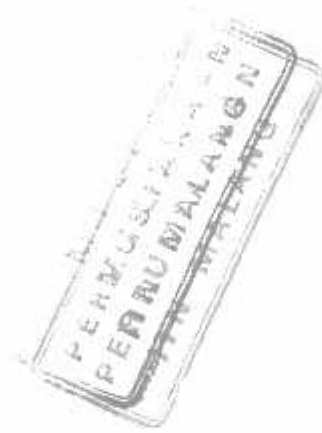


SISTEM MONITORING ATTITUDE HELIKOPTER RC MENGUNAKAN ANDROID SMARTPHONE MELALUI WIFI

SKRIPSI



**Disusun Oleh :
BOBBY ANANDA MARTIYAN
NIM. 09.12.526**

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

LEMBAR PERSETUJUAN

**SISTEM MONITORING ATTITUDE HELIKOPTER RC
MENGUNAKAN ANDROID SMARTPHONE MELALUI WIFI**

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Strata Satu (S-1)*

Disusun oleh :

BOBBY ANANDA MARTIYAN

NIM. 09.12.526

Mengetahui,



Ketua Program Studi Teknik Elektro S-1

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

Diperiksa dan Disetujui

**Mengetahui
Pembimbing I**

A handwritten signature in black ink, appearing to be 'Arvuanto'.

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

**Mengetahui
Pembimbing II**

A handwritten signature in black ink, appearing to be 'Yuli Wahyuni'.

Yuli Wahyuni, ST, MT
NIP.P.1031200456

**PRODI 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 : Bobby Ananda M
NIM : 09.12.526
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer

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, 30 Juli 2013

Yang membuat Pernyataan,



Bobby Ananda M
Nim : 09.12.526

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAK	ii
KATA PENGANTAR	iii
LEMBAR PERNYATAAN	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	1
1.3. Tujuan Pembahasan	2
1.4. Batasan Masalah	2
1.5. Metodologi Penelitian.....	2
1.6. Sistematika Penulisan	3
BAB II LANDASAN TEORI	
2.1. Android Operating Sistem	4
2.1.1. Accelerometer Android Sensor	5
2.1.2. Prinsip kerja Accelerometer	10
2.2. Sejarah Pengembangan Android OS	11
2.3. Jaringan Wifi.....	13
2.4. Pengertian Wifi	14
2.5. IDE Eclipse	15
2.5.1. Pengertian Eclipse	15
2.5.2. Arsitektur Eclipse	16
2.6. Pengertian Aplikasi Eclipse.....	18
2.7. Helikopter RC	21
2.8. Pengenalan Netbeans 7.....	24
2.9. Notebook	26

BAB III PERENCANAAN DAN PEMBUATAN ALAT	
3.1. Analisa Sistem	27
3.2. Blog Diagram Sistem.....	28
3.3. Perencanaan Sistem	29
3.3.1. Perencanaan Sistem Hardware	29
3.3.2. Perencanaan Sistem Software	30
3.4. Flowchart Program	35
BAB IV HASIL DAN ANALISA HASIL	
4.1. Implementasi Sistem	38
4.2. Pengujian aplikasi pada PC.....	39
4.3. Pengujian aplikasi pada smartphone	42
4.3.1. Pengujian Wifi.....	42
4.3.2. Pengujian Koneksi ke server.....	43
4.3.3. Pengujian Setting Port	43
4.3.4. Pengujian Run dan Stop koneksi ke server	44
4.3.5. Pengujian receive data	44
4.4. Pengujian Alat.....	45
4.4.1. Pengujian arah gerakan accelometer	46
4.4.2. Pengujian simulasi pergerakan helikopter (animasi).....	46
BAB V PENUTUP	
5.1. Kesimpulan	50
5.2. Saran	50
DAFTAR PUSTAKA	51
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Smartphone Sony Ericsson Xperia Mini Pro.....	5
Gambar 2.2 Desain accelerometer.....	10
Gambar 2.3 Tampilan Wifi.....	14
Gambar 2.4 Select a workspace.....	18
Gambar 2.5 Update SDK	19
Gambar 2.6 Browse file SDK	19
Gambar 2.7 Prosesing update SDK.....	20
Gambar 2.8 Update direktori.....	20
Gambar 2.9 Browsing direktori.....	21
Gambar 2.10 Proses selesai	21
Gambar 2.11 Helicopter RC.....	24
Gambar 2.12 Tampilan utama Netbeans 7.....	24
Gambar 2.13 Tools Pada aplikasi Utama Netbeans 7.....	25
Gambar 3.1 Desain Sistem	28
Gambar 3.2 Proses pengiriman data.....	29
Gambar 3.3 Pemrosesan Koneksi Aplikasi.....	30
Gambar 3.4 Tampilan Utama Aplikasi Eclipse.....	31
Gambar 3.5 Tampilan utama netbeans7.....	32
Gambar 3.6 Desain tampilan aplikasi pada PC.....	32
Gambar 3.7 Desain tampilan aplikasi pada Android.....	34
Gambar 3.8 Flowchart Server/HP.....	35
Gambar 3.9 Flowchart Client	36
Gambar 3.10 Flowchart sistem aplikasi	37
Gambar 4.1 Tampilan Utama Aplikasi.....	41
Gambar 4.2 Proses koneksi.....	41
Gambar 4.3 Koneksi berhenti.....	42
Gambar 4.4 Access Point Huawei.....	42
Gambar 4.5 Tampilan utama Android.....	43

Gambar 4.6	Pengaturan Wifi.....	43
Gambar 4.7	Searching ip adress	44
Gambar 4.8	Proses setting port.....	44
Gambar 4.9	Server Running	45
Gambar 4.10	Server Off	45
Gambar 4.11	Proses receive data	46
Gambar 4.12	Tampilan animasi datar	47
Gambar 4.13	Heli bergerak serong kiri	48
Gambar 4.14	Heli bergerak serong Kanan	48
Gambar 4.15	Heli bergerak ke kiri	49
Gambar 4.16	Heli bergerak ke kanan	49
Gambar 4.17	Heli bergerak turun	50
Gambar 4.18	Heli bergerak naik	50

DAFTAR TABEL

	Halaman
Gambar 4.1 Arah gerakan Accelometer.....	47

SISTEM MONITORING ATTITUDE HELIKOPTER RC MENGUNAKAN ANDROID SMARTPHONE MELALUI WIFI

Bobby Ananda Martiyan

(09.12.526)

Dosen pembimbing

Dr. Eng. Aryuanto Soetedjo, ST, MT

Yuli Wahyuni, ST, MT

Abstrak

Perkembangan teknologi wireless seperti wifi pada smartphone sudah sangat berkembang pesat dan telah diimplimentasikan di berbagai bidang. Salah satunya seperti pemanfaatan sebagai penghubung atau perantara antara dua buah perangkat agar dapat saling melakukan pertukaran data melalui jaringan tanpa kabel (wifi). Dengan teknologi yang ada pada smartphone android penulis ingin mengembangkan pengontrolan gerakan menggunakan sensor accelometer dan jaringan wifi yang ada pada smartphone android dengan memanfaatkan perubahan gravitasi yang dikendalikan oleh helikopter RC.

Dalam pemebahasan ini mengenai monitoring attitude dengan memanfaatkan sensor accelometer yang ada di smartphone android. Pengontrolan gerakan dilakukan dengan Helikopter RC dan mengirimkan perubahan nilai acceometer kepada Pc/Laptop melalui jaringan wifi. Semakin tinggi derajat smartphone undroid maka semakin cepat proses pengiriman sensor. Program aplikasi ini diharapkan bisa memberikan masukan bagi perkembangan Monitoring jarak jauh dengan menggunakan teknologi yang lebih maju.

Kata kunci : Smartphone Android, sensor accelometer, wifi.

KATA PENGANTAR

Dengan mengucapkan syukur kehadiran Tuhan Yang Maha Esa yang dengan segala Kasih dan Anugerah-Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi berjudul: **"SISTEM MONITORING ATTITUDE HELIKOPTER RC MENGGUNAKAN ANDROID SMARTPHONE MELALUI WIFI "**. Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata I di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT. selaku rektor ITN Malang
2. Bapak Ir. H. Anang subardi, MT. selaku Dekan Fakultas Teknologi Industri.
3. Bapak M. Ibrahim Ashari, ST, MT. selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT. selaku Sekretaris Jurusan Teknik Elektro S-1 ITN Malang.
5. Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT. selaku Dosen Pembimbing I.
6. Ibu Yuli Wahyuni, ST, MT. selaku Dosen Pembimbing II.
7. Dan semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan. Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, agustus 2013

Penulis

BIOGRAFI PENULIS

Bobby ananda martiyan lahir pada tanggal 9 maret 1991 di kabupaten lamongan. Merupakan anak dari dua bersaudara bapak suyatno dan ibu suhartini. Penulis memulai pendidikan pada tahun 1996. Kemudian penulis melanjutkan pendidikannya di sd negeri jetis 3 lamongan dan lulus tahun 2002. Pada tahun 2003 penulis mengenyam pendidikan di smp negeri 1 lamongan sampai tahun 2006. Kemudian melanjutkan pendidikannya di sma negeri 1 lamongan 2006-2009. Setelah lulus sma penulis melanjutkan sekolah di institut teknologi nasional malang pada tahun 2009 jurusan teknik elektro S-1 konsentrasi komputer dan lulus pada tahun 2013.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Helikopter RC (*Remote-Controlled*) adalah Replika Helikopter dalam skala seperti sekian dari aslinya, memiliki sumber tenaga sendiri sehingga bisa terbang dengan karakteristik layaknya helicopter sungguhan, dan dikendalikan dilakukan dari jarak jauh oleh sang pilot menggunakan *Remote-Controlled*.

Penulis berkeinginan membangun sistem monitoring attitude helikopter rc menggunakan *android smart phone* melalui wifi. Hal ini dimaksudkan agar kita tidak harus memantau helikopter RC (*Remote-Controlled*) tersebut secara langsung di udara dan kita dapat memantau melalui Laptop. Sehingga melalui Laptop kita dapat mengetahui 6 sumbu gerakan yaitu 3 sumbu linier (atas - bawah, kanan - kiri, depan – belakang) dengan menggunakan *accelerometer* yang berada di dalam *android smart phone*. Keunggulan lagi dari kombinasi ini adalah akan didapatkan output gambar yang tiap detil gerakannya lebih halus dari pada perangkat handphone yang hanya menggunakan accelerometer. data dikirim secara realtime yang diterima oleh Laptop.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka disimpulkan permasalahan yang akan dituangkan dalam skripsi ini yaitu:

1. Bagaimana merancang dan membangun perangkat lunak dari Sistem Monitoring Attitude Helikopter Rc Menggunakan *Android Smart Phone* Melalui Wifi.
2. Bagaimana cara mengetahui nilai dari x, y dan z pada helicopter Remote - Controller menggunakan *accelerometer* yang berada di *android smartphone*
3. Bagaimana mengkomunikasikan android ke *Laptop* dengan benar.

1.3 Tujuan

Tujuan dari sistem monitoring attitude helikopter RC adalah untuk mengetahui pergerakan helikopter RC pada setiap terbang diudara secara langsung (real time) melalui laptop.

1.4 Batasan Masalah

Agar dalam pembuatan sebuah sistem monitoring ini mengarah sesuai tujuan yang diinginkan, maka dalam pembahasan dibatasi oleh beberapa hal:

1. *Android Smart Phone* sebagai sistem monitoring attitude.
2. Aplikasi untuk membangun sistem monitoring pada *android smart phone* adalah *Eclipse*.
3. Tidak mengukur jarak ketinggian pada Helikopter dan PC.

1.5 Metodologi

Metode yang digunakan dalam penyusunan skripsi ini adalah:

1. Kajian Literatur
Pengumpulan data dan informasi yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan pada perancangan alat.
 2. Studi analisa alat
Dimaksudkan untuk melakukan analisa dan pengujian alat yang telah dirancang apakah sesuai antara fungsi dengan kerja yang diharapkan.
 3. Perancangan dan pembuatan alat
Membuat diagram blok rangkaian yang sesuai dengan rencana kerja, yang kemudian direalisasikan dengan masalah perancangan dan pembuatan berdasarkan diagram blok rangkaian yang telah disusun.
 4. Pengujian Alat
Dari hasil perancangan yang dibuat kemudian dilakukan pengujian perangkat keras dan perangkat lunak komunikasi sebelum dilakukan integrasi.
 5. Pelaporan hasil pengujian serta kesimpulan.
-

1.6 Sistematika Penulisan.

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

BAB I : PENDAHULUAN

Berisi tentang latar belakang rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Membahas tentang dasar teori mengenai permasalahan yang berhubungan dengan penelitian.

BAB III : PERANCANGAN DAN ANALISA

Berisi mengenai tahap-tahap Sistem Monitoring Attitude Helikopter Rc Menggunakan *Android Smart Phone* Melalui Wifi. Hal ini meliputi perancangan arsitektur perangkat keras dan perangkat lunak

BAB VI : PEMBUATAN DAN PENGUJIAN

Berisi tentang pembahasan langkah-langkah pembuatan alat serta pengujian terhadap alat tersebut.

BAB V : PENUTUP

Berisi tentang semua kesimpulan yang berhubungan dengan penulisan skripsi, dan saran yang digunakan sebagai pertimbangan dalam pengembangan program selanjutnya.

BAB II

LANDASAN TEORI

2. 1. Android Operating Sistem

Android adalah system operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilis perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor system operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD). Pada system androidnya saya menggunakan android versi gingerbread. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antarmuka (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. Dengan beberapa penilaian di atas maka dapat diberikan android yang akan digunakan dalam pembuatan aplikasi dengan menggunakan android smartphone dengan spesifikasi smartphone Sony ericsson mini pro yang digunakan yaitu :

OS – Android OS v2.3 gingerbread, v4.0 ice cream sandwich

Chipset – Qualcomm MSM8255 snapdragon

CPU 1 GHz scorpion

PGU – Adreno 205

Sensors – accelerometer, proximity, compass

Messaging – sms, mms, email, push email, IM

Browser – HTML, adobe flash

GPS with A-GPS support

Type – LED backlit LCD, capacitive touchscreen, 16 M colours

Size – 320 x 480 piksel, 3.0 inci

Card slot – microSD up to 32 GB, 2 Gb included

Internal – 512 MB RAM, 400 MB storage

GPRS up to 86 kbps

EDGE up to 237 kbps

Speed – HSDPA 7.2 Mbps, HSUPA 5.76 Mbps

WLAN – Wifi 802.11 b/g/n, DLNA, Wifi hotspot

Bluetooth – v2.1 with A2DP

USB – microUSB v2.0

Bobot – 136gram



Gambar 2.1. Smartphone Sony Ericsson Xperia Mini Pro

2.1.1. Pengertian Accelerometer Android Sensor

Accelerometer adalah sebuah transduser yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi. Accelerometer juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan, mesin, dan juga bisa

digunakan untuk mengukur getaran yang terjadi di dalam bumi, getaran mesin, jarak yang dinamis, dan kecepatan dengan ataupun tanpa pengaruh gravitasi bumi. Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu. 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. Prinsip Kerja Accelerometer 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 diletakkan 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 hampir 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.

Berikut ini adalah gambar bagaimana proses accelerometer analog (dengan sistem mekanik maupun digital) bekerja :

1. Accelerometer digital yang bekerja berdasarkan temperatur
 2. Accelerometer analog yang bekerja berdasarkan sistem mekanik
 3. Tipe Accelerometer
 - Capacitive: lempengan metal pada sensor memproduksi sejumlah kapasitansi, perubahan kapasitansi akan mempengaruhi percepatan
 - Piezoelectric: kristal piezoelectric yang terdapat pada accelerometer jenis ini mengeluarkan tegangan yang selanjutnya dikonversi menjadi percepatan.
-

- Piezoresistive: lempengan yang secara resistansi akan berubah sesuai dengan perubahan percepatan.
- Hall effect: percepatan yang dirubah menjadi sinyal elektrik dengan cara mengukur setiap perubahan pergerakan yang terjadi pada daerah yang terinduksi magnet.
- Magnetoresistive: Perubahan percepatan diketahui berdasarkan resistivitas material karena adanya daerah yang terinduksi magnet
- Heat Transfer: percepatan dapat diketahui dari lokasi sebuah benda yang dipanaskan dan diukur ketika terjadi percepatan dengan sensor temperatur

4. Terminology pada sensor percepatan

- +1g, posisi diam sensor searah dengan arah vertikal bumi dan menghadap ke atas)
- 0g, posisi diam sensor searah dengan arah horizontal bumi
- +1g, posisi diam sensor searah dengan arah vertikal bumi dan menghadap ke atas
- Linearitas, selisih maksimum dari kurva antara tegangan yang dihasilkan dan gravitasi dengan garis lurus
- $Linearity = \frac{V_{out,0g} - 1/2(V_{out,+1g} + V_{out,-1g})}{1g}$ (1)
- Sensitivitas, ukuran seberapa banyak perubahan yang terjadi pada hasil output sensor berdasarkan perubahan percepatan yang dimasukkan. Satuan dari sensitivitas adalah volts/g
 $sensitivity = \frac{\Delta V_{out}}{\Delta g} = \frac{V_{out,+1g} - V_{out,-1g}}{2g}$ (2)

5. Spesifikasi Accelerometer spesifikasi dinamis:

Sensitivitas Toleransi sensitivitas Noise Amplitudo puncak Respon frekuensi Resonansi frekuensi Temperature output dari sensitivitas Range temperatur output

6. Spesifikasi elektrik:

Tegangan input Arus input Tegangan bias Waktu yang diperlukan untuk menyalakan accelerometer Pelindung

7. Spesifikasi mekanikal:

Range temperatur Berat Material untuk sensor Desain sensor Material pelapis (casing)

8. Contoh Penggunaan Accelerometer Transportasi:

Salah satu penggunaan accelerometer yang sangat umum yaitu dalam sistem airbag yang terdapat pada kendaraan, khususnya mobil. Accelerometer ini digunakan untuk mendeteksi penurunan percepatan yang sangat besar yang biasanya terjadi ketika terjadinya tabrakan antar kendaraan.

9. Bidang Medis:

10. Sport Watch,

Berupa jam tangan olahraga yang juga dapat menghitung berapa banyak langkah yang telah kita lakukan, menggunakan accelerometer untuk menghitung kecepatan dan jarak dari si pelari yang menggunakannya.

11. Science and Engineering:

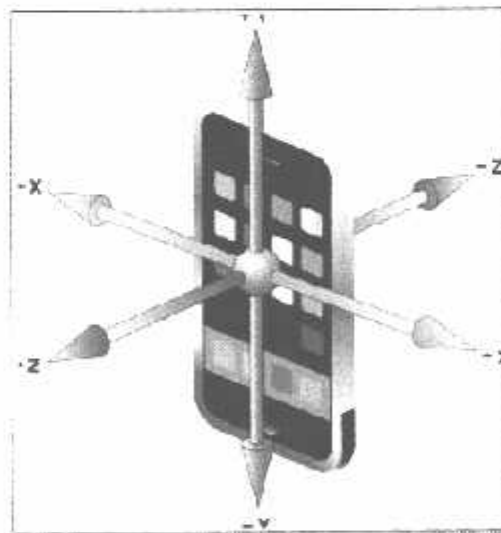
Accelerometer banyak digunakan untuk menghitung percepatan dan penurunan percepatan dari sebuah kendaraan. Accelerometer membantu untuk mengevaluasi performansi dari mesin dan sistem percepatan dan juga breaking system (sistem penurunan percepatan). Kecepatan yang biasa ditampilkan pada kendaraan anda umumnya didapatkan dari penggunaan accelerometer. Selain itu juga biasa digunakan untuk menghitung vibrasi pada kendaraan, mesin, bangunan, dan sistem keamanan pada kendaraan (safety installation). Accelerometer juga dapat mengkalkulasi percepatan yang diakibatkan oleh gravitasi bumi. Accelerometer yang menghitung gravitasi secara spesifik digunakan pada gravimetry, disebut sebagai gravimeter. Notebook atau laptop juga dilengkapi dengan accelerometer untuk mengevaluasi guncangan yang dirasakan oleh laptop tersebut.

12. Peralatan Elektronik:

Accelerometer pada laptop biasanya digunakan pada sistem Sudden Motion Sensor, yang biasa digunakan untuk mendeteksi jatuhnya laptop. Jika kondisi pada saat jatuh terdeteksi, hard disk drive yang ada akan

diproteksi sehingga tidak terjadi data loss. Sekarang ini juga terdapat notebook yang menggunakan accelerometer untuk secara otomatis mengubah arah layar (menjadi miring ataupun terbalik) sesuai dengan arah monitor tersebut ditegakkan (portrait atau landscape). Terdapat juga sejumlah handphone yang menggunakan accelerometer untuk mengubah lagu yang dimainkan (Track Switching). Camera recorder menggunakan accelerometer untuk menstabilkan gambar (image stabilization). Camera digital menggunakan accelerometer untuk menu pilihan anti blur ketika mengambil gambar.

Baru-baru ini Apple.Inc memperkenalkan sebuah gebrakan dengan mengkombinasikan 2 sensor gerakan yaitu antara *Accelerometer* dan *Gyroscope* pada sebuah perangkat handphone. Ini akan menyempurnakan fitur dari handphone yang hanya menggunakan *accelerometer* dalam mendeteksi gerakan. Dengan kombinasi ini maka akan didapatkan "6-axis motion sensing"/deteksi 6 sumbu gerakan yaitu 3 sumbu linier (atas-bawah, kanan-kiri, depan-belakang) dan 3 sumbu rotasi (rotasi roll, pitch and yaw seperti pada gambar 2). 1 keunggulan lagi dari kombinasi ini adalah akan didapatkan output gambar yang tiap detil gerakannya lebih halus dari pada perangkat handphone yang hanya menggunakan accelerometer.



Gambar 2.2. Desain accelerometer

2.1.2. Prinsip kerja Accelerometer

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 hampir 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. 2. Sejarah Pengembangan Android OS

Android dirilis pertama oleh google pada tanggal 5 november 2007, yang didukung oleh Open Handset Alliance, dan smartphone pertama yang memakai sistem operasi android adalah HTC Dream, yang dirilis pada 22 oktober 2008. pada penghujung tahun ini, android telah merajai penjualan smartphone, mengalah kan NOKIA, Iphone dan Blackberry. berikut beberapa versi android:

- a. **Android Versi 1.1** : 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.
- b. **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.

- c. **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.
 - d. **Android versi 2.0/2.1 Éclair** : 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.
 - e. **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, intergrasi 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.
 - f. **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.
 - g. **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.

- h. 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.
- i. 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 Dcletion dan tentunya fitur Undo.

2.3. Jaringan Wifi

Wi-Fi adalah teknologi jaringan nirkabel yang dapat memberikan akses internet dengan jarak hingga 100 meter, tergantung pada router Wi-Fi dan lingkungan sekitar Anda. Untuk menggunakan Wi-Fi, Anda harus tersambung ke titik akses nirkabel atau "hotspot". Beberapa hotspot tersedia untuk umum dan Anda dapat langsung menyambung ke hotspot itu. Sedangkan yang lain menerapkan fitur keamanan yang memerlukan penyiapan langkah lain, seperti sertifikat digital atau cara lain untuk memastikan bahwa hanya pengguna terotorisasi yang dapat tersambung.

2.4. Pengertian Wifi

Dalam ilmu jaringan komputer, pengertian Wireless perangkat keras yang memungkinkan perangkat wireless lain (seperti laptop, ponsel) untuk terhubung ke jaringan kabel menggunakan Wi-fi, bluetooth atau perangkat standar lainnya. Wireless umumnya dihubungkan ke router melalui jaringan kabel (kebanyakan telah terintegrasi dengan router) dan dapat digunakan untuk saling mengirim data antar perangkat wireless (seperti laptop, printer yang memiliki wifi) dan perangkat kabel pada jaringan. Fungsi Wifi berfungsi sebagai pengatur lalu lintas data, sehingga memungkinkan banyak Client dapat saling terhubung melalui jaringan. Sebagai Hub/Switch yang bertindak untuk menghubungkan jaringan lokal dengan jaringan wireless/nirkabel, Access point dapat memancarkan atau mengirim koneksi data / internet melalui gelombang radio, ukuran kekuatan sinyal juga mempengaruhi area coverage yang akan dijangkau, semakin besar kekuatan sinyal (ukurannya dalam satuan dBm atau mW) semakin luas jangkauannya.

Beberapa spesifikasi access point.

Deskripsi :

Higher speed :7.2M DL/ 5.76 UL, Longer battery life :1500 mAH , 4.5 hrs working time, OLED Display

OS :

Windows, MAC OS, 3G / 2G : Works with all 3G / 2G SIMs

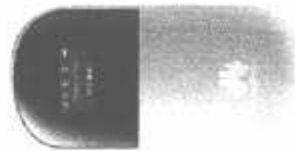
Features:

HSDPA : 7.2 Mbps(DL), HSUPA :5.76Mbps(UL), Plug & Play, Fully functional WiFi router with advanced routing functions,\ WPS Data Specificatins :

HSUPA/HSDPA/WCDMA, 2100MHz

Type : E560 , SM/GPRS/EDGE:900/1800/1900/850MHz, WLAN 2.401

GHz- 2.495GHz, USB Stick Micro-SD card slot (up to 4-32G)



Gambar 2.3. Tampilan Wifi

Penerapan Wireless Hotspot merupakan salah satu penerapan Wireless yang paling umum, dimana klien nirkabel dapat terhubung ke internet tanpa memperhatikan jaringan tertentu yang telah mereka sambungkan saat itu. Di kota besar atau di daerah tertentu hotspot umumnya disediakan dalam rumah makan, perpustakaan, stasiun, atau daerah publik lainnya yang memungkinkan banyak orang untuk dapat terus tersambung ke jaringan internet.

2. 5. IDE Eclipse

2.5.1. Pengertian Eclipse

IDE (Integrated Development Environment) adalah sebuah alat atau perangkat yang membantu dalam pembuatan sebuah program. Java membebaskan penggunanya untuk memilih IDE sendiri, diantaranya IDE Java yang sering digunakan adalah Netbeans dan Eclipse. Eclipse adalah sebuah IDE untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Eclipse memiliki komunitas open source yang bertujuan menghasilkan platform pemrograman terbuka. Eclipse terdiri dari framework yang dapat dikembangkan lebih lanjut, peralatan bantu untuk membuat dan mengatur software sejak awal hingga diluncurkan. Platform Eclipse didukung oleh ekosistem besar yang terdiri dari vendor teknologi, start-up inovatif, universitas, riset institusi serta individu. Banyak orang mengenal Eclipse sebagai IDE (Integrated Development Environment) untuk bahasa Java, tapi Eclipse lebih dari sekedar IDE untuk Java. Secara umum Eclipse digunakan untuk membangun

perangkat lunak inovatif berstandar industri, dan alat bantu beserta framework-nya membantu pekerjaan menjadi lebih mudah. Eclipse menyediakan tools yang dibutuhkan untuk membuat professional desktop, enterprise, web, dan mobile applications dengan bahasa pemrograman Java, C/C++, dan bahkan dynamic languages seperti PHP, JavaScript, Groovy, dan Ruby. Berikut ini adalah sifat dari Eclipse:

- **Multi-platform.** Menunjukkan bahwa Eclipse dapat berjalan pada sistem operasi Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- **Multi-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.
- **Multi-role.** Fungsi Eclipse selain sebagai IDE, juga untuk pengembangan aplikasi. Eclipse bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, tes perangkat lunak, pengembangan web, dan lain sebagainya.

Lisensi Eclipse menggunakan EPL (Eclipse Public License), yaitu lisensi yang memungkinkan organisasi untuk menjadikan Eclipse sebagai produk komersialnya, dan pada saat yang sama meminta orang yang melakukan perubahan untuk mengkontribusikan hasilnya kembali kepada komunitas.

2.5.2. Arsitektur Eclipse

Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Hal yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman Java. Berikut ini merupakan gambaran arsitektur dari Eclipse:

1. Eclipse Platform - Sebuah plug-in adalah unit terkecil dari fungsi Eclipse Platform yang dapat dikembangkan dan disampaikan secara terpisah. Platform runtime adalah sebuah kernel yang mengatur plug-in apa saja yang telah ter-install dan membuat register informasinya. Untuk mengurangi waktu start-up dan pemakaian resource, platform runtime tidak akan memuat plug-in jika tidak benar-benar diperlukan. Kecuali untuk kernel (Platform Runtime), semua fungsi Platform Eclipse terletak di plug-in.
 2. Workspace - Workspace (ruang kerja) adalah plug-in yang bertugas untuk mengatur sumber daya dari pengguna. Sumber daya di sini meliputi project yang dibuat pengguna, berkas yang berada di dalamnya, serta perubahan yang terjadi pada berkas dan project pada sumber daya lainnya. Workspace juga bertugas memberitahu plug-in mengenai perubahan yang ada, seperti jika ada berkas yang dibuat, dihapus, atau diubah. Berbagai alat dicolokkan ke dalam Eclipse Platform beroperasi pada file biasa dalam ruang kerja pengguna. Workspace ini terdiri dari satu atau lebih project, di mana setiap project berada pada direktori yang ditentukan pengguna yang sesuai dalam sistem file.
 3. Workbench - Struktur platform Eclipse memiliki workbench (meja kerja) yang bertugas menyediakan tampilan antar-muka (user interface) untuk melakukan interaksi dengan pengguna. API (Application Programming Interface) pada workbench dibangun dari dua komponen berikut:
 - a. SWT (Standard Widget Toolkit) - satu set widget dan perpustakaan grafis yang terintegrasi dengan sistem window asli tapi dengan OS API independen.
 - b. JFace - seperangkat alat (toolkit) yang diimplementasikan menggunakan SWT untuk menyederhanakan tugas pemrograman user interface.
 4. Team support - Platform Eclipse memungkinkan sebuah project di ruang kerja ditempatkan di bawah versi dan manajemen konfigurasi
-

repositori dengan tim terkait. Platform memiliki poin ekstensi dan API penyedia repositori yang memungkinkan jenis baru dari repositori tim dipasang masuk. Fungsi yang disediakan oleh repositori produk tim tertentu selalu mempengaruhi alur kerja pengguna, misalnya, dengan menambahkan langkah-langkah yang jelas untuk mengambil file dari repositori, untuk mengembalikan file hasil update ke repositori, dan untuk membandingkan versi file yang berbeda. Secara umum, plug-in tidak membutuhkan interaksi dengan komponen tim pendukung (team support) jika mereka tidak menyediakan layanan kontrol versi.

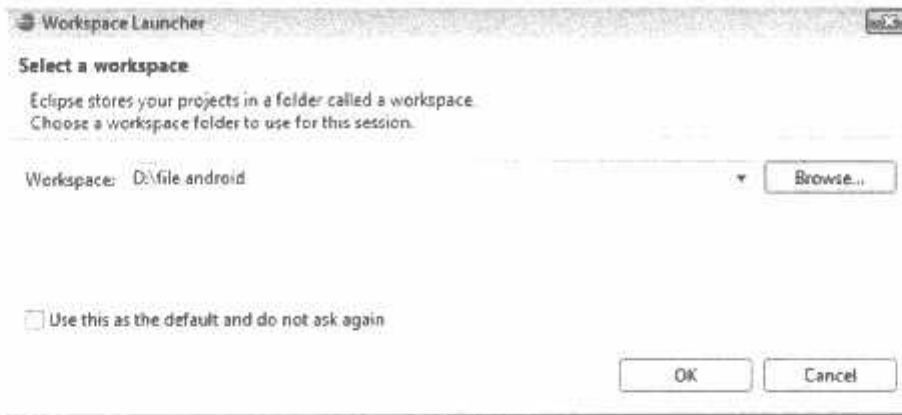
5. Help - Komponen bantuan (help) merupakan tambahan dari platform Eclipse, dan memiliki fungsi yang sama dengan plug-in pada Eclipse. Help menyediakan struktur navigasi tambahan yang memungkinkan pemrogram untuk menambahkan dokumentasi dalam bentuk berkas HTML. Platform Eclipse menggunakan server dokumentasi internal sendiri untuk menyediakan halaman web yang sebenarnya berasal dari dalam dokumen web.

2. 6. Pengertian Aplikasi Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). jadi dengan software eclipse kita dapat men-develop atau membuat sebuah program tidak hanya di android, kita bisa membuat program2 lain semisal berplatform java atau yang lainnya. disini menggunakan Eclipse karena software ini merupakan software free dan OpenSource, persis dengan background yang dimiliki masbudi yang terlahir dari komunitas opensource. langsung saja gan, mari kita belajar tentang eclipse.

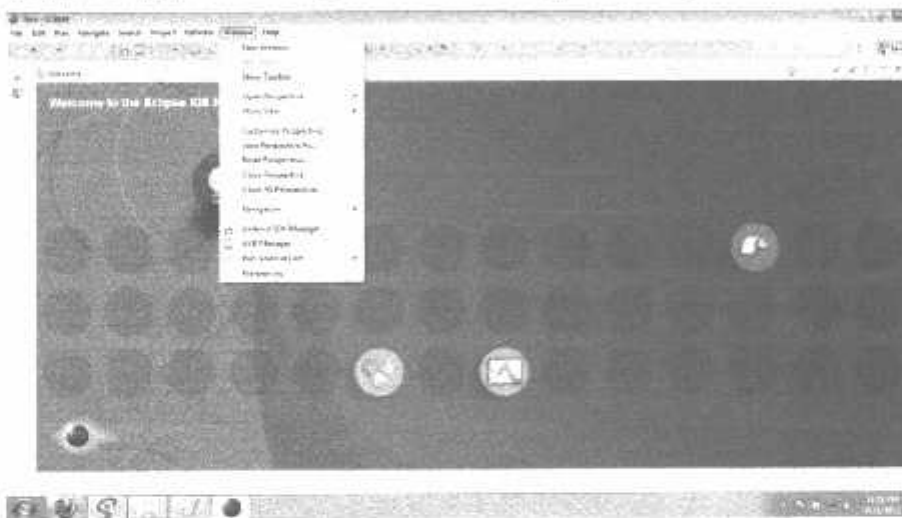
Instalasi Eclipse Untuk Android

1. Langkah pertama, buka eclipse kemudian akan muncul tampilan seperti dibawah ini, tampilan dibawah ini merupakan tampilan dimana kita akan menyimpan file kita, kemudian klik OK.



Gambar 2.4. Select a workspace

2. kemudian klik pada tab window dan klik preferences, hal ini dilakukan untuk mengupdate SDK (Software Development Kit) untuk android.



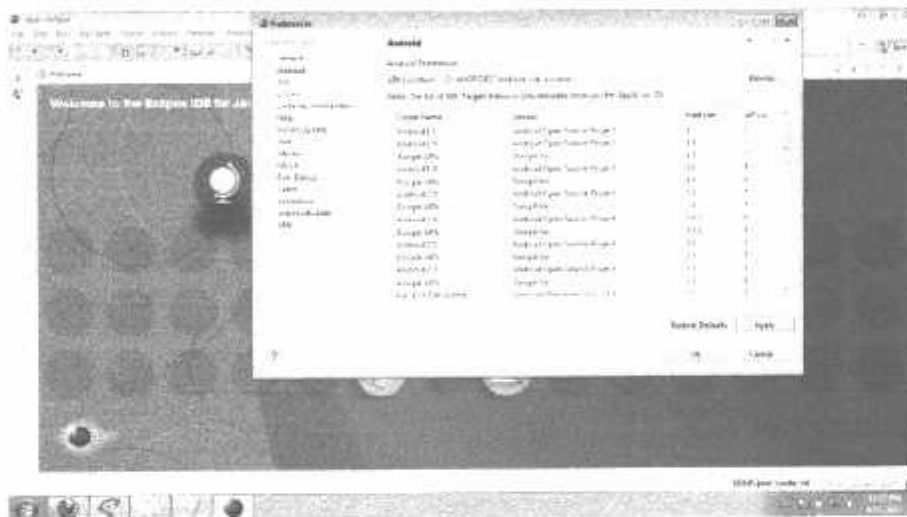
Gambar 2.5. Update SDK

3. Setelah itu, akan muncul tampilan dibawah ini : Masukkan lokasi dimana anda menyimpan file SDK nya, klik tombol Browse, dan apabila sudah ditemukan klik tombol OK



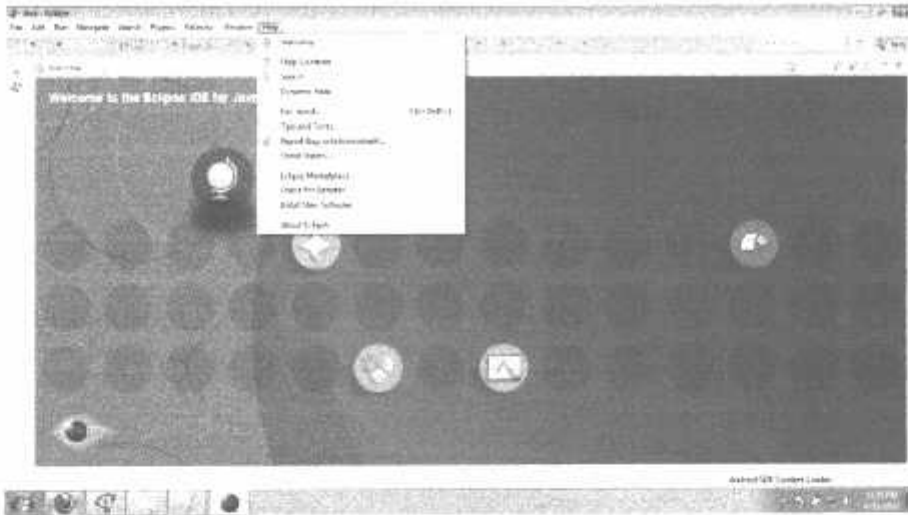
Gambar 2.6. Browse file SDK

4. Selanjutnya akan muncul tampilan seperti berikut : klik OK



Gambar 2.7. Prosesing update SDK

5. Dengan seperti itu, berarti anda sudah menginstall SDK nya, dan kemudian kita harus mengupdate ADT nya, langkah nya sebagai berikut :
Klik Tab Help, kemudian pilih Install New Software



Gambar 2.8. Update direktori

6. Kemudian akan muncul seperti tampilan dibawah ini, dan Klik Add, dan akan muncul tampilan ADD Repository. Name : bisa di isi bebas Kemudian pada Location klik archive , kemudian cari dimana file adt nya anda simpan, kemudian klik OK



Gambar 2.9. Browsing direktori

7. Apabila muncul tampilan dibawah ini, berarti Eclipse anda sudah terinstall ADT yang terbaru.



Gambar 2.10. Proses selesai

2.7. Helikopter RC

Helikopter RC (Remote Control) adalah Sebuah mainan yang beredar di pasaran umum sudah banyak sekali dengan model yang beraneka ragam pula. tetapi Helikopter RC (Remote Control) memiliki beberapa tingkatan kecepatan atau pengendalian. Sebelum mengenal lebih dalam system RC Helikopter/Remote control Helikopter, kita harus mengenal lebih dulu apa yang dimaksud dengan RC Heli 2ch, 3ch, 3.5ch, 4ch dan 6 channel tersebut RC Heli 2 channel adalah.

RC Heli yang dapat digerakan dengan 2 system yaitu :

- a. Gerakan Naik dan Turun
- b. Gerakan Belok Kiri dan Belok Kanan Maka RC Helicopter tersebut dapat digerakan/dikemudikan dengan Remote Control tetapi hanya ada 2 fungsi di remote controlnya.

RC Heli 3 channel adalah RC Heli yang dapat digerakan dengan 3 system yaitu :

- a. Gerakan Naik dan Turun
- b. Gerakan Belok Kiri dan Belok Kanan
- c. Gerakan Maju dan Mundur Maka RC Helicopter tersebut dapat digerakan/dikemudikan dengan Remote Control tetapi hanya ada 3 fungsi di remote controlnya.

RC Heli 3.5 channel adalah RC Heli yang dapat digerakan dengan 3.5 system yaitu:

- a. Gerakan Naik dan Turun
- b. Gerakan Belok Kiri dan Belok Kanan
- c. Gerakan Maju dan Mundur dan Gyroscope. Maka RC Helicopter tersebut dapat digerakan/dikemudikan dengan Remote Control tetapi hanya ada 3.5 fungsi di remote controlnya disertai pengatur balance kiri atau kanan agar tetap stabil.

RC Heli 4 channel adalah RC Heli yang dapat digerakan dengan 4 system yaitu :

- a. Gerakan Naik dan Turun
- b. Gerakan Belok Kiri dan Belok Kanan
- c. Gerakan Maju dan Mundur
- d. Serong Kiri dan Serong Kanan. Maka RC Helicopter tersebut dapat digerakan/dikemudikan dengan Remote Control tetapi hanya ada 4 fungsi di remote controlnya. Ada yang sudah disertai dengan LCD Display untuk mengetahui posisi Helicopter.

RC Heli 6 channel adalah RC Heli yang dapat digerakan dengan 6 system yaitu :

- a. Gerakan Naik dan Turun
- b. Gerakan Belok Kiri dan Belok Kanan
- c. Gerakan Maju dan Mundur
4. Serong Kiri dan Serong Kanan
5. Posisi Helicopter Terbalik

Posisi Helicopter berdiri. Maka RC Helicopter tersebut dapat digerakan/dikemudikan dengan Remote Control tetapi hanya ada 6 fungsi di remote controlnya. Biasanya disertai Trim untuk setingan pada setiap gerakan, berikut LCD display untuk mengetahui posisi Helicopter dan umumnya Remote control 6 channel sudah bis dwi fungsi yaitu untuk mengendalikan Helicopter dan sebagai Simulator pada komputer. Berikut RC yang digunakan untuk perhitungan Accelerator.

Specification:

1. Main rotor diameter: 520mm
-

2. Weight: 305g, Length: 52cm, hight: 15.5cm, width: 8.5cm
3. Power system: 370 motor
4. Attitude: 100m flaying
5. Transmitter: standard 4CH
6. Mix controller: 4in1 controller (W/Gyro,mixer,ESC,receiver)
7. Digital Servo Digital: 8.0g,1.3kg.CM,0.12S/60°
8. Battery: 7.4V-850mAh (included)
9. Included: Simulator Cable + CD + Neckstrap + Training Gear
10. Time: about 8 minutes

Features:

- WL Toys V912
- Large single blade RC helicopter with Gyro
- Transmitter with LCD screen
- Ready for flight out of box except 8 "AA" battery (sold separately)



Gambar 2.11. Helicopter RC

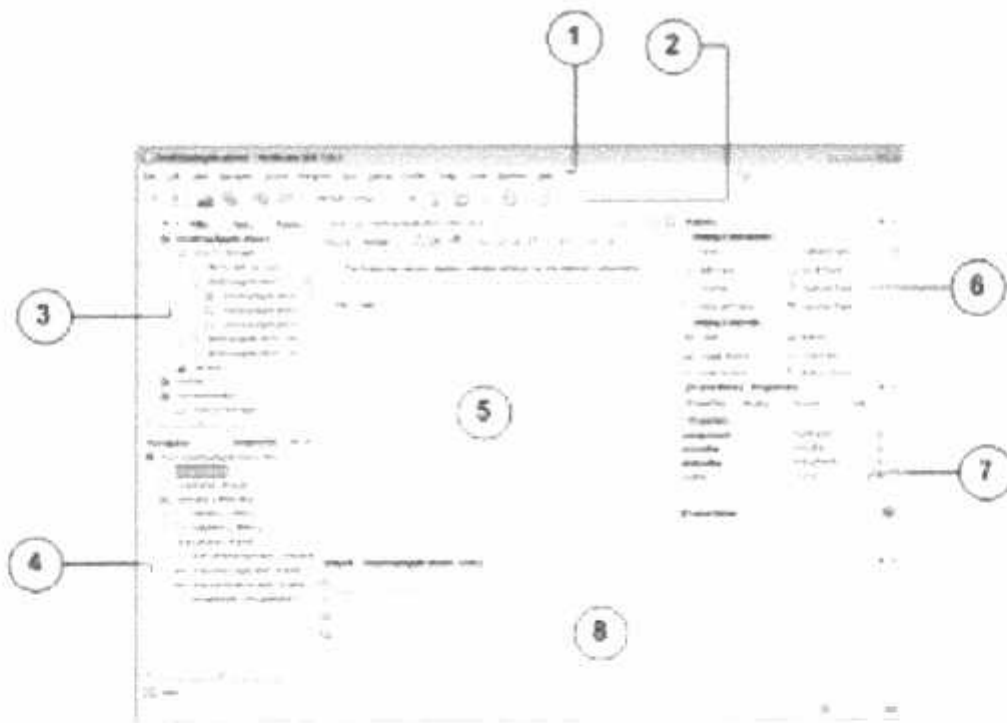
2. 8. Pengenalan Netbeans 7

IDE NetBeans adalah sebuah lingkungan pengembangan, sebuah tools untuk programmer menulis, mengompilasi, mencari kesalahan dan menyebarkan program. IDE NetBeans ditulis dalam Java, namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas IDE NetBeans. IDE NetBeans adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. Berikut adalah tampilan utama Netbeans 7.



Gambar 2.12. Tampilan utama Netbeans 7

NetBeans mengacu pada dua hal, yakni platform untuk pengembangan desktop java, dan sebuah Integrated Development Environment (IDE) yang dibangun menggunakan platform NetBeans. Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen-komponen perangkat lunak modular yang disebut 'modul'. Sebuah modul adalah suatu arsip Java (Java Archive) yang membuat kelas-kelas Java untuk berinteraksi dengan NetBeans Open API dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan powerful. Berikut Tolls yang terdapat dalam tampilan netbeans 7



Gambar 2.13. Tools Pada aplikasi Utama Netbeans 7

Lingkungan pengembangan yang terintegrasi pada IDE NetBeand, memudahkan pengguna untuk membuat beragam aplikasi yang mudah.

1. Menu Bar
2. Toolbar
3. Project Explorer
4. Daftar Komponen yang digunakan
5. Jendela Utama
6. Kontrol Komponen (Componen Pallete)
7. Jendela Properties
8. Jendela Keluaran (debuging)

2. 9. Notebook

Laptop atau Notebook adalah komputer bergerak yang berukuran relatif kecil dan ringan, beratnya berkisar dari 1-6 kg, tergantung ukuran, bahan, dan spesifikasi laptop tersebut. Sumber daya laptop berasal dari baterai atau adaptor A/C yang dapat digunakan untuk mengisi ulang baterai dan menyalakan laptop itu sendiri. Baterai laptop pada umumnya dapat bertahan sekitar 1 hingga 6 jam sebelum akhirnya habis, tergantung dari cara pemakaian, spesifikasi, dan ukuran

baterai. Laptop terkadang disebut juga dengan komputer notebook atau notebook saja. Sebagai komputer pribadi, laptop memiliki fungsi yang sama dengan komputer desktop (desktop computers) pada umumnya. Komponen yang terdapat di dalamnya sama persis dengan komponen pada desktop, hanya saja ukurannya diperkecil, dijadikan lebih ringan, lebih tidak panas, dan lebih hemat daya.

Komputer jinjing kebanyakan menggunakan layar LCD (Liquid Crystal Display) berukuran 10 inci hingga 17 inci tergantung dari ukuran laptop itu sendiri. Selain itu, papan ketik yang terdapat pada laptop juga kadang-kadang dilengkapi dengan papan sentuh yang berfungsi sebagai "pengganti" tetikus. Papan ketik dan tetikus tambahan dapat dipasang melalui soket Universal Serial Bus maupun PS/2 jika tersedia. Berbeda dengan komputer desktop, laptop memiliki komponen pendukung yang didesain secara khusus untuk mengakomodasi sifat komputer jinjing yang portabel. Sifat utama yang dimiliki oleh komponen penyusun laptop adalah ukuran yang kecil, hemat konsumsi energi, dan efisien. Komputer jinjing biasanya berharga lebih mahal, tergantung dari merek dan spesifikasi komponen penyusunnya, walaupun demikian harga komputer jinjing pun semakin mendekati desktop seiring dengan semakin tingginya tingkat permintaan konsumen.

BAB III PERENCANAAN DAN PEMBUATAN ALAT

Dalam bab ini akan dibahas perancangan dan pembuatan alat. Pembahasan akan dilakukan pada setiap blok rangkaian, cara kerja masing-masing blok rangkaian, perhitungan dan fungsi masing-masing blok rangkaian tersebut. Secara garis besar terdapat dua bagian perangkat yang ada yaitu :

- Perancangan perangkat keras (*Hardware*).
- Perancangan perangkat lunak (*Software*).

Pada perancangan perangkat keras akan meliputi seluruh *peripheral* yang digunakan pada sistem ini. Pada perancangan perangkat lunak akan meliputi diagram alir dan *software* secara umum, akan tetapi kedua perangkat ini dalam kerjanya akan saling menunjang satu sama lain.

3.1. Analisa Sistem

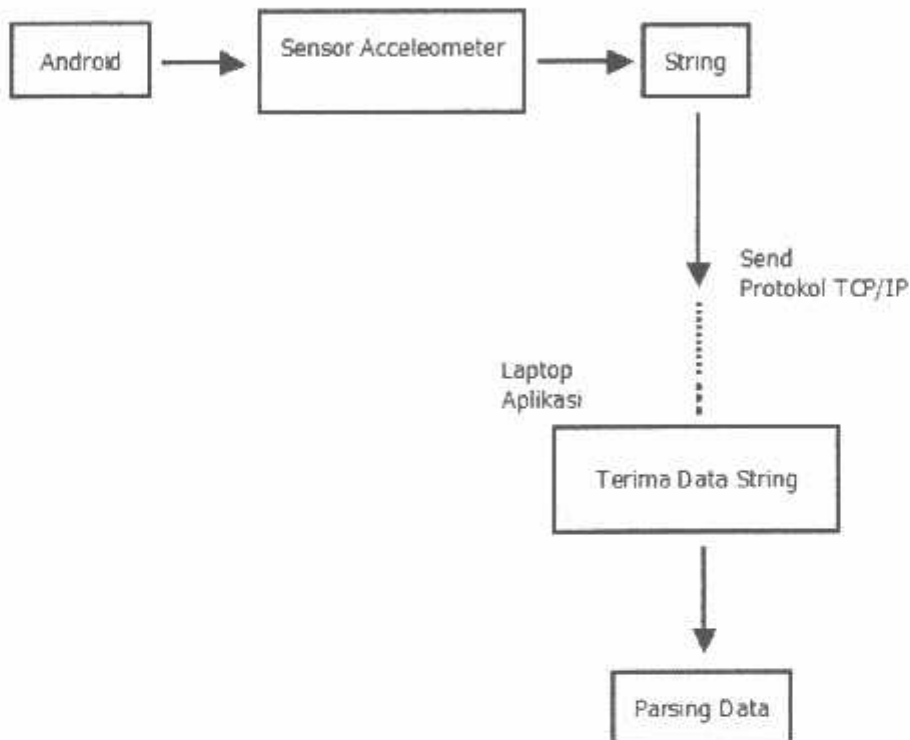
Sistem pengontrol *accelerometer* pada helicopter RC adalah sebuah pengontrol yang digunakan untuk memonitoring sebuah akselerasi helicopter RC. Dalam sistem ini akan menggunakan Sinyal Wifi yang nantinya akan dikontrol *accelerometer* pada notebook/laptop yang akan menggunakan aplikasi Netbeans 7 Yang akan menjadi perantara antara smartphone android dengan aplikasi delphi yang akan mengirim dan menerima data dari keduanya. Dan system ini juga menggunakan Smartphone Android pengontrol yang sudah memiliki jaringan *wireless* dan sensor Accelometer. Disini *Wireless* sebagai penghubung antara Smartphone Android dengan Aplikasi pada laptop yang sudah terpasang module yang dihubungkan hardware access point , dan sensor Accelometer yang bisa digunakan untuk mengukur percepatan termasuk percepatang gravitasi bumi.

4. **Laptop** berfungsi sebagai pengendali atau pengontrolan terhadap hasil dari sensor *accelometer*. Dengan menggunakan aplikasi Netbeans.

3.3. Perencanaan Sistem

3.3.1. Perencanaan Sistem hardware

Dalam perancangan Hardware dapat dijelaskan dengan diagram sebagai berikut. Koneksi dan antara hubungan Hardware dan Aplikasi Software yang digunakan.



Gambar 3.2. Proses pengiriman data

Penjelasan dari masing-masing blok pada gambar 3.2 adalah sebagai berikut.

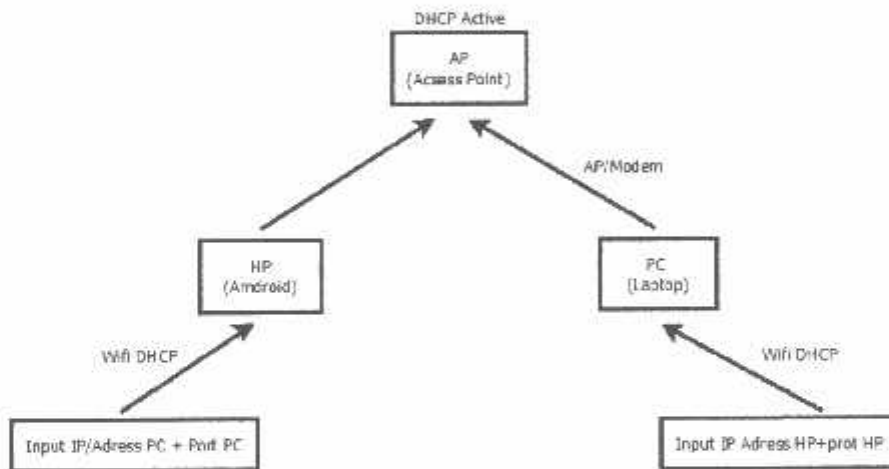
1. Smartphone Android adalah Pusat pengontrol hardware, meliputi sensor *Accelometer*. Smartphone Android dapat mengirimkan data *Accelometer* melalui jaringan wifi ke Laptop/Notebook.
2. AP (access point) adalah sebuah teknologi wireless (tanpa kabel) yang mampu menyediakan layanan komunikasi data dan suara secara real-time dengan jarak Access point dapat memancarkan atau mengirim

koneksi data / internet melalui gelombang radio, ukuran kekuatan sinyal juga mempengaruhi area coverage yang akan dijangkau, semakin besar kekuatan sinyal (ukurannya dalam satuan dBm atau mW) semakin luas jangkauannya. Access point bekerja sebagai penghubung/koneksi antara smartphone Android dengan Laptop/Notebook.

3. Aplikasi Delphi adalah aplikasi yang terdapat pada sebuah notebook yang memungkinkan terkoneksi dengan perangkat hardware yang terkoneksi dalam sebuah jaringan dengan bantuan access point dimana penerimaan yang telah *diconvert* menjadi data string.

3.3.2. Perancangan Sistem Software

Perancangan dalam perangkat aplikasi ini meliputi aplikasi yang terdapat pada smartphone android dan yang terdapat pada Laptop/Notebook, seperti gambar 3.3, berikut.



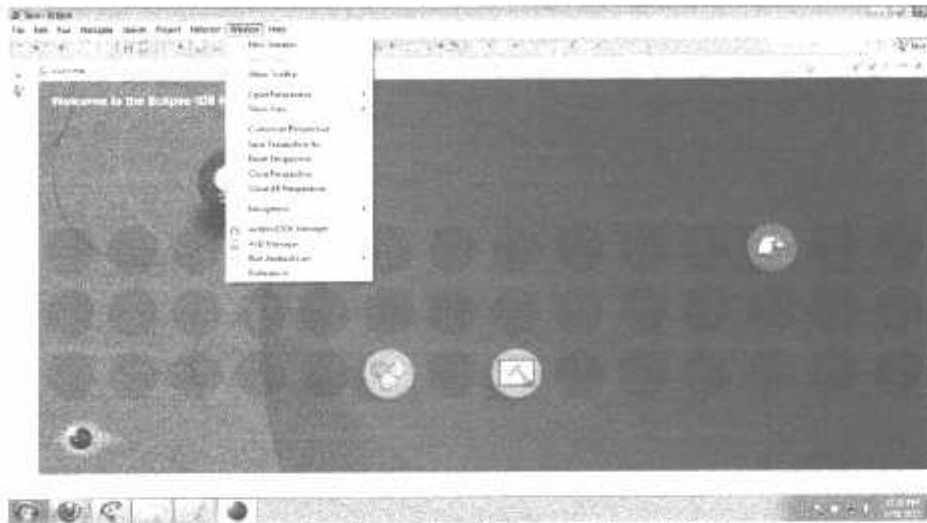
Gambar 3.3. Pemrosesan Koneksi Aplikasi

Dalam gambar di atas menunjukkan koneksi antara android dengan PC (laptop). Penjelasan masing-masing bagian gambar 3.3. :

1. PC setting IP /Adress PC sebagai client yang akan dihubungkan dengan smartphone.
2. Handphone smartphone disetting sebagai server yang akan dihubungkan pada PC yang telah di setting client, dengan jaringan Wireles.

3. Mengaktifkan DHCP dengan menggunakan Access point dimana sebagai interface dari handphone dan PC.

Pada perancangan pembuatan aplikasi pada android ini menggunakan aplikasi Eclipse yang nantinya aplikasi akan ditanam pada android dengan ekstensi .jdk. berikut tampilan utama aplikasi eclipse.



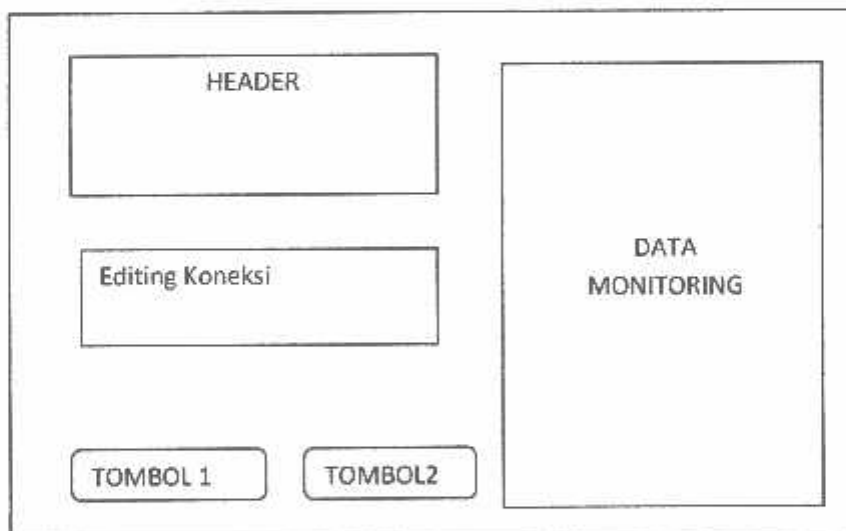
Gambar 3.4. Tampilan Utama Aplikasi Eclipse

Aplikasi eclipse ini berfungsi sebagai pembuat aplikasi yang akan digunakan pada smartphone android. Seperti pada umumnya yang akan dipasang pada playstore smartphone lainnya pada umumnya. Yang nantinya aplikasi ini akan terkoneksi pada jaringan wifi. Yang selanjutnya akan dihubungkan dengan aplikasi pada computer berikut tampilan utama aplikasi Delphi sebagai pembuatan proses monitoring pada laptop/PC.



Gambar 3.5. Tampilan utama netbeans7

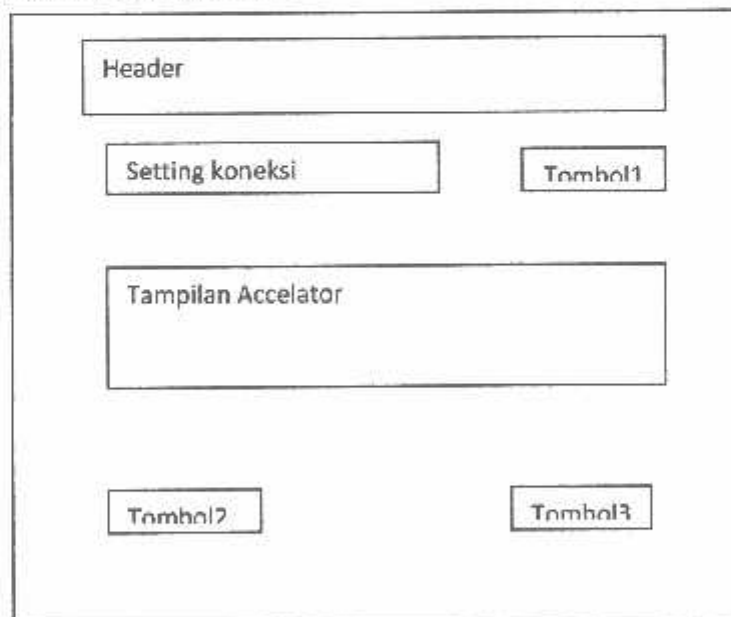
Gambar diatas menunjukkan tampilan utama aplikasi netbeans7 yang digunakan sebagai tampilan utama pemrograman sebagai monitoring pada *accelometer*, dengan sebuah tampilan dimana hasil terperinci dalam pengontrolan aplikasi tersebut. Berikut desain aplikasi pada tampilan pada PC dan Android.



Gambar 3.6. Desain tampilan aplikasi pada PC

Penjelasan desain tampilan pada aplikasi PC dimana terdapat beberapa menu sebagai berikut :

- Header : sebagai pemberitahuan tampilan utama atau judul aplikasi dipergunakan untuk user yang baru mengetahui sebuah aplikasi.
- Edit Koneksi : Pada Menu sebagai berikut menggunakan berhad mengganti data koneksi dimana terdapat Ip Adress Dan Port yang selalu di setting pada saat koneksi data aplikasi.
- Tombol 1 dan 2 : dalam perancangan aplikasi yang terdapat pada PC tersebut tombol “1” di persiapkan untuk tombol “Connect” dalam prosesnya pencari jaringan antara handphone smartphone dengan aplikasi dalam PC, memulai proses koneksi jaringan. Tombol “2” dipersiapkan sebagai “receive” dipergunakan untuk proses penerima data realtime yang akan dikirimkan pada aplikasi yang terdapat pada handphone smartphone.
- Data Monitoring : Dalam Menu sebagai berikut dimana data penerima diproses atau dimonitoring secara realtime dalam data monitoring seperti diatas. Hasil yang diterima dari proses monitoring ini adalah nilai dari X, Y, dan Z pada aplikasi Accelometer dalam handphone smartphone yang digunakan.



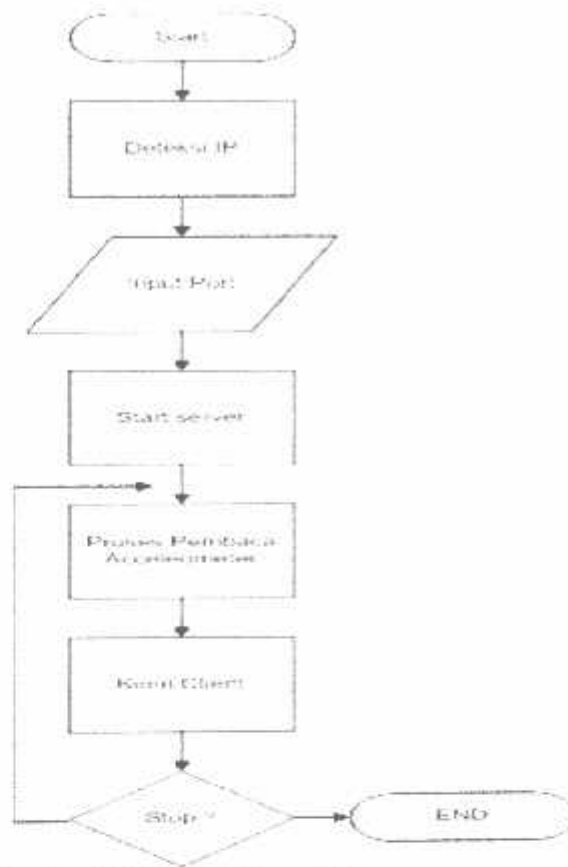
Gambar 3.7. Desain tampilan aplikasi pada Android

Penjelasan desain tampilan pada aplikasi handphone smartphone dimana terdapat beberapa menu sebagai berikut :

- Header : sebagai pemberitahuan tampilan utama atau judul aplikasi dipergunakan untuk user yang baru mengetahui sebuah aplikasi.
- Setting Koneksi : Pada Menu berikut menggunakan berhak mengganti data koneksi dimana terdapat Ip Adress Dan Port yang selalu di seting pada saat koneksi data aplikasi.
- Tombol 1 : Dipersiapkan untuk tombol “set” untuk mensetting port pada jaringan.
- Tombol 2 : Dipersiapkan untuk tombol “start” digunakan sebagai pemulai accelometer yang terdapat pada smartphone yang nantinya data akan dikirim pada jaringan wifi untuk dihubungkan ke PC secara realtime.
- Tombol 3 : Dipersiapkan untuk tombol “ Stop” tombol ini digunakan untuk memberhentikan proses transfer data monitoring.
- Tampilan accelerometer : Dalam Menu berikut dimana data penerimanya dalam data monitoring seperti diatas. Hasil yang diterima dari proses monitoring ini adalah nilai dari X, Y, dan Z pada aplikasi Accelometer dalam handphone smartphone yang digunakan.

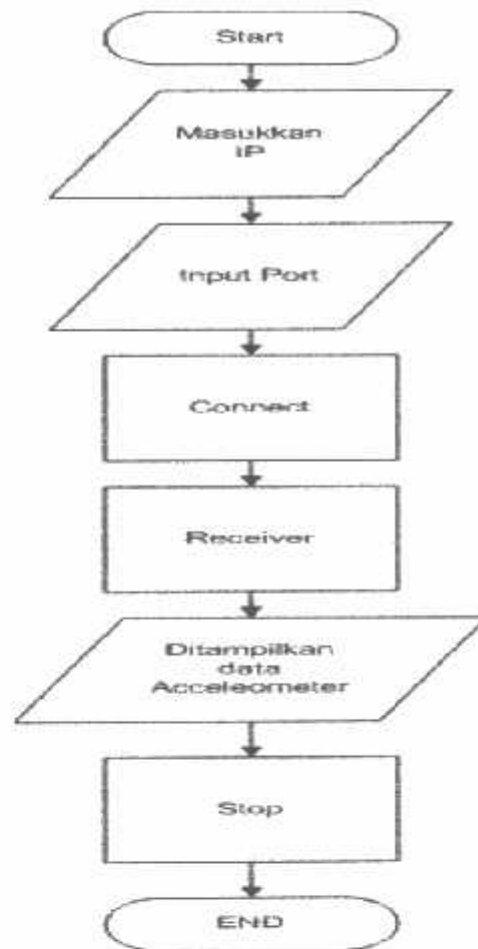
3.4. Flowchart Program

Proses perancangan monitoring accelometer ini, dipergunakan flowchart program dari server (Handphone smartphone) dan client (PC/Laptop), pada gambar 3.4. dan 3.5. berikut :



Gambar 3.8 flowchart Server/HP

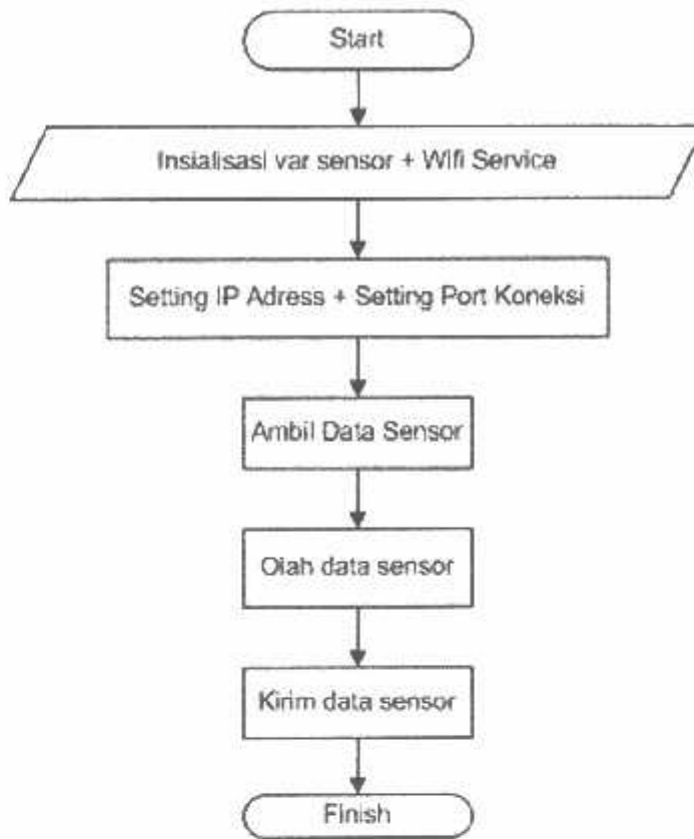
Penjelasan alur diatas merupakan pendeteksi IP yang selanjutnya penginputan port pada smartphone, memulai program dengan proses pembacaan accelerometer yang akan kirim ke client. Jika berhentikan aplikasi maka aplikasi kembali ke proses awal tanpa harus mensetting ulang port.



Gambar 3.9 flowchart Client

Pembuatan flowchart pada perancangan laporan ini untuk mempermudah dalam proses pengerjaan untuk mencapai tujuan yang diinginkan. Flowchart diatas menerangkan tentang alur sistem aplikasi dimana server digunakan dalam handphone smartphone dan clien berada pada PC atau laptop.

Dibawah ini terdapat alur program keseluruhan untuk mengetahui proses pengiriman data accelerometer ke aplikasi monitoring pada PC atau laptop. dengan alur mulai menginisialisasikan variable sensor dan wifi, kemudian mensetting IP Adress dan Port, aplikasi ini telah dihubungkan antara smartphone dengan PC/Laptop. langkah selanjutnya proses pengambilan data, dan diolah data dari sensor kemudian pengiriman data sensor ke dalam aplikasi realtime pada PC/laptop. terlihat pada gambar 3.10. dibawah ini.



Gambar 3.10 flowchart sistem aplikasi

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

Tahap pengujian sistem merupakan proses menjalankan aplikasi yang telah dirancang yang dapat dijalankan. Tahapan ini merupakan lanjutan dari proses perancangan sesuai dengan spesifikasi dan desain sistem. Untuk mengetahui sistem yang dirancang sesuai dengan fungsi yang diharapkan, dilakukan pengujian terhadap sistem aplikasi tersebut baik secara keseluruhan atau subsistem. Selain itu pengujian juga dilakukan agar dapat menemukan beberapa permasalahan yang mungkin timbul pada saat alat ini beroperasi untuk kemudian diperbaiki sampai pada tingkat kesalahan sekecil mungkin sehingga didapatkan hasil yang baik. Berikut penjelasan mengenai prosedur Aplikasi.

4.1. Implementasi Sistem

Berikut ini adalah spesifikasi perangkat lunak sebagai sistem pendukung aplikasi yang dibuat dalam hal ini dibagi antara perangkat keras dan lunak:

- a. Spesifikasi pada hardware PC/Laptop :
 - Prosesor : Intel Core i3 2.40
 - Ram 2 Gb
 - Wifi onboard
- b. Spesifikasi pada hardware Android Smart Phone
 - CPU 1 GHz scorpion
 - Bobot – 136gram
 - Internal – 512 MB RAM
 - Jaringan Wifi
- c. Spesifikasi pada hardware pada helicopter
 - Main rotor diameter: 520mm
 - Weight: 305g, Length: 52cm, hight: 15.5cm, width: 8.5cm
 - Power system: 370 motor
 - Attitude: 100m flaying
 - Transmitter: standard 4CH
 - Mix controller: 4in1 controller (W/Gyro,mixer,ESC,receiver)
 - Digital Servo Digital: 8.0g,1.3kg CM,0.12S/60°

- Battery: 7.4V-850mAh (included)
 - Included: Simulator Cable + CD + Neckstrap + Training Gear
 - Time: about 8 minutes
- d. Spesifikasi pada hardware pada Wifi
- Higher speed :7.2M DL/ 5.76 UL,
 - Operating Sistem : Windows, MAC OS, 3G / 2G
 - Features: HSDPA : Plug & Play, Fully functional WiFi router with advanced routing functions,\ WPS Data
 - Type : WLAN 2.401 GHz- 2.495GHz,
- e. Spesifikasi Software
- Sistem Operasi Windows : Microsoft Windows 7
 - Java : versi JDK & JRE
 - Android : Versi 2.3.4 gingerbread
 - Aplikasi Netbeans 7
 - Aplikasi Eclipse 8

4.2. Pengujian aplikasi pada PC

Pengujian Jaringan Wifi dilakukan dengan mencoba komunikasi dari Wifi Android dan PC (Komputer pengujian) yang dihubungkan melalui access point sebagai Pengontol ip address yang diberikkan. sebelumnya untuk mengaktifkan di windows, caranya yaitu kita tinggal masuk ke komputer yang masih memakai OS Windows. pilih aplikasi java “ copter log client” dimana sebagai tampilan utama dalam aplikasi yang berada dalam PC (Komputer Pengujian).



Gambar 4.1 Tampilan Utama Aplikasi

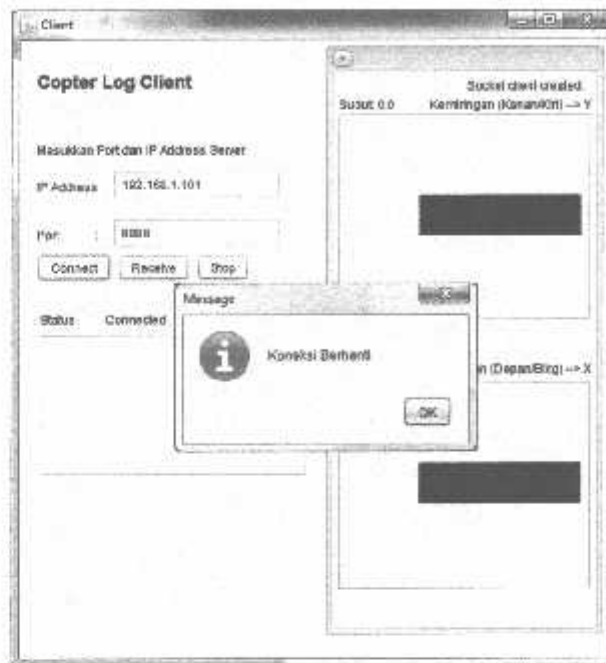
Cara menghubungkan aplikasi ke android tersebut dengan menggunakan mengisi IP address "192.168.1.101" dan alamat Port "8888" seperti gambar 4.2. dibawah ini.



Gambar 4.2 Proses koneksi

Setelah itu pilih tombol "connect" yang nantinya akan dihubungkan dengan aplikasi android yang tentunya dengan menggunakan perantara access point dalam jaringan wifi.

Untuk menghentikan aplikasi yang telah terhubung pilih tombol “stop” untuk mengakhiri hubungan ke server android seperti gambar 4.3. berikut.



Gambar 4.3 Koneksi berhenti

4.3 Pengujian aplikasi pada smartphone

Sebelumnya untuk menampilkan CLN Android handphone, yang telah dimasukkan pada Handphone android dengan nama aplikasi ChoptcrLogNew.

4.3.1. Pengujian Wifi

Pengujian wifi adalah pengujian perangkat keras access point untuk mengetahui apakah perangkat berfungsi dengan normal atau tidak. Perangkat yang di gunakan adalah access point Huawei e5. Skema pengujian adalah dengan menyalakan access point dan mencoba menghubungkan/mengkoneksikan beberapa perangkat client. Pengujian menunjukkan bahwa access point berfungsi dengan normal di mana setiap client terhubung mendapatkan IP secara otomatis seperti yang ditunjukkan dalam gambar 4.4.



Gambar 4.4. Access Point Huawei

4.3.2. Pengujian Koneksi ke server

- Tampilan Desktop pada Android sony Ericson mini



Gambar 4.5 Tampilan utama Android

- Cek aplikasi wifi pada android pada pengaturan android, untuk memastikan aplikasi tersambung terlihat pada gambar 4.6. berikut



Gambar 4.6. Pengaturan Wifi

4.3.3. Pengujian Setting Port

- Pengujian setting port adalah menguji apakah port yang telah di masukan dapat di set atau tidak. Pengujian dilakukan dengan memasukkan 4 digit angka lalu menekan tombol "set". Buka Aplikasi CLN pada android, tunggu hingga terditeksi aplikasi android dengan jaringan wifi seperti berikut.



IP terdeteksi.

Gambar 4.7. Searching ip adress

- Jika telah terdeteksi Ip address maka masukkan nilai port dengan "8888" kemudian klik tombol set untuk mengaktifkan aplikasi CLN pada android.



Port telah di-set..

Gambar 4.8. Proses setting port

4.3.4. Pengujian Run dan Stop koneksi ke server

Pengujian start server adalah menguji apakah server berjalan dengan normal atau tidak. Pengujian dilakukan dengan menekan tombol start. Setelah proses pensettingan ip address telah sukses aplikasi telah terkoneksi untuk pekatifan server pada android pilih tombol "Start" dan tunggu hingga terdeteksi atau muncul tampilan pada layar android "server running" seperti gambar 4.9. berikut.



Server running.

Gambar 4.9. Server Running

Hal ini menunjukkan server pada android telah berkerja dengan baik, selanjutnya untuk menghentikan server pilih tombol “stop”. Pengujian stop server adalah menguji server dapat dimatikan dengan normal. Pengujian di lakukan dengan menekan tombol “stop”, menghentikan proses pengiriman data pada server dan tunggu hingga muncul pemberitahuan “server off”.

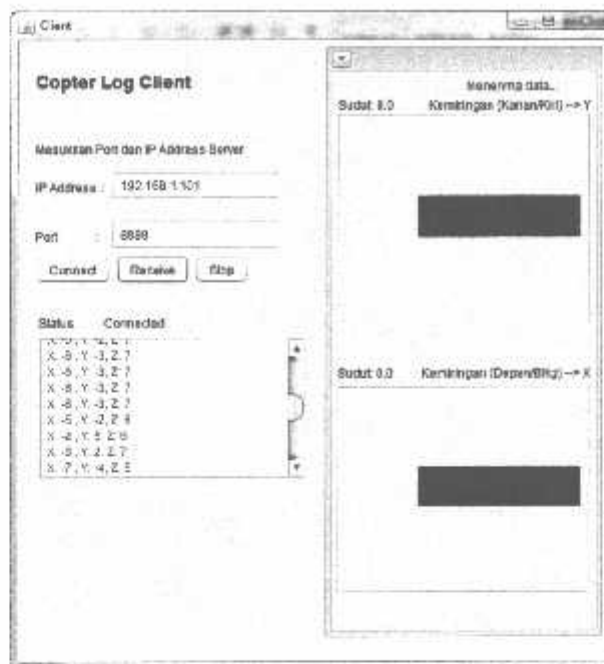


Server off..

Gambar 4.10. Server Off

4.3.5. Pengujian receive data

Dalam pengujian proses server baik PC maupun Android telah dihubungkan maka pengujian selanjutnya proses pengiriman data secara realtime dengan pilihan “receive” pada PC/Laptop dimana data yang diterima adalah nilai x, y dan z. yang dikirim dari aplikasi CLN pada android.



Gambar 4.11. Proses receive data

4.4 Pengujian Alat

Pengujian alat bertujuan untuk mengetahui kinerja system secara keseluruhan, apakah sudah sesuai dengan spesifikasi yang direncanakan.

1. Langkah pertama adalah pastikan aplikasi pada android telah hidup dan aplikasi pada PC/Laptop. aktifkan mode "Set" pada Android.
2. Di smartphone Android hidupkan Sinyal Wifi dan buka aplikasi CLN android, setelah itu CLN android akan mencari devices Wifi, jika sudah terdeteksi hubungkan Smartphone Android dengan Wifi dengan menekan tombol "Set" yang ada di aplikasi CLN android.
3. Langkah kedua Tekan Tombol "Receive" pada Aplikasi PC/Laptop.
4. Setelah Smartphone Android terhubung dengan PC/Laptop maka Helikopter RC bisa langsung dijalankan.

4.4.1 Pengujian arah gerakan accelometer

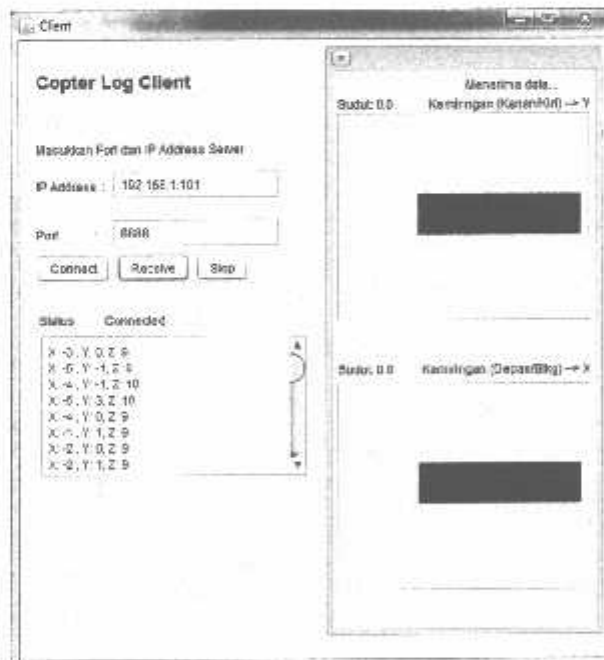
Untuk mengetahui arah gerakan Helikopter RC dengan cara melihat nilai x,y, dan z. Arah gerakan Accelometer di Tabel 4.1.

Tabel 4.1 Arah gerakan Accelometer

Nilai sensor Accelometer	Arah gerakan
X bernilai negatif (-)	Helikopter bergerak ke kanan
X bernilai positif (+)	Helikopter bergerak ke kiri
Y bernilai negatif (-)	Helikopter bergerak maju
Y bernilai positif (+)	Helikopter bergerak mundur
Z bernilai negatif (-)	Kesetabilan Helikopter
Z bernilai positif (+)	Kesetabilan Helikopter

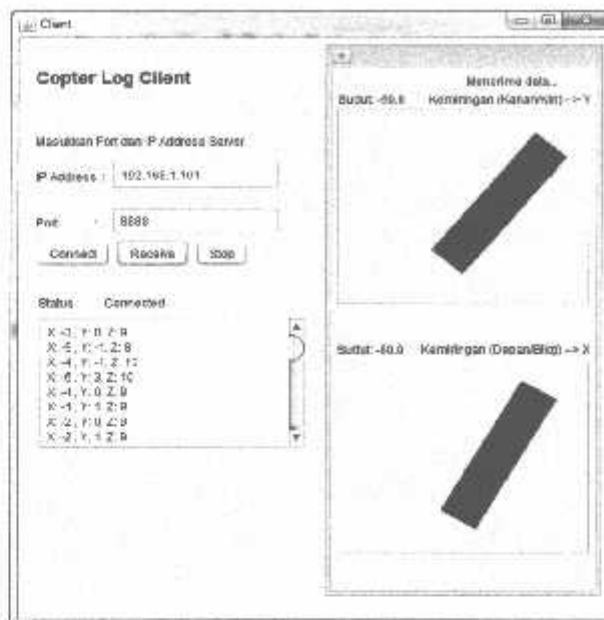
4.4.2 Pengujian simulasi pergerakan helikopter (animasi)

Proses akhir pengujian simulasi pergerakan dengan menggunakan animasi, dimana aplikasi helikopter dapat dimonitoring dengan aplikasi yang terdapat pada PC/laptop, proses pengujian ini terdapat beberapa bagian yang dapat diperhatikan seperti tampilan awal gambar 4.12. Tampilan animasi datar sebagai berikut.



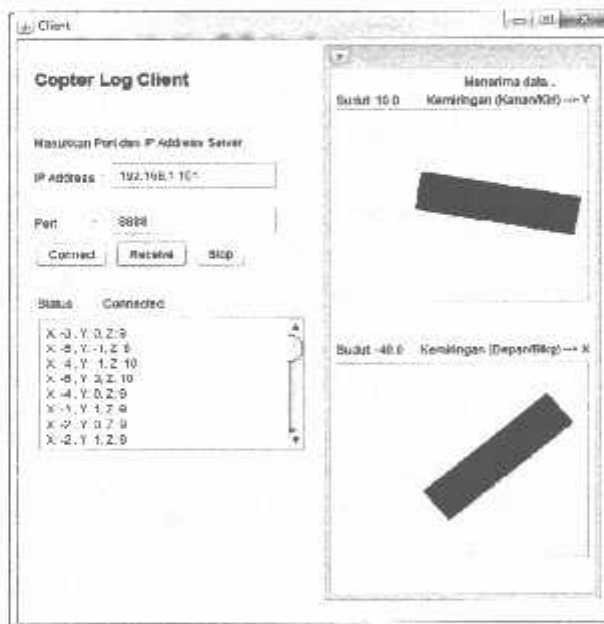
Gambar 4.12. Tampilan animasi datar

Dalam tampilan animasi ini helikopter ada pada posisi datar atau stabil pada saat di atas dengan tetap proses pengiriman attitude dilakukan secara realtime.



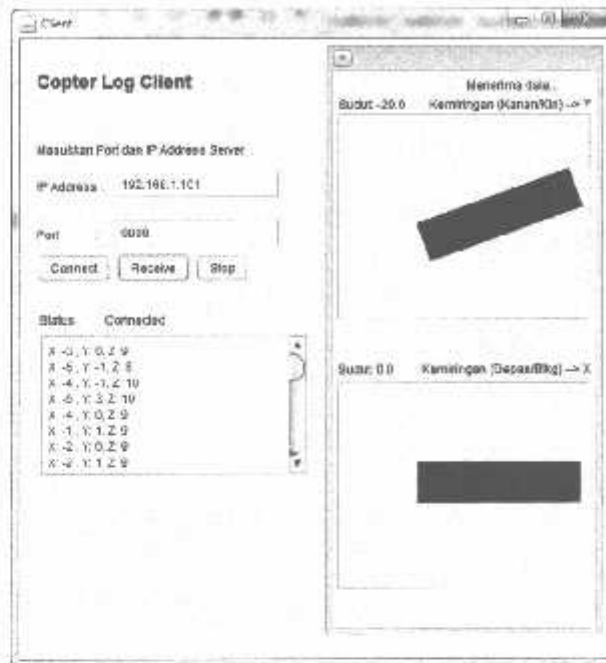
Gambar 4.13.Heli bergerak serong kiri

Tampilan selanjutnya helikopter digerakkan serong kiri. Selanjutnya helikopter bergerak serong kanan dengan pengirimana nilai x, y dan z mengikuti pergerakan attitude.



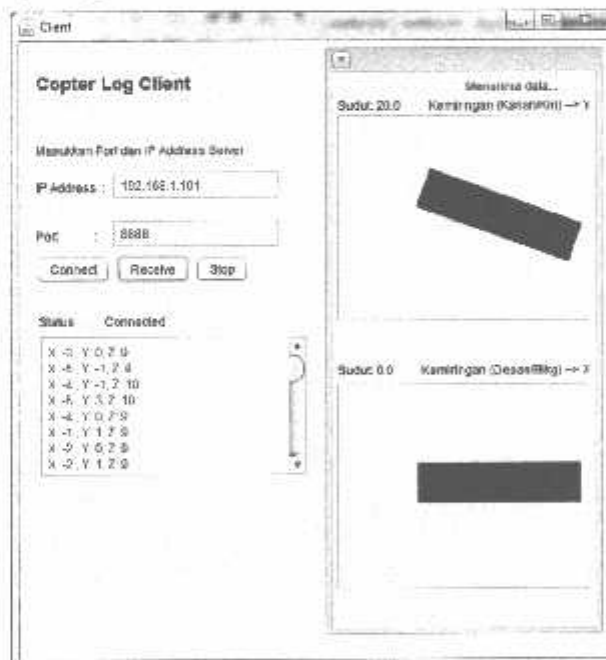
Gambar 4.14.Heli bergerak serong Kanan

Selanjutnya helikopter bergerak ke kiri dengan pergeseran nilai $y = -20$ $x = 0$ seperti gambar 4.15 berikut



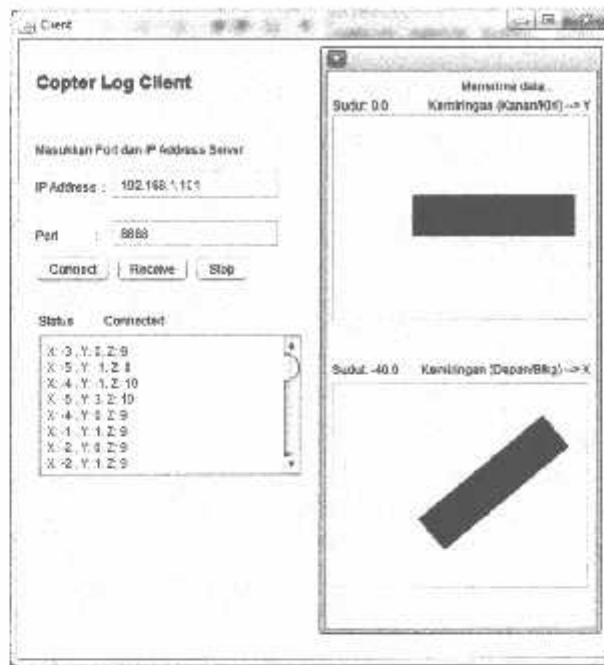
Gambar 4.15. Heli bergerak ke kiri

Helikopter dengan nilai $y=20$ dan nilai $x=0$, maka animasi akan bergerak ke kanan seperti terlihat pada gambar 4.16



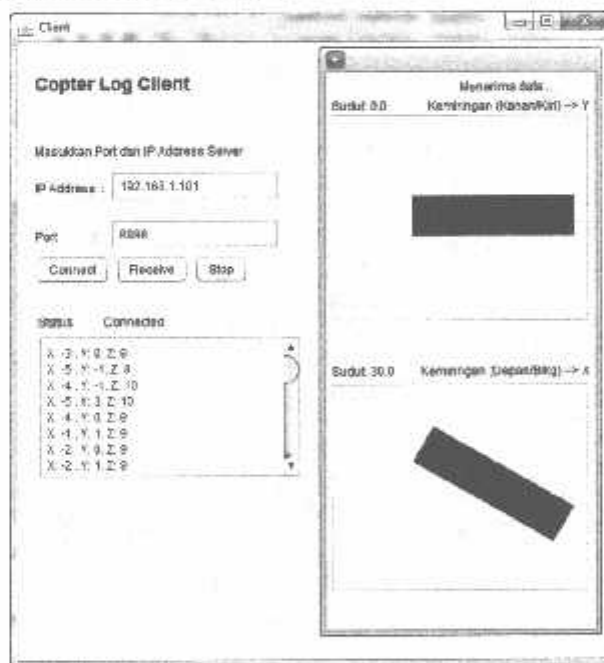
Gambar 4.16. Heli bergerak ke kanan

Helikopter dengan nilai $y=0$ dan nilai $x=-40$, maka animasi akan bergerak turun seperti terlihat pada gambar 4.17 berikut.



Gambar 4.17.Heli bergerak turun

Helikopter dengan nilai $y=0$ dan nilai $x= 40$, maka animasi akan bergerak naik seperti terlihat pada gambar 4.18 berikut.



Gambar 4.17.Heli bergerak naik

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan pengujian sistem dapat di simpulkan pengujian terhadap:

1. Sistem ini telah berhasil melakukan koneksi antara Smartphone Android dengan Laptop dengan media wifi.
2. Aplikasi android dapat mendeteksi x, y, dan z dan mengirimkan data tersebut melalui wifi.
3. Simulasi dapat menunjukkan pergerakan dengan penambahan animasi sesuai dengan keadaan helikopter (serong kanan-kiri, naik-turun).
4. Berdasarkan pengujian, jarak optimal agar server masih dapat mengirim data yakni kurang dari 12 meter.

5.2 Saran

Dari pengalaman yang diperoleh selama penyelesaian skripsi ini ada beberapa hal yang perlu diperhatikan dalam pengembangan alat ini dikemudian hari. Meskipun alat ini sudah dapat bekerja dengan baik dan sesuai dengan sistem yang direncanakan masih ada hal-hal yang perlu ditingkatkan baik dari sisi *hardware* maupun *software*-nya, adapun saran yang penulis berikan :

1. Untuk mengembangkan dan melengkapi metode monitoring sebaiknya digunakan pula camera yang sangat berguna dalam monitoring lebih jauh.
2. Sebagai pengembangan lebih lanjut aplikasi tersebut sebaiknya menanamkan sensor pada helikopter RC untuk pengoptimalan gerak dan penerimaan sensor yang lebih sempurna.
3. Animasi dapat di buat dalam bentuk 3 dimensi agar tampak lebih nyata.

DAFTAR PUSTAKA

- 1) Kimmo & Karvinen, Tero. *Make arduino bots and gadgets*, Penerbit O'REILLY, Canada 2011
- 2) Safaat, Nazaruddin H. *Pemrograman aplikasi mobile smartphone dan tablet Pc berbasis android*, Penerbit Informatika, Bandung 2012
- 3) A. Gow, Gordon & K.Smith Richard. *Mobile and Wireless Communication : An Introduction*. Poland 2006
- 4) Andi Publisher, Wahana Komputer, *PAS : Membangun Aplikasi Bisnis dengan Netbeans 7*. Indonesia 2012
- 5) Comic Reader Mobi (Jasakom). *Aplikasi Terbaik Handphone & Tablet Android*. Indonesia 2012

10/1/78

1

2

3

4

5

Lampiran 1 source kodekodesumberaplikasi desktop client

```
package com.pttif.thebeast;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.MediaTracker;
import java.awt.geom.AffineTransform;
import java.awt.image.BufferedImage;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.DataInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.Timer;
import java.util.TimerTask;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

public class Client_side extends javax.swing.JFrame {

    Socket cl_socket;
    String ip_addr, str_port;
    Integer ip_port;
    DataInputStream dis = null;
    DataOutputStream dos = null;
    //ObjectInputStream ois;
    //Thread cl_thread;

    private Image img;
    AffineTransform idn = new AffineTransform();
    Timer timer = new Timer();
    AnimationCanvas canvas, canvas2;
    String [] data_x_y_z;
    String msg = "";
    int x = 0, y = 0, z = 0;
    final int delay = 500;
    boolean receive = true;
    String dir = "up";
    public Client_side() {
        initComponents();
        canvas = new AnimationCanvas(1);
        canvas2 = new AnimationCanvas(2);
        panel.add(canvas);
        panel2.add(canvas2);
        canvas.start();
        canvas2.start();
    }

    @SuppressWarnings("unchecked")
```

```

// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    btn_connect = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    edit_ip = new javax.swing.JTextField();
    jScrollPane2 = new javax.swing.JScrollPane();
    edit_port = new javax.swing.JTextField();
    jInternalFrame1 = new javax.swing.JInternalFrame();
    txtData = new javax.swing.JLabel();
    panel = new javax.swing.JPanel();
    panel2 = new javax.swing.JPanel();
    lbl_sudut1 = new javax.swing.JLabel();
    lbl_sudut2 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    txt_status = new javax.swing.JLabel();
    btnReceive = new javax.swing.JButton();
    btnStop = new javax.swing.JButton();
    jScrollPane3 = new javax.swing.JScrollPane();
    txt_test = new javax.swing.JTextArea();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Client"); // NOI18N

    btn_connect.setText("Connect");
    btn_connect.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btn_connectActionPerformed(evt);
        }
    });

    jLabel1.setFont(new java.awt.Font("Lucida Grande", 1, 18));
    jLabel1.setText("Copter Log Client");

    jLabel2.setText("Masukkan Port dan IP Address Server");

    jLabel3.setText("IP Address :");

    jLabel4.setText("Port :");

    edit_ip.setText("192.168.1.100");
    jScrollPane1.setViewportView(edit_ip);

    edit_port.setText("8888");
    jScrollPane2.setViewportView(edit_port);

    jInternalFrame1.setVisible(true);

    txtData.setText("data ");

    panel.setBorder(javax.swing.BorderFactory.createEtchedBorder());

```



```

org.jdesktop.layout.GroupLayout panelLayout = new org.jdesktop.layout.GroupLayout(panel);
panel.setLayout(panelLayout);
panelLayout.setHorizontalGroup(
panelLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(0, 210, Short.MAX_VALUE)
);
panelLayout.setVerticalGroup(
panelLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(0, 202, Short.MAX_VALUE)
);

panel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

org.jdesktop.layout.GroupLayout panel2Layout = new org.jdesktop.layout.GroupLayout(panel2);
panel2.setLayout(panel2Layout);
panel2Layout.setHorizontalGroup(
panel2Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(0, 210, Short.MAX_VALUE)
);
panel2Layout.setVerticalGroup(
panel2Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(0, 203, Short.MAX_VALUE)
);

lbl_sudut1.setText("jLabel5");

lbl_sudut2.setText("jLabel5");

jLabel5.setText("Kemiringan (Kanan/Kiri --> Y");

jLabel7.setText("Kemiringan (Depan/Blkg --> X");

org.jdesktop.layout.GroupLayout jInternalFrame1Layout = new
org.jdesktop.layout.GroupLayout(jInternalFrame1.getContentPane());
jInternalFrame1.getContentPane().setLayout(jInternalFrame1Layout);
jInternalFrame1Layout.setHorizontalGroup(
jInternalFrame1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(jInternalFrame1Layout.createSequentialGroup()
.add(jInternalFrame1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(jInternalFrame1Layout.createSequentialGroup()
.add(jInternalFrame1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(133, 133, 133)
.add(txtData))
.add(jInternalFrame1Layout.createSequentialGroup()
.addContainerGap()
.add(lbl_sudut1)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 41,
Short.MAX_VALUE)
.add(jLabel5))
.add(jInternalFrame1Layout.createSequentialGroup()
.addContainerGap()
.add(panel1, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.add(jInternalFrame1Layout.createSequentialGroup()
.addContainerGap()
.add(jInternalFrame1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(jInternalFrame1Layout.createSequentialGroup()

```

```

        .add(lbl_sudut2)
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 35,
Short.MAX_VALUE)
        .add(jLabel7))
        .add(panel2, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
        .addContainerGap()
    );
jInternalFrame1Layout.setVerticalGroup(
jInternalFrame1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
    .add(jInternalFrame1Layout.createSequentialGroup()
        .addContainerGap()
        .add(txtData)
        .add(3, 3, 3)

        .add(jInternalFrame1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
            .add(lbl_sudut1)
            .add(jLabel5))
            .add(2, 2, 2)
            .add(panel, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 40,
Short.MAX_VALUE)

            .add(jInternalFrame1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
                .add(lbl_sudut2)
                .add(jLabel7))
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                .add(panel2, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .add(38, 38, 38))
            );
jLabel6.setText("Status");

txt_status.setText("Waiting Connection...");

btnReceive.setText("Receive");
btnReceive.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
btnRecciveActionPerformed(evt);
}
});

btnStop.setText("Stop");
btnStop.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
btnStopActionPerformed(evt);
}
});

txt_test.setColumns(20);
txt_test.setRows(5);
jScrollPane3.setViewportView(txt_test);

```

```

org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
    .add(layout.createSequentialGroup()
        .add(17, 17, 17)
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING, false)
                .add(org.jdesktop.layout.GroupLayout.LEADING, jLabel1,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .add(org.jdesktop.layout.GroupLayout.LEADING, jLabel2,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .add(layout.createSequentialGroup()
                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                    .add(jLabel3)
                    .add(jLabel4))
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                    .add(jScrollPane2, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 167,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                    .add(jScrollPane1, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 167,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))
                .add(layout.createSequentialGroup()
                    .add(btn_connect)
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                    .add(btnReceive)
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                    .add(btnStop))
                .add(layout.createSequentialGroup()
                    .add(3, 3, 3)
                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                        .add(jScrollPane3, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 273,
Short.MAX_VALUE)
                        .add(layout.createSequentialGroup()
                            .add(layout.createSequentialGroup()
                                .add(jLabel6)
                                .add(30, 30, 30)
                                .add(txt_status))))))
                .add(18, 18, 18)
                .add(jInternalFrame1)
                .add(ContainerGap())
            );
layout.setVerticalGroup(
layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
    .add(layout.createSequentialGroup()
        .add(layout.createSequentialGroup()
            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                .add(layout.createSequentialGroup()
                    .add(layout.createSequentialGroup()
                        .add(20, 20, 20)
                        .add(jLabel1, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 43,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .add(37, 37, 37)
                        .add(jLabel2)
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING,
false)
                            .add(layout.createSequentialGroup()
                                .add(layout.createSequentialGroup()
                                    .add(16, 16, 16)

```

```

        .add(jLabel3, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 19,
Short.MAX_VALUE))
        .add(layout.createSequentialGroup()
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)
        .add(jScrollPane1)))
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING,
false)
        .add(layout.createSequentialGroup()
        .add(22, 22, 22)
        .add(jLabel4, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 19,
Short.MAX_VALUE))
        .add(layout.createSequentialGroup()
        .add(18, 18, 18)
        .add(jScrollPane2)))
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
        .add(btn_connect)
        .add(btnReceive)
        .add(btnStop))
        .add(30, 30, 30)
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
        .add(jLabel6)
        .add(txt_status))
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
        .add(jScrollPane3, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 143,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
        .add(layout.createSequentialGroup()
        .add(ContainerGap()
        .add(jInternalFrame1, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))
        .add(ContainerGap(24, Short.MAX_VALUE))
    );

pack();
} // </editor-fold>

private void btn_connectActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ip_addr = edit_ip.getText().toString();
    str_port = edit_port.getText().toString();
    ip_port = Integer.parseInt(str_port);
    try {
        cl_socket = new Socket(ip_addr, ip_port);

        txtData.setText("Socket client created.");
        JOptionPane.showMessageDialog(null, "Koneksike Server Berhasil");
        txt_status.setText("Connected");
    } catch (UnknownHostException ex) {
        JOptionPane.showMessageDialog(null, "Koneksike Server Gagal !!!");
        txt_status.setText("Not Connected");
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(null, "Koneksike Server Gagal !!!");
        txt_status.setText("Not Connected");
    }
}

private void btnReceiveActionPerformed(java.awt.event.ActionEvent evt) {

```

```

// TODO add your handling code here:
try {
timer.scheduleAtFixedRate(new TimerTask() {
@Override
public void run() {
try {
if (!cl_socket.isClosed()) {
receive = true;
txtData.setText("Mencrima data...");

BufferedReader in = new BufferedReader(new InputStreamReader(cl_socket.getInputStream()));
msg = in.readLine();

if (msg != null) {
data_x_y_z = msg.split(",");

x = Integer.valueOf(data_x_y_z[0].replaceAll("X: ", "").trim());
y = Integer.valueOf(data_x_y_z[1].replaceAll("Y: ", "").trim());
z = Integer.valueOf(data_x_y_z[2].replaceAll("Z: ", "").trim());

txt_test.append(msg);
txt_test.append("\n");
}
}
if (!canvas.started) {
canvas.start();
}
} catch (IOException ex) {
Logger.getLogger(Client_side.class.getName()).log(Level.SEVERE, null, ex);
}
}, delay, delay);

} catch (Exception ex) {
System.out.println(ex.getMessage());
}
}

class AnimationCanvas extends JLabel implements Runnable {
Thread t;

Image img;

BufferedImage bi;

//double x, y;

double rotate_y = 0, rotate_x = 0;

double scale;

double anchorX, anchorY;

public boolean started = false;
boolean first = true;
int arah = 1; // 1 = miring kanankiri; 2 = depanbelakang/angguk2

```

```

AnimationCanvas(intarah) {
this.arah = arah;
setBackground(Color.GREEN);

setSize(320, 320);

img = getToolkit().getImage(getClass().getResource("img.jpg"));
MediaTrackermt = new MediaTracker(this);
mt.addImage(img, 1);

try {
mt.waitForAll();
} catch (Exception ex) {
System.out.println(ex.getMessage());
}

if (img.getWidth(this) == -1) {
System.out.println("Not valid.");
System.exit(0);
}

rotate_y = 0;
rotate_x = 0;
}

void step() {
// if (dir.equals("up")) {
// if (y == 9) {
// dir = "down";
// }
// else {
// y++;
// }
// }
// else if (dir.equals("down")) {
// if (y == -9) {
// dir = "up";
// }
// else {
// y--;
// }
// }
// }

if (arah == 1) {
rotate_y = y * 10;
}
if (arah == 2) {
rotate_x = x * 10;
}
}

@Override
public void paintComponent(Graphics g) {
super.paintComponent(g);
Dimension d = getSize();

bi = new BufferedImage(d.width, d.height, BufferedImage.TYPE_INT_ARGB);
Graphics2D big = bi.createGraphics();

```

```

step();

anchorX = 80;
anchorY = 20;

AffineTransform at = new AffineTransform();
at.setToIdentity();

    //if (first) {
at.translate(d.width/4, d.height/4);
    // first = false;
    //}

if (arah == 1) {
at.rotate(Math.toRadians(rotate_y), anchorX, anchorY);
    lbl_sudut1.setText("Sudut: " + rotate_y);
    }
else if (arah == 2) {
at.rotate(Math.toRadians(rotate_x), anchorX, anchorY);
    lbl_sudut2.setText("Sudut: " + rotate_x);
    }
big.drawImage(img, at, this);

    Graphics2D g2D = (Graphics2D) g;
g2D.drawImage(bi, 0, 0, null);

big.dispose();
}

public void start() {
    t = new Thread(this);
t.setPriority(Thread.MIN_PRIORITY);
t.start();
started = true;
}

public void stop() {
if (t != null) {
t.interrupt();
}

    t = null;
started = false;
}

@Override
public void run() {
    Thread me = Thread.currentThread();
while (t == me) {
if (receive) {
repaint();
}
//    repaint();
try {
t.sleep(delay);
} catch (Exception ex) {}
}
}

```

```

        t = null;
    }
}

private void btnStopActionPerformed(java.awt.event.ActionEvent evt) {
try {
    // TODO add your handling code here:
    receive = false;

    ip_port = 0;
    ip_addr = "";
    cl_socket.close();
    JOptionPane.showMessageDialog(null, "Koneksi Berhenti");
    txtData.setText("Socket telah ditutup.");
    timer.cancel();
    } catch (UnknownHostException ex) {
        //Logger.getLogger(Client_side.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IOException ex) {
        //Logger.getLogger(Client_side.class.getName()).log(Level.SEVERE, null, ex);
    }
}

@Override
public void paint(Graphics g) {
    Graphics2D g2d = (Graphics2D)g;
    AffineTransform trans = new AffineTransform();
    trans.setTransform(idn);
    trans.rotate(Math.toRadians(45));
    g2d.drawImage(img, trans, this);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Client_side.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Client_side.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Client_side.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

java.util.logging.Logger.getLogger(Client_side.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        @Override
        public void run() {
            new Client_side().setVisible(true);
        }
    });
    // Variables declaration - do not modify
    private javax.swing.JButton btnReceive;
    private javax.swing.JButton btnStop;
    private javax.swing.JButton btn_connect;
    private javax.swing.JTextField edit_ip;
    private javax.swing.JTextField edit_port;
    private javax.swing.JFrame jInternalFrame1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JLabel lbl_sudut1;
    private javax.swing.JLabel lbl_sudut2;
    private javax.swing.JPanel panel;
    private javax.swing.JPanel panel2;
    private javax.swing.JLabel txtData;
    private javax.swing.JLabel txt_status;
    private javax.swing.JTextArea txt_test;
    // End of variables declaration
}

/*
 *      //dos = new DataOutputStream(cl_socket.getOutputStream());
    //while (ip_port!=null){
        //dis = new DataInputStream(cl_socket.getInputStream());
        //RecMessageclThread = new RecMessage(dis, txt_test);
        //clThread.setDaemon(true);
        //clThread.setName("Andro");
        //clThread.start();
        //txt_test.append("\n"+dis.readUTF().toString());
    //}
 */

```

Handwritten text, possibly a signature or name, located in the upper middle section of the page.



```

packagecom.skripsi.copterlognew;

importjava.io.BufferedWriter;
importjava.io.IOException;
importjava.io.OutputStreamWriter;
importjava.io.PrintWriter;
importjava.net.ServerSocket;
importjava.net.Socket;

importandroid.app.Activity;
importandroid.content.Context;
importandroid.hardware.Sensor;
importandroid.hardware.SensorEvent;
importandroid.hardware.SensorEventListener;
importandroid.hardware.SensorManager;
importandroid.net.wifi.WifiInfo;
importandroid.net.wifi.WifiManager;
importandroid.os.Bundle;
importandroid.os.Handler;
importandroid.os.PowerManager;
importandroid.util.Log;
importandroid.view.Display;
importandroid.view.Menu;
importandroid.view.View;
importandroid.view.WindowManager;
importandroid.widget.Button;
importandroid.widget.EditText;
importandroid.widget.TextView;

public class MainActivity extends Activity implements SensorEventListener {

    // HW / Sensors
    floatmAccSensorX,mAccSensorY,mAccSensorZ,mOrientX,mOrientY,mOrientZ;
    floatmGySensorX,mGySensorY,mGySensorZ;
    double
    gravity_X,linear_acceleration_X,gravity_Y,linear_acceleration_Y,gravity_Z,linear_acceleration_Z
    ;
    intlinearIntX,linearIntY,linearIntZ;

    Display mDisplay;
    SensorManagermSensorManager;
    PowerManagermPowerManager;
    WindowManagermWindowManager;
    intmyIp;
    int port = 0;
    String IP = "";
    int i = 0;
    // GUI
    EditTexttxt_port;
    TextViewtxt_IP,txt_port_info,txt_X,txt_Y,txt_Z,txt_status;
    Button btn_set, btn_start, btn_stop;

    // realtime/thread
    Thread myCommThread = null;
    protected static final int MSG_ID = 0x1337;
    ServerSocketss = null;
    String mClientMsg = "";
    Thread myCommsThread = null;

```

```

PrintWriter data = null;
Socket socket = null;
boolean ready = true;
String data_accel = "";

Handler timerHandler;
Runnable sendToClient;

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

    initGUI();

    initSensor();

    checkIP();

    timerHandler = new Handler();
    sendToClient = new Runnable() {
        @Override
        public void run() {
            // TODO Auto-generated method stub
            sendData();
            timerHandler.postDelayed(this, 500);
        }
    };
}

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

@Override
protected void onResume() {
    super.onResume();
    timerHandler.postDelayed(sendToClient, 500);
}

@Override
protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(this,
    mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER));

    timerHandler.removeCallbacks(sendToClient);
}

@Override
protected void onStop() {
    super.onStop();
    try {
        // make sure you close the socket upon exiting
        if (ss != null) ss.close();
    } catch (IOException e) {

```

```

        e.printStackTrace();
    }
}

void initGUI() {
    txt_port = (EditText) findViewById(R.id.txt_port);
    txt_IP = (TextView) findViewById(R.id.txt_IP);
    txt_port_info = (TextView) findViewById(R.id.txt_port_info);
    txt_X = (TextView) findViewById(R.id.textView5);
    txt_Y = (TextView) findViewById(R.id.textView4);
    txt_Z = (TextView) findViewById(R.id.textView6);
    txt_status = (TextView) findViewById(R.id.txt_status);
    txt_status.setText("Aplikasi siap.");
}

void initSensor() {
    mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

    mSensorManager.registerListener(this,
        mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
        SensorManager.SENSOR_DELAY_NORMAL);
}

void checkIP() {
    try {
        WifiManager myWifiManager =
(WifiManager) getSystemService(Context.WIFI_SERVICE);
        WifiInfo myWifiInfo = myWifiManager.getConnectionInfo();
        myIp = myWifiInfo.getIpAddress();
        int intMyIp3 = myIp/0x1000000;
        int intMyIp3mod = myIp%0x1000000;
        int intMyIp2 = intMyIp3mod/0x10000;
        int intMyIp2mod = intMyIp3mod%0x10000;
        int intMyIp1 = intMyIp2mod/0x100;
        int intMyIp0 = intMyIp2mod%0x100;
        IP = intMyIp0 + "." + intMyIp1 + "." + intMyIp2 + "." + intMyIp3;
        txt_IP.setText(
            String.valueOf(intMyIp0)
            + "." + String.valueOf(intMyIp1)
            + "." + String.valueOf(intMyIp2)
            + "." + String.valueOf(intMyIp3)
        );
        Log.d("IP", IP);
        txt_status.setText("IP terdeteksi.");
    } catch (Exception ex) {
        Log.i("IP", ex.getMessage());
    }
}

void start_server() {
    myCommsThread = new Thread(new CommsThread());
    myCommsThread.start();
    Log.d("Server", "Server thread started.");
    txt_status.setText("Server running..");
}

@Override
public void onAccuracyChanged(Sensor arg0, int arg1) {
    // TODO Auto-generated method stub
}

@Override
public void onSensorChanged(SensorEvent event) {

```

```

// TODO Auto-generated method stub
try {
    mAccSensorX = event.values[0];
    gravity_X = ((0.8f * 9.8) + ((1 - 0.8f) * mAccSensorX)); //Low Pass Filter
    linear_acceleration_X = event.values[0] - gravity_X; //High Pass Filter
//    linearIntX = (int)(Math.round(linear_acceleration_X));
    linearIntX = (int)mAccSensorX;
    mAccSensorY = event.values[1];
    gravity_Y = ((0.8f * 9.8) - ((1 - 0.8f) * mAccSensorY));
//    linear_acceleration_Y = event.values[2] - gravity_Y;
    linearIntY = (int)(Math.round(linear_acceleration_Y));
    linearIntY = (int)mAccSensorY;
    mAccSensorZ = event.values[2];
    gravity_Z = ((0.8f * 9.8) + ((1 - 0.8f) * mAccSensorZ));
//    linear_acceleration_Z = event.values[2] - gravity_Z;
    linearIntZ = (int)(Math.round(linear_acceleration_Z));
    linearIntZ = (int)mAccSensorZ;
    txt_X.setText("X: " + String.valueOf(linearIntX));
    txt_Y.setText("Y: " + String.valueOf(linearIntY));
    txt_Z.setText("Z: " + String.valueOf(linearIntZ));
} catch (Exception ex) {
    Log.d("Sensor", ex.getMessage());
}
}

public void btn_OnClick(View v) {
    switch(v.getId()) {
        case R.id.btn_set:
            if (!txt_port.getText().equals("")) {
                port = Integer.valueOf(txt_port.getText().toString());
                Log.d("PORT", String.valueOf(port));
                txt_port_info.setText(String.valueOf(port));
                txt_status.setText("Port telah di-set.");
            }
            break;
        case R.id.btn_start:
            if (!IP.equals("") && port != 0) {
                Log.d("Server", "Starting...");
                start_server();
            }
            break;
        case R.id.btn_stop:
            try {
                //socket.close();
                //ss.close();
                socket = null;
                ss = null;
                myCommsThread.stop();
                myCommsThread = null;
                txt_status.setText("Server off..");
            } catch (Exception e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            break;
        default:
            break;
    }
}
}

```

```

void sendData() {
    //Log.d("Send", "data");
    try {
        if (ss != null && socket != null) {
            data_accel = "X: " + linearIntX + ", Y: " + linearIntY + ", Z: "
+ linearIntZ + "\r\n";
            data.println(data_accel);
            data.flush();
            Log.d("Server", "Data sent: " + data_accel);
        }
    } catch (Exception ex) {
        Log.d("Send Data", ex.getMessage());
    }
}

class CommsThread implements Runnable {
public void run() {
    try {
        // buat server
        if (port != 0) {
            ss = new ServerSocket(port);
            Log.d("Server", "Server Socket created.");

            socket = ss.accept();
            Log.d("Socket", "Socket Created.");

            data = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream()), true);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

**FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK**

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417635 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : **Bobby Ananda Martiyan**
Nim : **09.12.526**
Jurusan : **Teknik Elektro**
Konsentrasi : **Teknik Komputer S-1**
Masa Bimbingan : **Semester Genap 2012-2013**
Judul : **SISTEM MONITORING ATTITUDE HELIKOPTER
RC MENGGUNAKAN ANDROID SMARTPHONE
MELALUI WIFI**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)

Pada Hari : **Jumat**
Tanggal : **18 Agustus 2013**
Dengan Nilai : **76,25 (B+) *7***

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

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

Sekretaris Majelis Penguji

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

ANGGOTA PENGUJI

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP.P.103000365

Dosen Penguji II

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



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik komputer, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : **Bobby Ananda Martiyan**
Nim : **09.12.526**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik komputer**
Masa Bimbingan : **Semester Genap 2012-2013**
Judul Skripsi : **SISTEM MONITORING ATTITUDE HELIKOPTER RC
MENGUNAKAN ANDROID SMARTPHONE
MELALUI WIFI**

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	18 Agustus 2013	<ul style="list-style-type: none">• Proses pembacaan accelometernya buat flowchart bab 3• Tampilan di ganti helikopter di buat bisa berputar 3 dimensi	
2.	Penguji II			

Disetujui:

Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP.P.103000365

Penguji II

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

Mengetahui:

Dosen Pembimbing I

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

Dosen Pembimbing II

Yuli Wahyuni, ST, MT
NIP.P.1031200456



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Bobby.
NIM : 0912526
Perbaikan meliputi :

- Proses pembacaan. Accelerometer nya buat flowchart. → BAB 3.
- tampilan diganti helikopter →
Abrat : bisa berputar 3 dimensi.

Malang,

(_____)



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

1.	Nim	: 0912526		
2.	Nama	: BOBBY ANANDA MARTIYAN		
3.	Konsentrasi Jurusan	: Teknik Komputer		
4.	Jadwal Pelaksanaan:	Waktu	Tempat	
		29 April 2013	09:00	III.1.3
5.	Judul proposal yang diseminarkan Mahasiswa	SISTEM MONITORING ATTITUDE HELIKOPTER RC MENGGUNAKAN ANDROID SMARTPHONE MELALUI WIFI		
6.	Perubahan judul yang diusulkan oleh Kelompok Dosen Keahlian			
7.	Catatan:	Tujuan harus mengeru, kemiringanya seperti apa, jarak antara pesawat dengan laptop, berat pesawat terhadap android		
	Catatan:			
	Persetujuan judul Skripsi			
8.	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
	(.....)	 (Michael Ardita.....)	 (.....)	
	Mengetahui, Ketua Program Studi Teknik Elektro S-1 M. Ibrahim Ashari, ST, MT NIP. P 1030100358	Disetujui, Calon Dosen Pembimbing ybs		
		Pembimbing I (.....)	Pembimbing II (Yuu wahyuni)	



ampiran : 1 (satu) berkas
Pembimbing Skripsi

kepada : Yth. Bapak/Ibu **Dr. Eng. Aryuanto Soetedjo, ST, MT**
 Dosen Teknik Elektro S-1
 ITN Malang

Yang bertanda tangan dibawah

Nama : **BOBBY ANANDA MARTIYAN**
Nim : **0912526**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing untuk penyusunan Skripsi dengan judul :

"SISTEM MONITORING ATTITUDE HELIKOPTER RC MENGGUNAKAN ANDROID SMARTPHONE MELALUI WIFI"

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Ketua Program Studi Teknik Elektro S-1

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

Hormat Kami

BOBBY ANANDA MARTIYAN
NIM. 0912526



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

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

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **BOBBY ANANDA MARTIYAN**

Nim : **0912526**

Semester : **VIII (Delapan)**

Jurusan : **Teknik Elektro S-1**


Konsentrasi : **Teknik Komputer**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" SISTEM MONITORING ATTITUDE HELIKOPTER RC MENGGUNAKAN ANDROID SMARTPHONE MELALUI WIFI"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami


Dr. Eng. Aryuanto Soetedjo, ST, MT

NIP. 1030800417

*) Coret yang tidak perlu



FORMULIR BIMBINGAN SKRIPSI

Nama : Bobby Ananda Martiyan
Nim : 09.12.526
Masa Bimbingan : Semester Genap 2012-2013
Judul Skripsi : SISTEM MONITORING ATTITUDE HELIKOPTER RC
MENGGUNAKAN ANDROID SMART PHONE MELALUI WIFI

No	Tanggal	Uraian	Paraf Pembimbing
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing I

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

Form S-4b



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-137/EL-FTI/2013

1 Mei 2013

Lampiran : -

Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Bapak/Ibu **Yuli Wahyuni, ST, MT**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **BOBBY ANANDA MARTIYAN**
Nim : **0912526**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Genap Tahun Akademik 2012 - 2013 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358



PERMOHONAN PERSETUJUAN SKRIPSI

Yang Bertanda Tangan Dibawah Ini:

Nama : BOBBY ANANDA MARTIYAN
 NIM : 0912526
 Semester : III
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-I
 Konsentrasi : TEKNIK ENERGI LISTRIK
TEKNIK ELEKTRONIKA
TEKNIK KOMPUTER DAN INFORMATIKA
TEKNIK KOMPUTER
TEKNIK TELEKOMUNIKASI
 Alamat : Jl. Mastrip. 58. LAMONGAN

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat SKRIPSI Tingkat Sarjana. Untuk melengkapi permohonan tersenut, bersama ini kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan- persyaratan pengambilan SKRIPSI adalah sebagai berikut:

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan laporan Praktek Kerja *ada nilai* (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh matakuliah > 134 sks dengan IPK > 2 dan tidak ada nilai *E* (.....)
5. Telah mengikuti secara aktif kegiatan seminar Skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenarannya data tersebut diatas
 Recording Teknik Elektro S-I

[Signature]
 (.....)

Malang, 9 April 2013
 Pemohon

[Signature]
 (.....)

Ket: