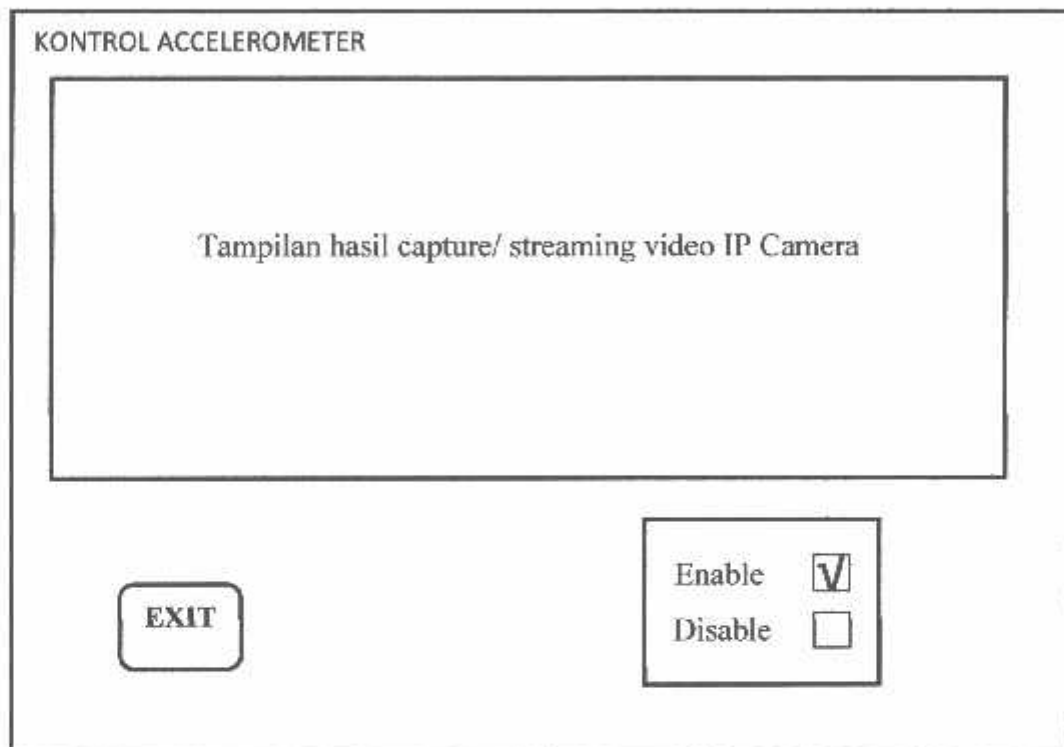


Gambar 3.15 Tampilan awal aplikasi

Gambar diatas adalah desain awal pembuatan tampilan utama aplikasi kendali dan monitoring pada smartphone android. Bagian atasnya merupakan tampilan

streaming video sebagai monitoring. Pada bagian bawahnya, di sebelah kiri terdapat menu exit untuk keluar dari aplikasi. Sedangkan bagian kanannya terdapat kolom pilihan mode kontrol yaitu menggunakan accelerometer dan tombol.

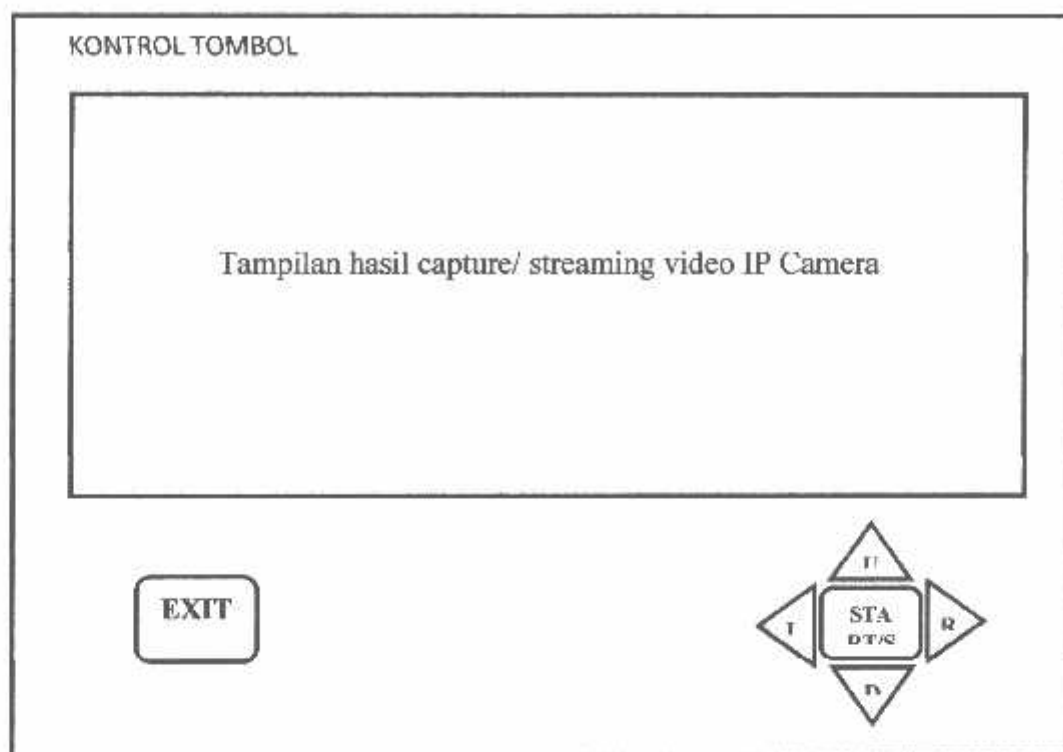
3.3.2.2 Rancangan Desain Mode Kontrol (Accelerometer)



Gambar 3.16 Tampilan kontrol Accelerometer

Gambar diatas adalah rancangan desain tampilan pilihan mode kontrol menggunakan accelerometer. Pada bagian atasnya merupakan tampilan streaming video sebagai monitoring. Pada bagian bawah kiri terdapat tombol exit, sedangkan bagian bawah kanan terdapat pilihan enable dan disable untuk mengaktifkan aplikasi kendali.

3.3.2.2 Rancangan Desain Mode Kontrol (Tombol)



Gambar 3.17 Tampilan kontrol Tombol

Gambar diatas adalah rancangan desain tampilan pilihan mode kontrol menggunakan tombol. Pada bagian atasnya merupakan tampilan streaming video sebagai monitoring. Pada bagian bawah kiri terdapat tombol exit, sedangkan bagian bawah kanan terdapat pilihan enable dan disable untuk mengaktifkan aplikasi kendali.

3.3.3 Konfigurasi IP Camera

Untuk dapat mengakses gambar/streaming video pada IP Camera, terlebih dahulu melakukan konfigurasi IP Camera, berikut langkah-langkah konfigurasi IP Camera :

1. Menghubungkan IP Camera dengan Laptop menggunakan kabel LAN
2. Buka web browser, lalu ketikkan alamat IP 192.168.1.134 yang merupakan IP default dari IP Camera.
3. Ketikkan MENU > SETTING > BASIC > NETWORK > WIRELESS, setelah itu nyalakan wifi.
4. Connect kan IP Camera dengan Aces Point, setelah connect pilih Aces Point yang akan di hubungkan dengan IP Camera. Jika Accs Point terdapat

password maka ketikkan password, kemudian klik “OK”.



Gambar 3.18 Menghubungkan IP Camera dengan Laptop

5. Karena Acces Point menggunakan DHCP, maka IP Camera tersebut akan mendapatkan alamat IP secara DHCP/acak.
6. Sebelum itu connect kan laptop dengan Acces Point, kemudian lepaskan kabel LAN yang tadinya menghubungkan antara Laptop dengan IP Camera, setelah itu buka kembali browser dan ketikkan alamat IP Camera yang di dapatkan dari Acces Point.



Gambar 3.19 Tampilan gambar dari IP Camera



BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Program aplikasi monitoring dan kontroling robot merupakan sebuah sistem yang mampu melakukan pengendalian dan monitoring pergerakan robot dari jarak jauh. Aplikasi ini merupakan pengembangan dari aplikasi pengontrol robot dengan Bluetooth dan sensor accelerometer pada smartphone android. Pengembangan yang dilakukan adalah dengan menambahkan aplikasi monitoring agar nantinya robot dapat dipantau dari jarak jauh setelah keluar dari pemantauan pengendali.

Pada tahap implementasi ini merupakan proses perubahan rancangan dan spesifikasi yang telah disusun sebelumnya menjadi suatu aplikasi yang siap untuk dijalankan. Implementasi pembuatan aplikasi kendali dan monitoring ini yakni sebagai robot pengintai.

Dalam pembuatan aplikasi ini digunakan beberapa pendukung sistem yang terdiri dari software dan hardware:

1. Software

Di dalam pembuatan aplikasi kendali dan monitoring robot ini menggunakan program utama yaitu IDE Eclipse.

2. Hardware

Selain software yang digunakan dalam pembuatan aplikasi ini juga digunakan hardware untuk mengoperasikan sistem ini. Hardware yang digunakan adalah robot beroda empat yang telah di modifikasi sesuai dengan kebutuhan sistem. Modifikasi tersebut meliputi penambahan IP Camera wireless dan bluetooth.

4.2 Pengujian Sistem

Pengujian adalah tahap uji coba terhadap smartphone dengan aplikasi yang dibuat agar dapat di ambil kesimpulan apakah aplikasi sudah mencapai tujuan yang diharapkan atau tidak dan untuk mengetahui kemungkinan – kemungkinan kesalahan yang terjadi pada aplikasi.

4.2.1 Spesifikasi Pengujian

Pada tahap ini di jelaskan spesifikasi kebutuhan yang digunakan saat proses pengujian. Pada Tabel 4.1 dapat dilihat spesifikasi kebutuhan yang digunakan saat

melakukan pengujian aplikasi kendali dan monitoring robot beroda yang di lengkapi IP Camera.

Tabel 4.1 Spesifikasi Kebutuhan Untuk Pengujian

Nama Kebutuhan	Spesifikasi
Mobile Phone	<ul style="list-style-type: none"> - Sistem Oprasi : Android 4.04 - Layar : 4 inci - Prosesor : 1ghz - RAM : 512mb

Berikut tampilan utama aplikasi kendali dan monitoring robot beroda berbasis smartphone android.



Gambar 4.1 Tampilan Utama Aplikasi Kendali Dn Monitoring

Gambar 4.1 merupakan tampilan utama aplikasi kendali dan monitoring robot beroda. Pada tampilan aplikasi ini, pengguna tidak hanya dapat melakukan pengontrolan robot melaikan bisa melakukan monitoring robot dari jarak jauh ketika robot beroda, keluar dari pantauan pengendali. Pada Aplikasi tersebut terdapat dua pilihan mode kendali, yaitu kendali robot dengan accelerometer dan kendali robot menggunakan tombol. Pada tampilan gambar di atas dapat di lihat, di sebelah kiri merupakan tampilan streaming video untuk monitoring pergerakan robot dan di sebelah kiri terdapat tombol (maju,mundur,kanan,kiri dan stop/start). Pada bagian atas kanan terdapat keterangan bahwa mode kendali apa yang digunakan.

4.2.2 Pengujian Bluetooth

Pengujian Bluetooth dilakukan dengan mencoba komunikasi Bluetooth slave ke laptop atau computer. Pengujian dilakukan dengan menggunakan AccessPort.

- Sebelumnya koneksikan terlebih dahulu Bluetooth slave dengan laptop.
- Jalankan AccessPort
- Selanjutnya akan muncul tampilan jendela dari accessport seperti gambar 4.2 dibawah ini.



Gambar 4.2 jendela accessport

- Selanjutnya melakukan konfigurasi dengan klik icon pada pojok kiri atas
- Maka akan muncul jendela seperti gambar 4.3 dibawah ini.



Gambar 4.3 jendela konfigurasi accessport

- Pada gambar 4.2 diatas, pilih General, lalu pada bagian Port pilih COM6, COM6 didapat saat kita melakukan koneksi Bluetooth slave dengan laptop/komputer.

- Untuk baudratanya setting pada nilai 19200 sesuai dengan spesifikasi Bluetooth slavenya.
- Langkah selanjutnya mengirim karakter “aa;” lalu tekan send maka status TX akan menampilkan data yang dikirim. Seperti yang ditunjukkan pada gambar 4.4 dibawah ini.



Gambar 4.4 pengujian pengiriman data melalui Bluetooth

Pada gambar 4.4 di atas nilai RX tidak mengalami perubahan dikarenakan Bluetooth slave hanya menerima data dan tidak ada data balikan yang dikirim.

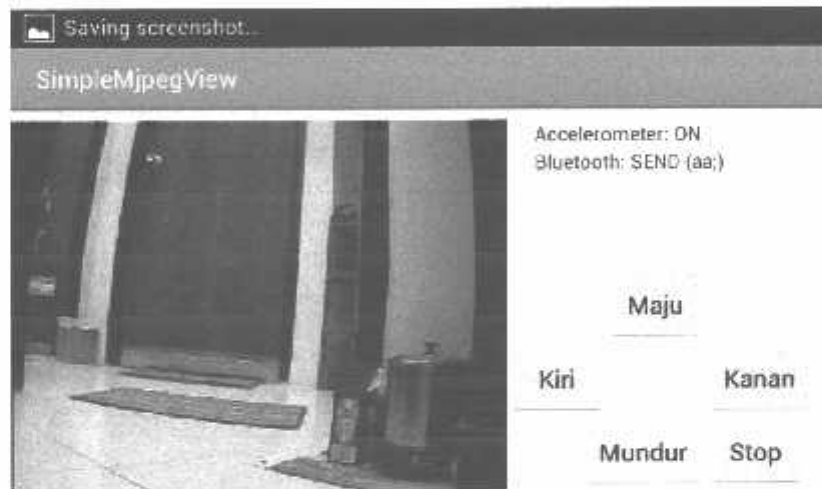
4.2.3 Pengujian Jarak Kontroler Dengan Robot

Pengujian pertama yakni jarak kontroler dengan robot merupakan faktor yang berpengaruh langsung terhadap proses deteksi bluetooth dan wifi pada kendali dan monitoring. Selama jalur komunikasi masih bisa terkoneksi maka proses monitoring dan kendali robot masih bisa dilakukan dengan baik. Pengujian jarak ini bertujuan untuk mengetahui rentang jarak yang optimal agar jalur komunikasi kendali dan monitoring dapat terdeteksi dengan baik. Pengujian dilakukan terhadap 3 rentang jarak, yakni jarak dekat 5 meter, jarak sedang 10 meter dan jarak jauh 15 meter. Pada gambar 4.5 merupakan tampilan yang menunjukkan pengujian pada jarak dekat yaitu ≤ 5 meter, dapat dilihat bahwa proses pengontrolan robot dan hasil monitoring dilakukan dengan baik, karena antara smartphone android dengan robot beroda masih terkoneksi dengan sangat baik.



Gambar 4.5 Pengujian pada jarak dekat

Pengujian berikutnya yakni pada jarak sedang 10 meter. Pada Gambar 4.6, proses pengontrolan robot dan hasil monitoring dapat dijalankan dan masih terkoneksi dengan cukup baik seperti halnya dilakukan pada jarak dekat.



Gambar 4.6 Pengujian pada jarak sedang

Pengujian terakhir pada jarak jauh yakni 15 meter. Pada gambar 4.7 menunjukkan bahwa aplikasi masih dapat dilakukan pengontrolan dengan cukup baik dan menampilkan hasil monitoring atau streaming video kurang jelas, di karenakan pada saat pengujian terhalang tembok sehingga jalur komunikasinya terganggu.



Gambar 4.7 Pengujian pada jarak jauh

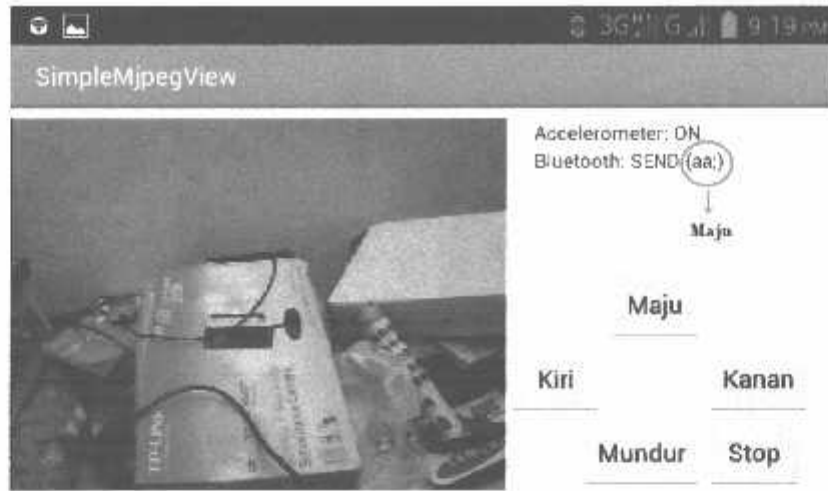
Berikut rangkuman hasil akhir pengujian jarak kontroler dengan robot dapat dilihat pada Tabel 4.2

Tabel 4.2
Tabel Hasil Pengujian Jarak Kontroler Dengan Robot

NO.	Jarak Dari Kontroler	Hasil Monitoring	Hasil Pengontrolan
1	Dekat 5 meter	Streaming video jelas	Sangat Baik
2	Sedang 10 meter	Streaming video jelas	Baik
3	Jauh 15meter	Streaming video kurang jelas	Cukup Baik

4.2.4 Pengujian Arah Gerak Robot

Untuk mengetahui arah gerakan robot beroda, yaitu dengan cara melihat nilai x, y , dan z . Pada Gambar 4.8 dapat dilihat tampilan aplikasi arah gerak robot beroda (maju, belok kanan, belok kiri, stop) pada saat pengujian dilakukan. Pada pengujian arah gerak robot ini bertujuan agar kontroler mengetahui bahwa arah pergerakan robot sesuai dengan data yang di kirim.



Gambar 4.8 Arah Robot Bergerak Maju

Gambar diatas merupakan tampilan arah gerak robot bergerak maju menggunakan pilihan mode control dengan accelerometer. Pada tampilan tersebut terdapat hasil monitoring atau streaming video. Bagian atas kanan merupakan keterangan mode control dengan menggunakan Accelerometer, dibawahnya merupakan keterangan bahwa bluetooth slave mengirimkan karakter "aa;". Karakter "aa;" tersebut merupakan perintah untuk arah robot beroda bergerak maju.



Gambar 4.9 Arah Robot Bergerak Kiri

Gambar diatas merupakan tampilan arah gerak robot bergerak ke kiri menggunakan pilihan mode control dengan accelerometer. Pada tampilan tersebut terdapat hasil monitoring atau streaming video. Bagian atas kanan merupakan keterangan mode control dengan menggunakan Accelerometer, dibawahnya merupakan keterangan

bahwa bluetooth slave mengirimkan karakter “ac;”. Karakter “aa;” tersebut merupakan perintah untuk arah robot beroda bergerak ke kiri.



Gambar 4.10 Arah Robot Bergerak Kanan

Gambar diatas merupakan tampilan arah gerak robot bergerak ke kanan menggunakan pilihan mode control dengan accelerometer. Pada tampilan tersebut terdapat hasil monitoring atau streaming video. Bagian atas kanan merupakan keterangan mode control dengan menggunakan Accelerometer, dibawahnya merupakan keterangan bahwa bluetooth slave mengirimkan karakter “ca;”. Karakter “ca;” tersebut merupakan perintah untuk arah robot beroda bergerak ke kanan.



Gambar 4.11 Robot beroda berhenti /stop

Gambar diatas merupakan tampilan arah gerak robot berhenti bergerak atau stop. Tampilan di atas menggunakan pilihan control dengan accelerometer. Pada tampilan

tersebut tetap menampilkan streaming video akan tetapi pengontrolannya terhenti karena pada tampilan aplikasi terlihat keterangan bahwa bluetooth slave mengirimkan karakter “ bb;”. Karakter “bb;” tersebut merupakan perintah untuk arah robot beroda berhenti bergerak atau stop.

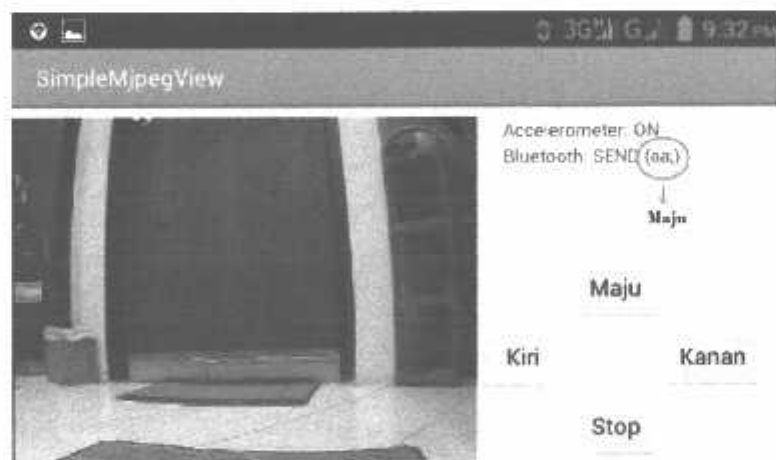
Berikut rangkuman hasil akhir pengujian arah gerak robot, dapat dilihat pada Tabel 4.3.

Tabel 4.3
Pengujian Arah Gerak Robot

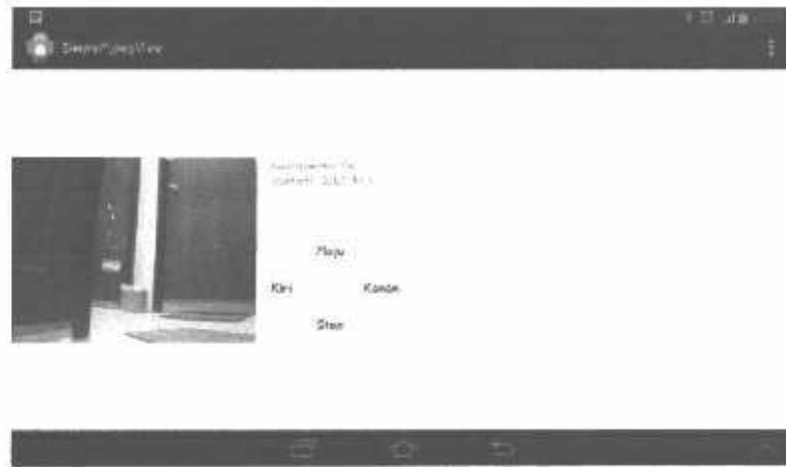
NO.	Perintah	Nilai Sensor Accelerometer	Parameter
1	Maju	X (negatif) ; Y (positif)	Mengirimkan karakter ‘ aa;’
2	Kiri	X (positif) ; Y (negatif)	Mengirimkan karakter ‘ ac;’
3	Kanan	X (positif) ; Y (positif)	Mengirimkan karakter ‘ ca;’
4	Stop	X (negatif) ; Y (negatif)	Mengirimkan karakter ‘ bb;’

4.2.5 Pengujian Pada Smartphone Berbeda

Pada tahap pengujian aplikasi ini dilakukan pada smartphone yang berbeda. Berikut hasil dari pengujian serta tampilan pengujian aplikasi. Tampilan pengujian aplikasi pada smartphone dengan layar 4 inci dan pada layar 7 inci. dapat dilihat pada gambar 9 dan Gambar 10.



Gambar 9 Pengujian pada layar 4 inci



Gambar 10 Pengujian pada layar 7 inci

Berikut hasil pengujian pada smartphone yang berbeda, dapat dilihat pada table 3.

Tabel 3 Pengujian pada smartphone berbeda

No.	Handphone	Spesifikasi	Hasil Pengujian
1	Lenovo	<ul style="list-style-type: none"> - Sistem Operasi : Android 4.04 - Layar 4 inci - Prosesor : 1 Ghz - RAM : 512mb 	<ul style="list-style-type: none"> - Aplikasi kendali dan monitoring berjalan lancar - Tampilan aplikasi stabil pada posisi landscape dan fullscreen
2.	Smasung Tab Galaxy 2	<ul style="list-style-type: none"> - Sistem Operasi : Android 4.12 - Prosesor : 1Ghz - RAM : 1GB 	<ul style="list-style-type: none"> - Aplikasi berjalan lancar - Tampilan aplikasi menjadi tidak full screen



BAB V PENUTUP

5.1 Kesimpulan

Setelah dilakukan analisis dan pengujian pada aplikasi, maka didapatkan kesimpulan sebagai berikut:

1. Sistem ini telah berhasil melakukan koneksi antara Smartphone Android dengan robot beroda sebagai media pengirim perintah kendali terhadap robot melalui media bluetooth.
2. Sistem ini telah berhasil melakukan koneksi antara smartphone android dengan IP Camera sebagai pengirim streaming video untuk monitoring pergerakan robot melalui media wifi.
3. Sistem ini telah berhasil melakukan pengendalian robot beroda dengan memanfaatkan sensor accelerometer pada smartphone android serta melakukan monitoring robot yang di lengkapi IP Camera.
4. Pada saat menampilkan hasil monitoring, proses streaming video berjalan dengan lancar dan tidak terpengaruh oleh pergerakan sensor accelerometer saat melakukan kendali robot beroda.
5. Tampilan aplikasi kendali dan monitoring pada layar 4 inci stabil pada posisi landscape dan full screen, sedangkan pada layar 7 inci tampilan aplikasi kendali dan monitoring tidak full screen atau tidak *compatible*, karena pada saat pembuatan awal aplikasi kendali dan monitoring ini di *setting* pada layar 4 inci.
6. Robot yang digunakan sebagai perangkat pengujian pada aplikasi kendali dan monitoring berbasis android ini yaitu robot untuk riset.
7. Jarak jangkauan aplikasi kendali dan monitoring dengan robot ± 10 meter.
8. Pergerakan robot beroda terlalu responsife.
9. Spesifikasi minimum aplikasi kendali dan monitoring robot berbasis android versi 4.0 (ice cream sandwich)

5.2 Saran

Dalam pembuatan dan pengujian aplikasi ini terdapat beberapa kendala yang cukup berpengaruh, maka dari itu penulis menyampaikan saran sebagai berikut:

1. Untuk pengontrolan dan monitoring robot beroda disarankan minimal

menggunakan android versi 4.0

2. Untuk pengembangan selanjutnya, komunikasi antara robot beroda dengan Smartphone Android di anjurkan menggunakan media komunikasi yang lain selain bluetooth seperti wifi untuk menambah jarak jangkauan pengendalian robot.



DAFTAR PUSTAKA

- [1] Andriatma, Ryza. 2013. *Rancang Bangun Sistem Pengontrolan Robot Dengan Bluetooth dan Sensor Accelerometer Pada Smartphone Android*. Malang.
- [2] Anharku. (2009). Bluetooth. *Bluetooth*.
- [3] Anonim, 2010. Android Developers,
URL:<http://developer.android.com/index.html/>, diakses tanggal 12 Juni 2013
- [4] Chery, Situsblog,
URL:<http://cheryslearning.blogspot.com/2012/01/perkembangan-android.html/>,
19 April 2013
- [5] E. Simpson, J. (2002). *Just XML*. Yogyakarta: ANDI.
- [6] Hermawan, B. (2004). *Menguasai Java2 dan Object Oriented Programing*. Yogyakarta: ANDI.
- [7] Kadir, A. (2005). *Dasar Pemograman Java*. Yogyakarta. ANDI.
- [8] Pradana, Fitranda Arys, [jurnal], (<http://digilib.its.ac.id/IIS/download-jurnal/lego-mindstorm-nxt/>, diakses 18 April 2013)
- [9] Winoto, Ardi. (2008). *Mikrokontroler AVR ATmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*. Bandung. INFORMATIKA
-



LAMPIRAN



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : SU'AADA NURLAILA
NIM : 09.12.509
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer
MASA BIMBINGAN : SEMESTER GENAP 2012/2013
JUDUL : APLIKASI PENGENDALIAN DAN MONITORING ROBOT YANG
DI LENGKAPI IP CAMERA BERBASIS ANDROID

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Jumat
Tanggal : 16 Agustus 2013
Dengan Nilai : 81,25 (A) *12-*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP.Y.1030100358

Sekretaris Majelis Penguji

Dr. Eng. Aryanto S, ST, MT
NIP.P.1030800417

ANGGOTA PENGUJI

Dosen Penguji I

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Dosen Penguji II

Yuli Wahyuni ST, MT
NIP.P. 1031200456



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : SU'AADA NURLAILA
NIM : 09.12.509
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer
MASA BIMBINGAN : SEMESTER GENAP 2012/2013
JUDUL : APLIKASI PENGENDALIAN DAN MONITORING ROBOT
YANG DI LENGKAPI IP CAMERA BERBASIS ANDROID

Tanggal	Uraian	Paraf
Penguji I 16-08-2013		
Penguji II 16-08-2012	<ol style="list-style-type: none">1. Perbaikan pada kesimpulan2. Pengujian ditambahkan3. Abstrak diperbaiki4. Program diperbaiki5. Tujuan diperbaiki	

Disetujui,

Dosen Penguji I

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Dosen Penguji II

Yuli Wahyuni ST, MT
NIP.P. 1031200456

Mengetahui,

Dosen Pembimbing I

Dr. Eng. Aryuanto Soetedjo, ST, MT
NIP.P.1030800417

Dosen Pembimbing II






Bambang Prio Hartono, ST, MT
NIP.Y.1028400082



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341)-417636 Malang

BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Komputer

1.	Nim	: 0912509		
2.	Nama	: SUADA NURLAILA		
3.	Konsentrasi Jurusan	: Teknik Komputer		
4.	Jadwal Pelaksanaan:	Waktu	Tempat	
	25 April 2013	09:00	III.1.4	
5.	Judul proposal yang diseminarkan Mahasiswa	APLIKASI PENGENDALIAN DAN MONITORING ROBOT YANG DI LENGKAPI IP CAMERA BERBASIS ANDROID		
6.	Perubahan judul yang diusulkan oleh Kelompok Dosen Keahlian			
7.	Catatan :			
8.	Catatan :			
	Persetujuan judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
	 (..... <i>Yuw</i>)	 (.....)	 (.....)	
	Mengetahui, Ketua Program Studi Teknik Elektro S-1	Disetujui, Calon Dosen Pembimbing ybs		
 M. Ibrahim Ashari, ST, MT NIP. P 1030100358	Pembimbing I	Pembimbing II		
	 <i>Bambang Priatna</i>	(.....)		



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : SU'AADA NURLAILA
Nim : 09.12.509
Masa Bimbingan : Maret 2013 s/d agustus2013
Judul Skripsi : "APLIKASI PENGENDALIAN DAN MONITORING
ROBOT YANG DILENGKAPI IP CAMERA BERBASIS
ANDROID"

No	Tanggal	Uraian	Paraf Pembimbing
1	10/07 ¹³	Acc BAB I, 2, 3	
2	27-07-2013	Acc BAB IV	
3	27-07-2013	Acc BAB V	
4	27-07-2013	Acc Makalah	
5			
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing,

Dr. Eng. Aryuanto Soetedjo, ST, MT
NIP. 1030800417

Form.S-4b



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : SU'AADA NURLAILA
Nim : 09.12.509
Masa Bimbingan : Maret 2013 s/d agustus2013
Judul Skripsi : "APLIKASI PENGENDALIAN DAN MONITORING
ROBOT YANG DILENGKAPI IP CAMERA BERBASIS
ANDROID"

No	Tanggal	Uraian	Paraf Pembimbing
1	10-07-2013	Acc BAB I, II	
2	17-07-2013	Acc BAB III	
3	27-07-2013	Revisi BAB IV	
4	27-07-2013	Acc BAB IV, V, Makalah	
5			
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing,

Bambang Prio Hartono, ST, MT

NIP.Y. 1028400082

Form.S-4b

LISTING PROGRAM

MainActivity.java

```
package com.camera.simplemjpeg;

import java.io.IOException;

public class MainActivity extends Activity implements SensorEventListener {

    // accelerometer
    float gravityX, gravityY, gravityZ;
    float lin_accel_X, lin_accel_Y, lin_accel_Z;
    private SensorManager mSensorManager;
    private Sensor mSensor;
    float X, Y, Z, l, r;
    int l2, r2;
    float g = 9.81f;

    // Bluetooth
    // private static final int REQUEST_ENABLE_BT = 1;
    private static final String bt_HwAddr = "00:17:A0:01:69:EF";
    BluetoothAdapter btAdapter;
    TBlue bt;
    Handler timerHandler;
    Runnable sendToBT;
    int skips = 0;
    int sent = 0;
    String bt_command = ";";

    // MJPEG Viewer
    private static final boolean DEBUG = false;
    private MjpegView mv = null;
    String URL;

    // for settings (network and resolution)
    private static final int REQUEST_SETTINGS = 0;

    private int width = 320;
    private int height = 240;

    private String s_http = "http://";
    private int ip_ad1 = 192;
    private int ip_ad2 = 168;
    private int ip_ad3 = 43;
    private int ip_ad4 = 117;
    private int ip_port = 80;
    private String ip_command = "video.mjpg";

    private boolean suspending = false;

    // UI Components
    TextView txt_X, txt_Y, txt_Z, txt_bt, txt_accel;
    Button btn_maju, btn_kiri, btn_kanan, btn_mundur, btn_stop;

    // other
    private static final String APP_TAG = "APP";
    private static final String ACCEL_TAG = "Accel";
    private static final String BT_TAG = "BT";
    private static final String MJPEG_TAG = "MJPEG";
```

MainActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // vars
    X = 0; // lin_accel_X = 0;
    Y = 0; // lin_accel_Y = 0;
    Z = 0; // lin_accel_Z = 0;

    /*
    // UI components
    txt_X = (TextView) findViewById(R.id.txt_X);
    txt_Y = (TextView) findViewById(R.id.txt_Y);
    txt_Z = (TextView) findViewById(R.id.txt_Z);
    */
    txt_bt = (TextView) findViewById(R.id.txt_bt);
    txt_accel = (TextView) findViewById(R.id.txt_accel);

    btn_maju = (Button) findViewById(R.id.btn_maju);
    btn_kiri = (Button) findViewById(R.id.btn_kiri);
    btn_kanan = (Button) findViewById(R.id.btn_kanan);
    btn_mundur = (Button) findViewById(R.id.btn_mundur);
    btn_stop = (Button) findViewById(R.id.btn_stop);

    // Camera prefs
    SharedPreferences preferences = getSharedPreferences("SAVED_VALUES",
MODE_PRIVATE);
    width = preferences.getInt("width", width);
    height = preferences.getInt("height", height);
    ip_ad1 = preferences.getInt("ip_ad1", ip_ad1);
    ip_ad2 = preferences.getInt("ip_ad2", ip_ad2);
    ip_ad3 = preferences.getInt("ip_ad3", ip_ad3);
    ip_ad4 = preferences.getInt("ip_ad4", ip_ad4);
    ip_port = preferences.getInt("ip_port", ip_port);
    ip_command = preferences.getString("ip_command", ip_command);

    URL = s_http + ip_ad1 + "." + ip_ad2 + "." + ip_ad3 + "." + ip_ad4 + "/" +
ip_command;

    mv = (MjpegView) findViewById(R.id.mv);
    if(mv != null){
        mv.setResolution(width, height);
    }

    // Bluetooth handler
    timerHandler = new Handler();
    sendToBT = new Runnable() {
        public void run() {
            sendDataBT(bt_command);
            timerHandler.postDelayed(this, 250);
        }
    };

    btn_maju.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View arg0) {
```

MainActivity.java

```
// TODO Auto-generated method stub
bt_command = "aa;";
    }
});
btn_kiri.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        bt_command = "ac;";
    }
});
btn_kanan.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        bt_command = "ca;";
    }
});
btn_mundur.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        bt_command = "cc;";
    }
});
btn_stop.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        bt_command = "bb;";
    }
});

// Logs
Log.i(APP_TAG, "App initialized..");
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.layout.option_menu, menu);
    Log.i(APP_TAG, "Menu created..");
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.settings:
            Intent settings_intent = new Intent(MainActivity.this,
SettingsActivity.class);
            settings_intent.putExtra("width", width);
            settings_intent.putExtra("height", height);
            settings_intent.putExtra("ip_ad1", ip_ad1);
            settings_intent.putExtra("ip_ad2", ip_ad2);
            settings_intent.putExtra("ip_ad3", ip_ad3);
            settings_intent.putExtra("ip_ad4", ip_ad4);
            settings_intent.putExtra("ip_port", ip_port);

```

MainActivity.java

```

        settings_intent.putExtra("ip_command", ip_command);
        startActivityForResult(settings_intent, REQUEST_SETTINGS);
        return true;
    }
    return false;
}

public void onActivityResult(int requestCode, int resultCode, Intent data) {
    switch (requestCode) {
        case REQUEST_SETTINGS:
            if (resultCode == Activity.RESULT_OK) {
                width = data.getIntExtra("width", width);
                height = data.getIntExtra("height", height);
                ip_ad1 = data.getIntExtra("ip_ad1", ip_ad1);
                ip_ad2 = data.getIntExtra("ip_ad2", ip_ad2);
                ip_ad3 = data.getIntExtra("ip_ad3", ip_ad3);
                ip_ad4 = data.getIntExtra("ip_ad4", ip_ad4);
                ip_port = data.getIntExtra("ip_port", ip_port);
                ip_command = data.getStringExtra("ip_command");

                if(mv!=null){
                    mv.setResolution(width, height);
                }
                SharedPreferences preferences = getSharedPreferences("SAVED_VALUES",
MODE_PRIVATE);
                SharedPreferences.Editor editor = preferences.edit();
                editor.putInt("width", width);
                editor.putInt("height", height);
                editor.putInt("ip_ad1", ip_ad1);
                editor.putInt("ip_ad2", ip_ad2);
                editor.putInt("ip_ad3", ip_ad3);
                editor.putInt("ip_ad4", ip_ad4);
                editor.putInt("ip_port", ip_port);
                editor.putString("ip_command", ip_command);

                editor.commit();

                new RestartApp().execute();
            }
            break;
    }
}

public class DoRead extends AsyncTask<String, Void, MjpegInputStream> {
    protected MjpegInputStream doInBackground(String... url) {
        //TODO: if camera has authentication deal with it and don't just not work

        HttpResponse res = null;
        DefaultHttpClient httpClient = new DefaultHttpClient();
        HttpParams httpParams = httpClient.getParams();
        HttpConnectionParams.setConnectionTimeout(httpParams, 5 * 1000);
        Log.i(MJPEG_TAG, "1. Sending http request");
        try {
            res = httpClient.execute(new HttpGet(Uri.create(url[0])));
            Log.i(MJPEG_TAG, "2. Request finished, status = " +
res.getStatusLine().getStatusCode());
            if(res.getStatusLine().getStatusCode()==401){
                //You must turn off camera User Access Control before this will work
            }
        }
    }
}

```

MainActivity.java

```

        return null;
    }
    return new MjpegInputStream(res.getEntity().getContent());
} catch (ClientProtocolException e) {
    e.printStackTrace();
    Log.i(MJPEG_TAG, "Request failed-ClientProtocolException", e);
    //Error connecting to camera
} catch (IOException e) {
    e.printStackTrace();
    Log.i(MJPEG_TAG, "Request failed-IOException", e);
    //Error connecting to camera
}
return null;
}

protected void onPostExecute(MjpegInputStream result) {
    mv.setSource(result);
    if(result!=null) result.setSkip(1);
    mv.setDisplayMode(MjpegView.SIZE_BEST_FIT);
    mv.showFps(false);
}

}

public class RestartApp extends AsyncTask<Void, Void, Void> {
    protected Void doInBackground(Void... v) {
        MainActivity.this.finish();
        return null;
    }

    protected void onPostExecute(Void v) {
        startActivity((new Intent(MainActivity.this, MainActivity.class)));
    }
}

@Override
public void onSensorChanged(SensorEvent event) {
    X = event.values[0]/g;
    Y = event.values[1]/g;
    Z = event.values[2]/g;

    if (X < 0.0 && Y > 0.0) { // maju
        bt_command = "aa";
    }
    else if (X > 0.4) { // mundur/stop
        bt_command = "bb";
    }
    else if (X > 0.0 && Y < 0.0) { // kiri
        bt_command = "ac";
    }
    else if (X > 0.0 && Y > 0.0) { // kanan
        bt_command = "ca";
    }
}

/*
txt_X.setText("X = " + event.values[0]);
txt_Y.setText("Y = " + event.values[1]);
txt_Z.setText("Z = " + event.values[2]);
*/

```



```
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // TODO Auto-generated method stub
}

public void onStart() {
    if(DEBUG) Log.i(MJPEG_TAG, "onStart()");
    super.onStart();
}

@Override
public void onPause() {
    super.onPause();

    stopAccel();
    stopBT();

    timerHandler.removeCallbacks(sendToBT);
    txt_bt.setText("Bluetooth: Paused.");

    if(mv!=null){
        if(mv.isStreaming()){
            mv.stopPlayback();
            suspending = true;
        }
    }
}

@Override
public void onResume() {
    super.onResume();

    startAccel();
    timerHandler.postDelayed(sendToBT, 250);
    skips = 9999;

    if(DEBUG) Log.i(MJPEG_TAG, "onResume()");

    if(mv!=null){
        if(suspending){
            new DoRead().execute(URL);
            suspending = false;
        }
    }
}

public void onStop() {
    if(DEBUG) Log.i(MJPEG_TAG, "onStop()");
    super.onStop();
}

public void onDestroy() {
    if(DEBUG) Log.i(MJPEG_TAG, "onDestroy()");

    if(mv!=null){
        mv.freeCameraMemory();
    }
}
```

MainActivity.java

```

    }
    super.onDestroy();
}

void startAccel() {
    mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);

    if (mSensorManager.getSensorList(Sensor.TYPE_ACCELEROMETER).size() == 0) {
        txt_accel.setText("Accelerometer : NOT FOUND");
        Log.e(ACCEL_TAG, "Accelerometer not found.");
    }
    else {
        mSensor = mSensorManager.getSensorList(Sensor.TYPE_ACCELEROMETER).get(0);
        if (mSensorManager.registerListener(this, mSensor,
SensorManager.SENSOR_DELAY_NORMAL)) {
            txt_accel.setText("Accelerometer: ON");
            Log.i(ACCEL_TAG, "Accelerometer started.");
        }
    }
}

void stopAccel() {
    txt_accel.setText("Accelerometer: OFF.");
    Log.i(ACCEL_TAG, "Accelerometer stopped.");
    mSensorManager.unregisterListener(this, mSensor);
}

void startBT() {
    bt = new TBlue(bt_HwAddr);
    skips = 0;
    if (bt.streaming()) {
        txt_bt.setText("Bluetooth: ON");
        Log.i(BT_TAG, "Bluetooth started.");

        new DoRead().execute(URL);
    }
}

void stopBT() {
    txt_bt.setText("Bluetooth: OFF");
    Log.i(BT_TAG, "Bluetooth stopped");
    bt.close();
}

void sendDataBT(String cmd) {
    if (skips > 20) {
        stopAccel();

        startBT();
        startAccel();
        return;
    }

    if (!bt.streaming()) {
        skips++;
        return;
    }
    if (bt.streaming()) {

```

MainActivity.java

```
        bt.write(cmd);
        sent++;
        txt_bt.setText("Bluetooth: SEND (" + cmd + ")");
        skips = 0;
    }
    else {
        skips++;
    }
}
}
```

MjpegInputStream.java

```

package com.camera.simplemjpeg;

import java.io.BufferedInputStream;

public class MjpegInputStream extends DataInputStream {
    private final byte[] SOI_MARKER = { (byte) 0xFF, (byte) 0xD8 };
    private final byte[] EOF_MARKER = { (byte) 0xFF, (byte) 0xD9 };
    private final String CONTENT_LENGTH = "Content-Length";
    private final static int HEADER_MAX_LENGTH = 100;
    //private final static int FRAME_MAX_LENGTH = 40000 + HEADER_MAX_LENGTH;
    private final static int FRAME_MAX_LENGTH = 200000;
    private int mContentLength = -1;
    byte[] header = null;
    byte[] frameData = null;
    int headerLen = -1;
    int headerLenPrev = -1;

    int skip=1;
    int count=0;

    private static final String TAG="MJPEG";
    private static final boolean DEBUG=false;

    static {
        System.loadLibrary("ImageProc");
    }

    public native int pixeltobmp(byte[] jp, int l, Bitmap bmp);
    public native void freeCameraMemory();

    public static MjpegInputStream read(String surl) {
        try {
            URL url = new URL(surl);
            HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
            return new MjpegInputStream(urlConnection.getInputStream());
        } catch (Exception e) {}

        return null;
    }

    public MjpegInputStream(InputStream in) {
        super(new BufferedInputStream(in, FRAME_MAX_LENGTH));
    }

    private int getEndOfSequence(DataInputStream in, byte[] sequence)
        throws IOException
    {
        int seqIndex = 0;
        byte c;
        for(int i=0; i < FRAME_MAX_LENGTH; i++) {
            c = (byte) in.readUnsignedByte();
            if(c == sequence[seqIndex]) {
                seqIndex++;
                if(seqIndex == sequence.length){
                    return i + 1;
                }
            }
        }
    }

```

MjpegInputStream.java

```

        } else seqIndex = 0;
    }

    return -1;
}

private int getStartOfSequence(DataInputStream in, byte[] sequence)
    throws IOException
{
    int end = getEndOfSequence(in, sequence);
    return (end < 0) ? (-1) : (end - sequence.length);
}

private int getEndOfSequenceSimplified(DataInputStream in, byte[] sequence)
    throws IOException
{
    int startPos = mContentLength/2;
    int endPos = 3*mContentLength/2;

    skipBytes(headerLen+startPos);

    int seqIndex = 0;
    byte c;
    for(int i=0; i < endPos-startPos ; i++) {
        c = (byte) in.readUnsignedByte();
        if(c == sequence[seqIndex]) {
            seqIndex++;
            if(seqIndex == sequence.length){
                return headerLen + startPos + i + 1;
            }
        } else seqIndex = 0;
    }

    return -1;
}

private int parseContentLength(byte[] headerBytes)
    throws IOException, NumberFormatException, IllegalArgumentException
{
    ByteArrayInputStream headerIn = new ByteArrayInputStream(headerBytes);
    Properties props = new Properties();
    props.load(headerIn);
    return Integer.parseInt(props.getProperty(CONTENT_LENGTH));
}

public Bitmap readMjpegFrame() throws IOException {
    mark(FRAME_MAX_LENGTH);
    int headerLen;
    try{
        headerLen = getStartOfSequence(this, SOI_MARKER);
    }catch(IOException e){
        if(DEBUG) Log.d(TAG,"IOException in betting headerLen.");
        reset();
        return null;
    }
}

```

```

}
reset();

if(header==null || headerLen != headerLenPrev){
    header = new byte[headerLen];
    if(DEBUG) Log.d(TAG,"header renewed "+headerLenPrev+" -> "+headerLen);
}
headerLenPrev = headerLen;
readFully(header);

int ContentLengthNew = -1;
try {
    ContentLengthNew = parseContentLength(header);
} catch (NumberFormatException nfe) {
    ContentLengthNew = getEndOfSequenceSimplified(this, EOF_MARKER);

    if(ContentLengthNew < 0){
        if(DEBUG) Log.d(TAG,"worst case for finding EOF_MARKER");
        reset();
        ContentLengthNew = getEndOfSequence(this, EOF_MARKER);
    }
} catch (IllegalArgumentException e) {
    if(DEBUG) Log.d(TAG,"IllegalArgumentException in parseContentLength");
    ContentLengthNew = getEndOfSequenceSimplified(this, EOF_MARKER);

    if(ContentLengthNew < 0){
        if(DEBUG) Log.d(TAG,"worst case for finding EOF_MARKER");
        reset();
        ContentLengthNew = getEndOfSequence(this, EOF_MARKER);
    }
} catch (IOException e) {
    if(DEBUG) Log.d(TAG,"IOException in parseContentLength");
    reset();
    return null;
}
mContentLength = ContentLengthNew;
reset();

if(frameData==null){
    frameData = new byte[FRAME_MAX_LENGTH];
    if(DEBUG) Log.d(TAG,"frameData newed cl="+FRAME_MAX_LENGTH);
}
if(mContentLength + HEADER_MAX_LENGTH > FRAME_MAX_LENGTH){
    frameData = new byte[mContentLength + HEADER_MAX_LENGTH];
    if(DEBUG) Log.d(TAG,"frameData renewed cl="+mContentLength +
HEADER_MAX_LENGTH);
}

skipBytes(headerLen);

readFully(frameData, 0, mContentLength);

if(count++%skip==0){
    return BitmapFactory.decodeStream(new
ByteArrayInputStream(frameData,0,mContentLength));
}else{
    return null;
}
}

```

```

}

public int readMjpegFrame(Bitmap bmp) throws IOException {
    mark(FRAME_MAX_LENGTH);
    int headerLen;
    try{
        headerLen = getStartOfSequence(this, SOI_MARKER);
    }catch(IOException e){
        if(DEBUG) Log.d(TAG,"IOException in betting headerLen.");
        reset();
        return -1;
    }
    reset();

    if(header==null || headerLen != headerLenPrev){
        header = new byte[headerLen];
        if(DEBUG) Log.d(TAG,"header renewed "+headerLenPrev+" -> "+headerLen);
    }
    headerLenPrev = headerLen;
    readFully(header);

    int ContentLengthNew = -1;
    try {
        ContentLengthNew = parseContentLength(header);
    } catch (NumberFormatException nfe) {
        ContentLengthNew = getEndOfSequenceSimplified(this, EOF_MARKER);

        if(ContentLengthNew < 0){
            if(DEBUG) Log.d(TAG,"Worst case for finding EOF_MARKER");
            reset();
            ContentLengthNew = getEndOfSequence(this, EOF_MARKER);
        }
    }catch (IllegalArgumentException e) {
        if(DEBUG) Log.d(TAG,"IllegalArgumentException in parseContentLength");
        ContentLengthNew = getEndOfSequenceSimplified(this, EOF_MARKER);

        if(ContentLengthNew < 0){
            if(DEBUG) Log.d(TAG,"Worst case for finding EOF_MARKER");
            reset();
            ContentLengthNew = getEndOfSequence(this, EOF_MARKER);
        }
    }catch (IOException e) {
        if(DEBUG) Log.d(TAG,"IOException in parseContentLength");
        reset();
        return -1;
    }
    mContentLength = ContentLengthNew;
    reset();

    if(frameData==null){
        frameData = new byte[FRAME_MAX_LENGTH];
        if(DEBUG) Log.d(TAG,"frameData newed cl="+FRAME_MAX_LENGTH);
    }
    if(mContentLength + HEADER_MAX_LENGTH > FRAME_MAX_LENGTH){
        frameData = new byte[mContentLength + HEADER_MAX_LENGTH];
        if(DEBUG) Log.d(TAG,"frameData renewed cl="+mContentLength +
HEADER_MAX_LENGTH));
    }
}

```

MjpegInputStream.java

```
        skipBytes(headerLen);
        readFully(frameData, 0, mContentLength);
        if(count++%skip==0){
            return pixeltobmp(frameData, mContentLength, bmp);
        }else{
            return 0;
        }
    }
    public void setSkip(int s){
        skip = s;
    }
}
```


MjpegView.java

```
package com.camera.simplemjpeg;

import java.io.IOException;

public class MjpegView extends SurfaceView implements SurfaceHolder.Callback {

    public static final String TAG = "MJPEG";

    public final static int POSITION_UPPER_LEFT = 1;
    public final static int POSITION_UPPER_RIGHT = 1;
    public final static int POSITION_LOWER_LEFT = 1;
    public final static int POSITION_LOWER_RIGHT = 1;

    public final static int SIZE_STANDARD = 1;
    public final static int SIZE_BEST_FIT = 4;
    public final static int SIZE_FULLSCREEN = 8;

    SurfaceHolder holder;
    Context saved_context;

    private MjpegViewThread thread;
    private MjpegInputStream mIn = null;
    private boolean showFps = false;
    private boolean mRun = false;
    private boolean surfaceDone = false;

    private Paint overlayPaint;
    private int overlayTextColor;
    private int overlayBackgroundColor;
    private int ovlPos;
    private int dispWidth;
    private int dispHeight;
    private int displayMode;

    private boolean suspending = false;

    private Bitmap bmp = null;

    // image size

    public int IMG_WIDTH=320;
    public int IMG_HEIGHT=240;

    public class MjpegViewThread extends Thread {
        private SurfaceHolder mSurfaceHolder;
        private int frameCounter = 0;
        private long start;
        private String fps = "";

        public MjpegViewThread(SurfaceHolder surfaceHolder, Context context) {
            mSurfaceHolder = surfaceHolder;
        }

        private Rect destRect(int bmw, int bmh) {
            int tempx;
            int tempy;
            if (displayMode == MjpegView.SIZE_STANDARD) {
```

MjpegView.java

```
        tempX = (dispWidth / 2) - (bmw / 2);
        tempY = (dispHeight / 2) - (bmh / 2);
        return new Rect(tempX, tempY, bmw + tempX, bmh + tempY);
    }
    if (displayMode == MjpegView.SIZE_BEST_FIT) {
        float bmasp = (float) bmw / (float) bmh;
        bmw = dispWidth;
        bmh = (int) (dispWidth / bmasp);
        if (bmh > dispHeight) {
            bmh = dispHeight;
            bmw = (int) (dispHeight * bmasp);
        }
        tempX = (dispWidth / 2) - (bmw / 2);
        tempY = (dispHeight / 2) - (bmh / 2);
        return new Rect(tempX, tempY, bmw + tempX, bmh + tempY);
    }
    if (displayMode == MjpegView.SIZE_FULLSCREEN)
        return new Rect(0, 0, dispWidth, dispHeight);
    return null;
}

public void setSurfaceSize(int width, int height) {
    synchronized(mSurfaceHolder) {
        dispWidth = width;
        dispHeight = height;
    }
}

private Bitmap makeFpsOverlay(Paint p) {
    Rect b = new Rect();
    p.getTextBounds(fps, 0, fps.length(), b);

    // false indentation to fix forum layout
    Bitmap bm = Bitmap.createBitmap(b.width(), b.height(),
    Bitmap.Config.ARGB_8888);

    Canvas c = new Canvas(bm);
    p.setColor(overlayBackgroundColor);
    c.drawRect(0, 0, b.width(), b.height(), p);
    p.setColor(overlayTextColor);
    c.drawText(fps, -b.left, b.bottom-b.top-p.descent(), p);
    return bm;
}

public void run() {
    start = System.currentTimeMillis();
    PorterDuffXfermode mode = new PorterDuffXfermode(PorterDuff.Mode.DST_OVER);

    int width;
    int height;
    Paint p = new Paint();
    Bitmap ovl = null;

    while (mRun) {

        Rect destRect=null;
        Canvas c = null;
```

MjpegView.java

```
        if(surfaceDone) {
            try {
                if bmp == null {
                    bmp = Bitmap.createBitmap(IMG_WIDTH, IMG_HEIGHT,
Bitmap.Config.ARGB_8888);
                }
                int ret = mIn.readMjpegFrame(bmp);

                if (ret == -1)
                {
                    return;
                }

                destRect = destRect(bmp.getWidth(), bmp.getHeight());

                c = mSurfaceHolder.lockCanvas();
                synchronized (mSurfaceHolder) {

                    c.drawBitmap(bmp, null, destRect, p);

                    if (showFps) {
                        p.setXfermode(mode);
                        if (ovl != null) {

                            // false indentation to fix forum layout

                            height = ((ovlPos & 1) == 1) ? destRect.top :
destRect.bottom - ovl.getHeight();
                            width = ((ovlPos & 8) == 8) ? destRect.left :
destRect.right - ovl.getWidth();

                            c.drawBitmap(ovl, width, height, null);
                        }
                        p.setXfermode(null);
                        frameCounter++;
                        if ((System.currentTimeMillis() - start) >= 1000) {
                            fps = String.valueOf(frameCounter) + "fps";
                            frameCounter = 0;
                            start = System.currentTimeMillis();
                            if (ovl != null) ovl.recycle();

                            ovl = makeFpsOverlay(overlayPaint);
                        }
                    }
                }
            }
        }
    } catch (IOException e){
    } finally {
        if (c != null) mSurfaceHolder.unlockCanvasAndPost(c);
    }
}
}
}
```

```

private void init(Context context) {

    //SurfaceHolder holder = getHolder();
    holder = getHolder();
    saved_context = context;
    holder.addCallback(this);
    thread = new MjpegViewThread(holder, context);
    setFocusable(true);
    overlayPaint = new Paint();
    overlayPaint.setTextAlign(Paint.Align.LEFT);
    overlayPaint.setTextSize(12);
    overlayPaint.setTypeface(Typeface.DEFAULT);
    overlayTextColor = Color.WHITE;
    overlayBackgroundColor = Color.BLACK;
    ovlPos = MjpegView.POSITION_LOWER_RIGHT;
    displayMode = MjpegView.SIZE_STANDARD;
    dispWidth = getWidth();
    dispHeight = getHeight();
}

public void startPlayback() {
    if(mIn != null) {
        mRun = true;
        if(thread==null){
            thread = new MjpegViewThread(holder, saved_context);
        }
        thread.start();
    }
}

public void resumePlayback() {
    if(suspending){
        if(mIn != null) {
            mRun = true;
            SurfaceHolder holder = getHolder();
            holder.addCallback(this);
            thread = new MjpegViewThread(holder, saved_context);
            thread.start();
            suspending=false;
        }
    }
}

public void stopPlayback() {
    if(mRun){
        suspending = true;
    }
    mRun = false;
    if(thread!=null){
        boolean retry = true;
        while(retry) {
            try {
                thread.join();
                retry = false;
            } catch (InterruptedException e) {}
        }
        thread = null;
    }
    if(mIn!=null){

```

MjpegView.java

```

        try{
            mIn.close();
        }catch(IOException e){}
        mIn = null;
    }

}

public void freeCameraMemory(){
    if(mIn!=null){
        mIn.freeCameraMemory();
    }
}

public MjpegView(Context context, AttributeSet attrs) {
    super(context, attrs); init(context);
}

public void surfaceChanged(SurfaceHolder holder, int f, int w, int h) {
    if(thread!=null){
        thread.setSurfaceSize(w, h);
    }
}

public void surfaceDestroyed(SurfaceHolder holder) {
    surfaceDone = false;
    stopPlayback();
}

public MjpegView(Context context) { super(context); init(context); }
public void surfaceCreated(SurfaceHolder holder) { surfaceDone = true; }
public void showFps(boolean b) { showFps = b; }
public void setSource(MjpegInputStream source) {
    mIn = source;
    if(!suspending){
        startPlayback();
    }else{
        resumePlayback();
    }
}
}
public void setOverlayPaint(Paint p) { overlayPaint = p; }
public void setOverlayTextColor(int c) { overlayTextColor = c; }
public void setOverlayBackgroundColor(int c) { overlayBackgroundColor = c; }
public void setOverlayPosition(int p) { ovlPos = p; }
public void setDisplayMode(int s) { displayMode = s; }

public void setResolution(int w, int h){
    IMG_WIDTH = w;
    IMG_HEIGHT = h;
}

public boolean isStreaming(){
    return mRun;
}
}

```

SettingsActivity.java

```
package com.camera.simplemjpeg;

import com.camera.ip_cam.R;

public class SettingsActivity extends Activity {

    Button settings_done;

    Spinner resolution_spinner;
    EditText width_input;
    EditText height_input;

    EditText address1_input;
    EditText address2_input;
    EditText address3_input;
    EditText address4_input;
    EditText port_input;
    EditText command_input;

    Button address1_increment;
    Button address2_increment;
    Button address3_increment;
    Button address4_increment;

    Button address1_decrement;
    Button address2_decrement;
    Button address3_decrement;
    Button address4_decrement;

    RadioGroup port_group;
    RadioGroup command_group;

    int width = 640;
    int height = 480;

    int ip_ad1 = 192;
    int ip_ad2 = 168;
    int ip_ad3 = 2;
    int ip_ad4 = 1;
    int ip_port = 80;
    String ip_command = "?action=stream";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.settings);

        Bundle extras = getIntent().getExtras();

        ArrayAdapter<CharSequence> adapter =
            ArrayAdapter.createFromResource(this, R.array.resolution_array,
                android.R.layout.simple_spinner_item);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

        resolution_spinner = (Spinner) findViewById(R.id.resolution_spinner);
        resolution_spinner.setAdapter(adapter);

        width_input = (EditText) findViewById(R.id.width_input);
```

SettingsActivity.java

```
height_input = (EditText) findViewById(R.id.height_input);

address1_input = (EditText) findViewById(R.id.address1_input);
address2_input = (EditText) findViewById(R.id.address2_input);
address3_input = (EditText) findViewById(R.id.address3_input);
address4_input = (EditText) findViewById(R.id.address4_input);
port_input = (EditText) findViewById(R.id.port_input);
command_input = (EditText) findViewById(R.id.command_input);

port_group = (RadioGroup) findViewById(R.id.port_radiogroup);
command_group = (RadioGroup) findViewById(R.id.command_radiogroup);

if(extras != null){
    width = extras.getInt("width", width);
    height = extras.getInt("height", height);

    ip_ad1 = extras.getInt("ip_ad1", ip_ad1);
    ip_ad2 = extras.getInt("ip_ad2", ip_ad2);
    ip_ad3 = extras.getInt("ip_ad3", ip_ad3);
    ip_ad4 = extras.getInt("ip_ad4", ip_ad4);
    ip_port = extras.getInt("ip_port", ip_port);
    ip_command = extras.getString("ip_command");

    width_input.setText(String.valueOf(width));
    height_input.setText(String.valueOf(height));
    resolution_spinner.setSelection(adapter.getCount()-1);

    address1_input.setText(String.valueOf(ip_ad1));
    address2_input.setText(String.valueOf(ip_ad2));
    address3_input.setText(String.valueOf(ip_ad3));
    address4_input.setText(String.valueOf(ip_ad4));
    port_input.setText(String.valueOf(ip_port));
    command_input.setText(ip_command);
}

resolution_spinner.setOnItemClickListener(new
AdapterView.OnItemClickListener(){
    public void onItemClick(AdapterView<?> parent, View v1w, int arg2, long
arg3) {

        Spinner spinner = (Spinner)parent;
        String item = (String)spinner.getSelectedItem();
        if(item.equals("640x480")){
            width = 640;
            height = 480;
        }else if(item.equals("480x640")){
            width = 480;
            height = 640;
        }else if(item.equals("320x240")){
            width = 320;
            height = 240;
        }else if(item.equals("240x320")){
            width = 240;
            height = 320;
        }else if(item.equals("176x144")){
            width = 176;
            height = 144;
        }else if(item.equals("144x176")){
            width = 144;
```

SettingsActivity.java

```

        height = 176;
    }
    width_input.setText(String.valueOf(width));
    height_input.setText(String.valueOf(height));
}
public void onNothingSelected(AdapterView<?> parent) {
}
});

address1_increment = (Button)findViewById(R.id.address1_increment);
address1_increment.setOnClickListener(
    new View.OnClickListener(){
        public void onClick(View view){
            String s = address1_input.getText().toString();
            int val =ip_ad1;
            if(!"".equals(s)){
                val = Integer.parseInt(s);
            }
            if(val>=0 && val<255){
                val += 1;
            }else if(val < 0){
                val = 0;
            }else if(val >= 255){
                val = 255;
            }

            ip_ad1 = val;
            address1_input.setText(String.valueOf(ip_ad1),
BufferType.NORMAL);
        }
    }
);
address2_increment = (Button)findViewById(R.id.address2_increment);
address2_increment.setOnClickListener(
    new View.OnClickListener(){
        public void onClick(View view){
            String s = address2_input.getText().toString();
            int val =ip_ad2;
            if(!"".equals(s)){
                val = Integer.parseInt(s);
            }
            if(val>=0 && val<255){
                val += 1;
            }else if(val < 0){
                val = 0;
            }else if(val >= 255){
                val = 255;
            }

            ip_ad2 = val;
            address2_input.setText(String.valueOf(ip_ad2),
BufferType.NORMAL);
        }
    }
);
address3_increment = (Button)findViewById(R.id.address3_increment);

```


SettingsActivity.java

```

address3_increment.setOnClickListener(
    new View.OnClickListener(){
        public void onClick(View view){
            String s = address3_input.getText().toString();
            int val =ip_ad3;
            if(!"".equals(s)){
                val = Integer.parseInt(s);
            }
            if(val>=0 && val<255){
                val += 1;
            }else if(val < 0){
                val = 0;
            }else if(val >= 255){
                val = 255;
            }

            ip_ad3 = val;
            address3_input.setText(String.valueOf(ip_ad3),
BufferType.NORMAL);
        }
    });
address4_increment = (Button)findViewById(R.id.address4_increment);
address4_increment.setOnClickListener(
    new View.OnClickListener(){
        public void onClick(View view){
            String s = address4_input.getText().toString();
            int val =ip_ad4;
            if(!"".equals(s)){
                val = Integer.parseInt(s);
            }
            if(val>=0 && val<255){
                val += 1;
            }else if(val < 0){
                val = 0;
            }else if(val >= 255){
                val = 255;
            }

            ip_ad4 = val;
            address4_input.setText(String.valueOf(ip_ad4),
BufferType.NORMAL);
        }
    });
address1_decrement = (Button)findViewById(R.id.address1_decrement);
address1_decrement.setOnClickListener(
    new View.OnClickListener(){
        public void onClick(View view){
            String s = address1_input.getText().toString();
            int val =ip_ad1;
            if(!"".equals(s)){
                val = Integer.parseInt(s);
            }
            if(val>0 && val<=255){

```

SettingsActivity.java

```

        val -= 1;
    }else if(val <= 0){
        val = 0;
    }else if(val > 255){
        val = 255;
    }

    ip_ad1 = val;
    address1_input.setText(String.valueOf(ip_ad1),
BufferType.NORMAL);
    }
}

);

address2_decrement = (Button)findViewById(R.id.address2_decrement);
address2_decrement.setOnClickListener(
    new View.OnClickListener(){
        public void onClick(View view){
            String s = address2_input.getText().toString();
            int val =ip_ad2;
            if(!"".equals(s)){
                val = Integer.parseInt(s);
            }
            if(val>0 && val<=255){
                val -= 1;
            }else if(val <= 0){
                val = 0;
            }else if(val > 255){
                val = 255;
            }

            ip_ad2 = val;
            address2_input.setText(String.valueOf(ip_ad2),
BufferType.NORMAL);
        }
    }
);

address3_decrement = (Button)findViewById(R.id.address3_decrement);
address3_decrement.setOnClickListener(
    new View.OnClickListener(){
        public void onClick(View view){
            String s = address3_input.getText().toString();
            int val =ip_ad3;
            if(!"".equals(s)){
                val = Integer.parseInt(s);
            }
            if(val>0 && val<=255){
                val -= 1;
            }else if(val <= 0){
                val = 0;
            }else if(val > 255){
                val = 255;
            }

            ip_ad3 = val;
            address3_input.setText(String.valueOf(ip_ad3),

```

SettingsActivity.java

```

        BufferType.NORMAL);
    }
}
);
address4_decrement = (Button)findViewById(R.id.address4_decrement);
address4_decrement.setOnClickListener(
    new View.OnClickListener(){
        public void onClick(View view){
            String s = address4_input.getText().toString();
            int val =ip_ad4;
            if(!"".equals(s)){
                val = Integer.parseInt(s);
            }
            if(val>0 && val<=255){
                val -= 1;
            }else if(val <= 0){
                val = 0;
            }else if(val > 255){
                val = 255;
            }

            ip_ad4 = val;
            address4_input.setText(String.valueOf(ip_ad4),
BufferType.NORMAL);
        }
    }
);

port_group.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        if(checkedId == R.id.port_80){
            port_input.setText(getString(R.string.port_80));
        }else if(checkedId == R.id.port_8080){
            port_input.setText(getString(R.string.port_8080));
        }
    }
});

command_group.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener()
{
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        if(checkedId == R.id.command_streaming){
            command_input.setText(getString(R.string.command_streaming));
        }else if(checkedId == R.id.command_videofeed){
            command_input.setText(getString(R.string.command_videofeed));
        }
    }
});

settings_done = (Button)findViewById(R.id.settings_done);
settings_done.setOnClickListener(
    new View.OnClickListener(){
        public void onClick(View view){

            String s;

```

SettingsActivity.java

```
s = width_input.getText().toString();
if(!"".equals(s)){
    width = Integer.parseInt(s);
}
s = height_input.getText().toString();
if(!"".equals(s)){
    height = Integer.parseInt(s);
}
s = address1_input.getText().toString();
if(!"".equals(s)){
    ip_ad1 = Integer.parseInt(s);
}
s = address2_input.getText().toString();
if(!"".equals(s)){
    ip_ad2 = Integer.parseInt(s);
}
s = address3_input.getText().toString();
if(!"".equals(s)){
    ip_ad3 = Integer.parseInt(s);
}
s = address4_input.getText().toString();
if(!"".equals(s)){
    ip_ad4 = Integer.parseInt(s);
}

s = port_input.getText().toString();
if(!"".equals(s)){
    ip_port = Integer.parseInt(s);
}

s = command_input.getText().toString();
ip_command = s;

Intent intent = new Intent();
intent.putExtra("width", width);
intent.putExtra("height", height);
intent.putExtra("ip_ad1", ip_ad1);
intent.putExtra("ip_ad2", ip_ad2);
intent.putExtra("ip_ad3", ip_ad3);
intent.putExtra("ip_ad4", ip_ad4);
intent.putExtra("ip_port", ip_port);
intent.putExtra("ip_command", ip_command);

setResult(RESULT_OK, intent);
finish();
}
}
);
}
}
}
```

```

package com.camera.simplemjpeg;

import java.io.IOException;

public class TBlue {
    String address = null;
    String TAG = "tBlue";
    BluetoothAdapter localAdapter = null;
    BluetoothDevice remoteDevice = null;
    BluetoothSocket socket = null;

    public OutputStream outStream = null;
    public InputStream inStream = null;
    boolean failed = false;
    public TBlue(String address)
    {
        this.address=address.toUpperCase();
        localAdapter = BluetoothAdapter.getDefaultAdapter();

        if ((localAdapter!=null) && localAdapter.isEnabled()) {
            Log.i(TAG, "Bluetooth adapter found and enabled on phone. ");
        } else {
            Log.e(TAG, "Bluetooth adapter NOT FOUND or NOT ENABLED!");
            return;
        }

        connect();
    }

    public void connect() {
        Log.i(TAG, "Bluetooth connecting to "+address+"...");

        try {
            remoteDevice = localAdapter.getRemoteDevice(address);
        } catch (IllegalArgumentException e) {
            Log.e(TAG, "Failed to get remote device with MAC address. Wrong format? MAC
address must be upper case. ", e);
            return;
        }

        Log.i(TAG, "Creating RFCOMM socket...");

        try {
            Method m = remoteDevice.getClass().getMethod("createRfcommSocket", new
Class[] { int.class });
            socket = (BluetoothSocket) m.invoke(remoteDevice, 1);
            Log.i(TAG, "RFCOMM socket created.");
        } catch (NoSuchMethodException e) {
            Log.i(TAG, "Could not invoke createRfcommSocket.");
            e.printStackTrace();
        } catch (IllegalArgumentException e) {
            Log.i(TAG, "Bad argument with createRfcommSocket.");
            e.printStackTrace();
        } catch (IllegalAccessException e) {
            Log.i(TAG, "Illegal access with createRfcommSocket.");
            e.printStackTrace();
        } catch (InvocationTargetException e) {
            Log.i(TAG, "Invocation target exception: createRfcommSocket.");
        }
    }
}

```

```

        e.printStackTrace();
    }

    Log.i(TAG, "Got socket for device "+socket.getRemoteDevice());
    localAdapter.cancelDiscovery();
    Log.i(TAG, "Connecting socket...");

    try {
        socket.connect();
        Log.i(TAG, "Socket connected.");
    } catch (IOException e) {

        try {
            Log.e(TAG, "Failed to connect socket. ", e);
            socket.close();
            Log.e(TAG, "Socket closed because of an error. ", e);
        } catch (IOException eb) {
            Log.e(TAG, "Also failed to close socket. ", eb);
        }
        return;
    }

    try {
        outputStream = socket.getOutputStream();
        Log.i(TAG, "Output stream open.");
        inputStream = socket.getInputStream();
        Log.i(TAG, "Input stream open.");
    } catch (IOException e) {
        Log.e(TAG, "Failed to create output stream.", e);
    }
    return;
}

public void write(String s) {
    Log.i(TAG, "Sending \""+s+"\"... ");
    byte[] outBuffer= s.getBytes();

    try {
        outputStream.write(outBuffer);
    } catch (IOException e) {
        Log.e(TAG, "Write failed.", e);
    }
}

public boolean streaming() {
    return ( (inputStream!=null) && (outputStream!=null) );
}

public String read() {
    if (!streaming()) return "";

    String inStr="";

    try {
        if (0 < inputStream.available()) {
            byte[] inBuffer = new byte[1024];
            int bytesRead = inputStream.read(inBuffer);

```

TBlue.java

```
        inStr = new String(inBuffer, "ASCII");
        inStr=inStr.substring(0, bytesRead);
        Log.i(TAG, "byteCount: "+bytesRead+ ", inStr: "+inStr);
    }
    } catch (IOException e) {
        Log.e(TAG, "Read failed", e);
    }
    }
    return inStr;
}

public void close() {
    Log.i(TAG, "Bluetooth closing... ");
    try {
        socket.close();
        Log.i(TAG, "BT closed");
    } catch (IOException e2) {
        Log.e(TAG, "Failed to close socket. ", e2);
    }
}
}
```