

MENGENDALIKAN KAMERA IP DENGAN MENGGUNAKAN *HANDPHONE* BERBASIS ANDROID

Lucas de Jesus Silva¹⁾, Andrew Joewono²⁾, AntoniusWibowo²⁾
Email: kodachi.boys@yahoo.com

ABSTRAK

Android merupakan sebuah sistem operasi yang tengah berkembang saat ini. Dengan adanya teknologi ini setiap orang dapat membuat aplikasi sendiri pada ponsel berbasis Android. Beranjak dari hal tersebut, kini dapat dibuat suatu aplikasi yang dapat menggerakkan beberapa kamera IP yang tidak dilengkapi dengan fasilitas penggerak, yakni kamera IP D-Link DCS-910. Alat ini digunakan sebagai penggerak untuk memantau keadaan suatu area melalui perangkat mobile pada jarak yang jauh.

Alat ini terdiri dari: 2 buah motor servo, modul Wiznet110SR, sistem minimum mikrokontroler ATmega8, wireless router, kamera IP D-Link DCS-910, dan pengendali handphone Android. Handphone digunakan sebagai tempat untuk mengendalikan arah pergerakan dari kamera IP, dan memberikan perintah pengendalian. Untuk interface antara handphone dan kamera IP menggunakan Handphone berbasis Android. Kamera IP akan digerakkan oleh 2 buah motor servo. Perintah untuk menggerakkan motor servo dikirim secara serial ke mikrokontroler. Kamera IP yang terpasang pada motor servo akan bergerak mengikuti perintah yang dikirim oleh handphone Android. Hasil gambar yang diambil oleh kamera IP tersebut akan dikirim melalui wifi, dan ditampilkan pada handphone Android.

Dari hasil pengukuran, dan pengujian, dapat disimpulkan bahwa handphone Android sudah dapat mengontrol arah gerak dari motor servo, dan juga dapat menampilkan gambar dari kamera IP.

Kata kunci : pengendalian, kamera IP, *handphone*, Android, Wifi,

PENDAHULUAN

Teknologi *mobile programming* terus berkembang, dan saat ini menjadi semakin lengkap sejak adanya *handphone* (HP) berbasis Android. Dengan adanya teknologi ini setiap orang dapat membuat aplikasi sendiri pada ponsel berbasis Android. Konektivitas merupakan salah satu aspek yang mendukung perangkat untuk berinteraksi dengan perangkat-perangkat lain: seperti kamera *Internet Protocol* (IP) atau kamera digital berbasis IP yang merupakan salah satu perangkat yang dapat dikendalikan dari jarak jauh melalui perangkat *mobile*.

Ada beberapa kamera IP yang tidak dilengkapi dengan penggerak, sehingga jangkauan yang ditampilkan oleh kamera IP sangat terbatas. Dengan dibuatnya alat ini, dapat mengendalikan kamera IP untuk memantau keadaan suatu area melalui perangkat *mobile* pada jarak yang jauh.

Bahasa pemrograman yang digunakan adalah bahasa pemrograman *Java*, dan *Extensible Markup Language* (XML) yang akan mengendalikan kamera IP melalui *Wireless Fidelity* (Wifi). Selain itu, alat ini dapat digunakan untuk memantau keadaan suatu tempat yang sudah terpasang kamera IP dari jarak jauh.

Masalah yang muncul dalam pengerjaan alat adalah:

1. Membuat tampilan dari kamera IP pada *handphone* berbasis Android dengan menggunakan bahasa pemrograman *Java*, dan *XML*;
2. Mengendalikan kamera IP menggunakan motor servo;
3. Membuat rancangan mekanik agar kamera IP dapat bergerak mengikuti arah gerak dari motor servo.

Agar sistem ini lebih spesifik dan terarah, maka pembahasan masalah dalam program ini memiliki batasan-batasan sebagai berikut:

1. Menggunakan kamera IP *D-Link DCS-910* untuk memantau suatu lokasi;
2. Komunikasi antara *handphone* dengan mikrokontroler menggunakan *wireless router*;
3. Sistem operasi pada *handphone* menggunakan Android dan terdapat fasilitas *Wifi*;
4. Jangkauan yang bisa diakses melalui *wifi*: adalah *indoor* (100 meter), dan *outdoor* (300 meter).

Tujuan yang hendak dicapai dalam pembuatan alat ini ialah merancang dan membuat suatu sistem untuk memantau keadaan suatu tempat menggunakan kamera IP melalui *handphone* berbasis Android pada jarak jauh serta mengembangkan aplikasi Android untuk *handphone*.

¹⁾ Mahasiswa di Fakultas Teknik Jurusan Teknik Elektro Universitas Katolik Widya Mandala Surabaya

²⁾ Staf Pengajar di Fakultas Teknik Jurusan Teknik Elektro Universitas Katolik Widya Mandala Surabaya

TINJAUAN PUSTAKA

Handphone Android^[1]

Handphone yang digunakan adalah *handphone* yang berbasis Android. Android adalah sistem operasi (*OS*) untuk *handphone* yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi sendiri yang akan digunakan oleh bermacam peranti bergerak. Gambar 1 merupakan gambar dari *handphone* berbasis Android.



Gambar 1. Handphone berbasis Android

Android adalah sistem operasi yang digunakan pada *smartphone* dan juga *tablet Personal Computer (PC)*. Fungsinya sama seperti sistem operasi *Symbian* di Nokia, *iOS* di Apple, dan *BlackBerry*. *OS* di dalam Android tidak terikat ke satu merek *handphone* saja, beberapa vendor terkenal yang sudah memakai Android antara lain: Samsung, Sony Ericsson, HTC, Nexus, Motorola, dan lain-lain.

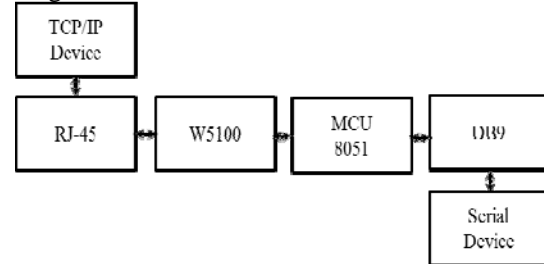
Android pertama kali dikembangkan oleh perusahaan bernama Android Inc, dan pada tahun 2005 diakuisisi oleh raksasa *Internet Google*. Android dibuat dengan basis Linux yang telah dimodifikasi, dan untuk setiap *release* diberi kode nama berdasarkan nama hidangan makanan. Keunggulan utama Android adalah gratis dan *open source*, sehingga membuat *smartphone* Android dijual lebih murah dibandingkan dengan *Blackberry* atau *iPhone* meski fitur (*hardware*) yang ditawarkan Android lebih baik. Beberapa fitur utama dari Android antara lain: *WiFi hotspot*, *Multi-touch*, *Multitasking*, *GPS*, *accelerometers*, *support java*, mendukung banyak jaringan (*GSM/EDGE*, *IDEN*, *CDMA*, *EV-DO*, *UMTS*, *Bluetooth*, *WiFi*, *LTE* dan *WiMAX*) serta juga kemampuan dasar *handphone* pada umumnya. Jenis-jenis *OS* Android di dalam aplikasi *mobile* yaitu:

- Android versi 1.1 dirilis pada 9 Maret 2009;
- Android versi 1.5 (*Cupcake*) dirilis pada pertengahan Mei 2009;
- Android versi 1.6 (*Donut*) dirilis pada September 2009;

- Android versi 2.0/2.1 (*Eclair*) dirilis pada 3 Desember 2009;
- Android versi 2.2 (*Froyo: Frozen Yoghurt*) diluncurkan pada 20 Mei 2010;
- Android versi 2.3 (*Gingerbread*) dirilis pada 6 Desember 2010;
- Android versi 3.0/3.1 (*Honeycomb*) dirilis pada Mei 2011;
- Android versi 4.0 (*Ice Cream*) dirilis pada akhir 2011.

Modul TCP/IP WIZ110SR^[2]

WIZ110SR merupakan sebuah modul yang mampu mengubah *TCP/IP* menjadi protokol serial *RS-232*. Aplikasi umum dari modul ini adalah mengoperasikan suatu perangkat yang memiliki *RS-232* melalui jaringan *ethernet*. Gambar 2 merupakan diagram blok modul *WIZ110SR*.



Gambar 2. Diagram blok modul *WIZ110SR*

Penjelasan Gambar 2:

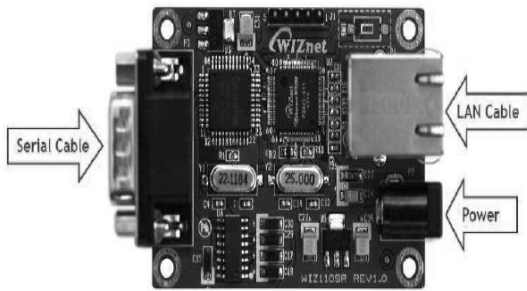
- *RJ-45*, dan transformer
RJ-45 merupakan konektor standar pada perangkat *ethernet*, dan memiliki 8 konektor. Konektor yang terdapat pada modul *WIZ110SR* adalah *female connector*.
- *W5100*
Ketika data diterima dari konektor *RJ-45*, *chip W5100* akan menyimpan data tersebut di *buffer*, mengolah sesuai protokol, dan mengirimkan data dalam bentuk serial pada *MCU 8051*. Ketika data serial diterima dari *MCU 8051*, *chip W5100* akan mengolah data tersebut sesuai dengan protokol dan mengirimkannya ke konektor *RJ-45*.
- *MCU 8051*
MCU 8051 mengontrol data dari *chip W5100* maupun dari konektor serial sesuai dengan konfigurasi *user*.
- *DB9*
DB9 adalah konektor standar pada perangkat serial *RS-232*. Konektor pada modul *WIZ110SR* adalah *male connector*.

Keunggulan modul ini antara lain:

- Mendukung koneksi *PPPoE*;

- Konfigurasi modul melalui *software via OS Windows*;
- Koneksi langsung dengan perangkat serial dengan konfigurasi serial yang mudah;
- Memiliki fitur keamanan dengan *password*.
- Antar muka *Ethernet 10/100 Mbps* dengan kecepatan serial sampai dengan *230 kbps*.

Gambar 3 merupakan gambar dari Modul WIZ110SR.



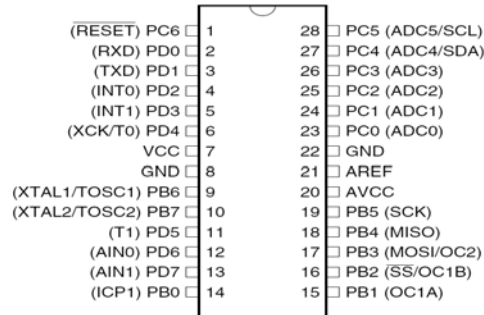
Gambar 3. Modul WIZ110SR

Mikrokontroler ATmega8^[3]

Mikrokontroler *ATmega 8* merupakan salah satu jenis mikrokontroler *AVR* yang diproduksi oleh *Atmel*. Hal yang membedakan mikrokontroler *AVR* yang satu dengan yang lainnya antara lain adalah kapasitas memori program seperti: *EEPROM*, dan *SRAM*, serta kemampuan khusus setiap mikrokontroler. *Atmega8* adalah sebuah mikrokontroler 8-bit *AVR* dengan performa kinerja yang tinggi dan rendah dalam penggunaan daya. *ATmega8* mempunyai kapasitas 8 kilobit memori program *ISP*, 512 bit *EEPROM*, dan 1 kilobit *internal SRAM*. *ATmega8* dilengkapi dengan 23 buah pin *I/O* yang terbagi menjadi 3 port yaitu port B, C, dan D. Selain itu, *ATmega8* juga mempunyai keistimewaan antara lain yaitu mempunyai: 2 buah *timer/counter* 8-bit, 1 buah *timer/counter* 16-bit, *Real Time Counter (RTC)* dengan osilator yang terpisah, 3 buah kanal *PWM*, 6 buah kanal *ADC* dengan ketelitian 10-bit, *Two-Wire Serial Interface (TWI)*, *Universal synchronous and Asynchronous serial Receiver and Transmitter (USART)*, *Serial Peripheral Interface (SPI)*, *Watchdog Timer*, *Analog Comparator*, dan *sleep mode*.

ATmega8 membutuhkan sumber tegangan antara 2,7-5,5 V untuk menjalankan fungsinya. Kristal yang dapat digunakan pada *ATmega8* antara 0-8 MHz. Gambar 4 menunjukkan konfigurasi pin *ATmega8*. Pada Gambar tersebut terlihat mikrokontroler *ATmega8* mempunyai 28 pin yang terdiri atas beberapa port. Port tersebut yang akan

digunakan dalam perancangan sistem minimum mikrokontroler *ATmega8* pada pengerjaan alat ini dengan menggunakan *handphone* berbasis *Android*.



Gambar 4. Konfigurasi pin *IC ATmega8*

TCP/IP^[4]

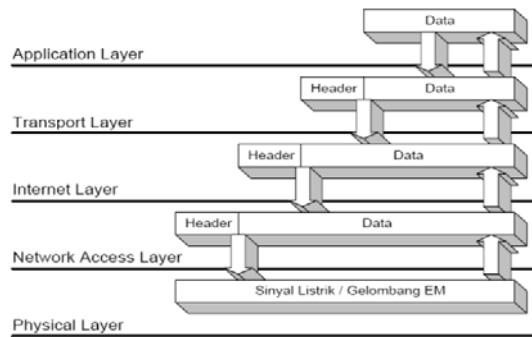
TCP/IP merupakan suatu protokol *suite* yang digunakan untuk mengirim data antar komputer di dalam jaringan tanpa batasan perangkat keras maupun perangkat lunak. Protokol ini juga bisa digunakan untuk pengiriman data, informasi maupun kendali di dalam jaringan komputer. *TCP/IP* terdiri atas beberapa *layer* (lapisan), yang mana tiap lapisan secara spesifik mengerjakan apa yang menjadi tugasnya tanpa terkait dengan tugas lapisan yang lain. Oleh *Internasional Standart Organisation (ISO)*, telah ditetapkan suatu konsep standar untuk arsitektur jaringan komputer yang dikenal dengan nama *OSI Open System Interconnection (OSI)*. *OSI* terdiri atas 7 lapisan, sementara *TCP/IP* hanya terdiri atas 5 lapisan. Namun, seluruh fungsi dari lapisan *OSI* telah tercakup dalam *TCP/IP*. Tabel 2 merupakan tabel perbandingan *layer OSI* terhadap *TCP/IP*.

Tabel 2. Perbandingan *layer OSI* terhadap *TCP/IP*

<i>Application</i>	<i>Application</i>
<i>Presentation</i>	
<i>Session</i>	
<i>Transport</i>	<i>Transport</i>
<i>Network</i>	<i>Intrnet</i>
<i>Data link</i>	<i>Network</i>
<i>Phisycal (hardware)</i>	<i>Phisycal (hardware)</i>
<i>OSI model</i>	<i>TCP/IP model</i>

Lapisan-lapisan dalam *TCP/IP* menggambarkan fungsi-fungsi, dan protokol dalam komunikasi antar 2 buah komputer. Setiap lapisan menerima data dari lapisan atas atau bawah, kemudian memproses data tersebut sesuai fungsinya, dan meneruskannya ke lapisan berikutnya. Untuk tiap data, tiap lapisan memiliki *header* sendiri sesuai dengan fungsinya. Pemberian *header* pada masing-masing lapisan disebut sebagai proses

enkapsulasi. Gambar 5 merupakan proses komunikasi data antar *layer TCP/IP*.



Gambar 5. Proses komunikasi data antar *layer TCP/IP*

Fungsi tiap lapisan pada *TCP/IP* dapat dijelaskan sebagai berikut:

1. *Physical (Hardware)*

Lapisan ini merupakan lapisan terbawah dalam *TCP/IP*. Lapisan ini mendefinisikan besaran fisik suatu media komunikasi seperti: tegangan, arus, maupun gelombang radio;

2. *Network*

Lapisan ini bertugas mengatur penyaluran data yang akan dikirimkan pada media fisik. Lapisan ini juga menjaga reliabilitas data yang akan dikirimkan. Pada lapisan ini, dilakukan deteksi dan koreksi kesalahan pada tingkat bit;

3. *Internet*

Lapisan ini mendefinisikan bagaimana dua pihak dapat berhubungan walaupun berada pada jaringan yang berbeda. Pada jaringan internet yang terdiri atas puluhan juta *host* dan ratusan ribu jaringan lokal, lapisan ini bertugas untuk menjamin agar suatu paket yang dikirimkan dapat menemukan tujuannya. Beberapa tugas penting pada lapisan ini adalah:

- *Addressing*, yaitu melengkapi tiap data dari alamat internet yang dituju yang dikenal dengan nama *Internet Protocol Address*;
- *Routing*, yaitu menentukan ke mana data dikirim mencapai tujuannya. Fungsi ini merupakan fungsi terpenting dalam *Internet Protocol*.

4. *Transport*

Lapisan ini menjamin reliabilitas data dikirim antara pengirim dan penerima pada tingkat segmen. Lapisan ini menjamin bahwa data yang diterima pada penerima adalah sama dengan data yang dikirimkan oleh pengirim.

Pada lapisan ini terdapat 2 protokol yang sering digunakan, yaitu:

- *User Datagram Protocol (UDP)*, protokol ini tidak menyediakan layanan pemeriksaan data, dan bersifat *connectionless*;
- *Transport Control Protocol (TCP)*, protokol ini menyediakan layanan pemeriksaan data dan bersifat *connection oriented*. *TCP* digunakan untuk pengiriman data yang menuntut keandalan. Protokol ini memiliki beberapa fungsi penting, yaitu:
 - *Flow control*, pengiriman data yang telah dipecah menjadi paket-paket tersebut harus diatur sedemikian rupa agar pengiriman tidak sampai mengirimkan data dengan kecepatan yang melebihi kemampuan penerima dalam menerima data;
 - *Error detection*, data dilengkapi dengan informasi yang dapat digunakan untuk memeriksa apakah data yang dikirimkan bebas dari kesalahan.

5. *Application*

Lapisan ini merupakan lapisan terakhir dalam arsitektur *TCP/IP* yang berfungsi mendefinisikan aplikasi yang dijalankan pada jaringan. Oleh karena itu terdapat banyak protokol pada lapisan ini sesuai dengan banyaknya aplikasi yang dapat dijalankan.

Standar Protokol Wifi

Dalam dunia *wireless network*, dikenal beberapa standar protokol radio yang biasa digunakan. Biasa dikenal dengan protokol radio 802.11 atau disebut dengan *wi-fi*. Protokol tersebut memungkinkan antar perangkat *wireless* saling berkomunikasi tanpa tergantung oleh merek perangkat tersebut. Protokol 802.11 sendiri memiliki beberapa sub-varian. Namun tidak semua dari sub-varian tersebut banyak dipakai oleh perangkat yang dijual di pasaran. Subvarian dari protokol radio 802.11 yang banyak dijumpai di pasaran adalah sebagai berikut:

- 802.11b

Disahkan oleh *IEEE* pada September 16, 1999. 802.11 adalah protokol radio yang digunakan oleh perangkat *wi-fi* semenjak dikomersialkan pada tahun 1999. Protokol ini beroperasi pada frekuensi 2,4-2,495 GHz, dan menggunakan metode modulasi *Direct Sequence Spread Spectrum (DSSS)*. Kecepatan maksimumnya adalah sebesar 11

Mbps dengan porsi data aktual sebesar 5 Mbps.

- 802.11g
Disahkan pada tahun 2003 dan terlambat dalam pengerjaannya. Terlepas dari keterlambatannya, 802.11g sekarang menjadi protokol *wi-fi* yang paling banyak digunakan oleh perangkat *wi-fi* pada saat ini. Protokol ini beroperasi pada frekuensi yang sama dengan 802.11b, tetapi menggunakan metode modulasi *Orthogonal Frequency Division Multiplexing (OFDM)*. Kecepatan maksimumnya adalah sebesar 54 Mbps dengan porsi data aktual sebesar 22 Mbps.
- 802.11a
Disahkan bersamaan dengan 802.11b. 802.11a beroperasi pada frekuensi 5.745 Ghz, dan 5.805 Ghz sehingga berakibat besarnya daya yang digunakan untuk memancarkan data pada jarak yang sama dengan 2.4 Ghz. Kecepatan maksimumnya sebesar 54 Mbps dengan porsi data aktual sebesar 27 Mbps. Tidak semua negara melegalkan penggunaan frekuensi 802.11a, sehingga 802.11a kurang populer di antara pengguna *wi-fi*.
- 802.11n
Merupakan pengembangan lebih lanjut dari 802.11g. Menggunakan metode modulasi yang sama dengan 802.11g. 802.11n mampu mencapai kecepatan maksimum sebesar 600 Mbps. Salah satu pengembangan dari 802.11n adalah diaplikasikannya *Multiple Input Multiple Output (MIMO)* dengan menggunakan metode *Space Division Multiplexing*.

Komunikasi Serial^[4]

Komunikasi serial terbagi menjadi 2, yaitu: Komunikasi serial sinkron dan Komunikasi serial asinkron. Komunikasi serial sinkron yaitu data dikirim bersamaan dengan sinyal *clock*. Contoh protokol yang menggunakan metode ini adalah *I²C Inter-Integrated Circuit (I²C)*, dan *Serial Peripheral Interface (SPI)*. Sedangkan, komunikasi serial asinkron yaitu data dikirim tanpa sinyal *clock*. Contoh protokol yang menggunakan metode ini adalah *Universal Asynchronous Receiver-Transmitter (UART) RS-232*.

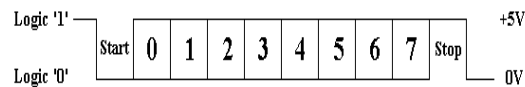
Pada modul WIZ110SR, protokol komunikasi serial yang digunakan adalah RS-232. Karakteristik elektrik dari RS-232 adalah sebagai berikut:

- *Space* adalah sebutan untuk logika 0 adalah antara -3 V sampai dengan -25 V terhadap *ground*;
- *Mark* adalah sebutan untuk logika 1 adalah antara +3 V sampai dengan +25 V terhadap *ground*;
- Daerah antara -3 V sampai dengan +3 V terhadap *ground* adalah tidak terdefinisi;
- *Single ended* terhadap *ground*;
- *Open Circuit* maksimal adalah sebesar 25 V terhadap *ground*;
- Arus *Close Circuit* maksimal adalah sebesar 500 mA.

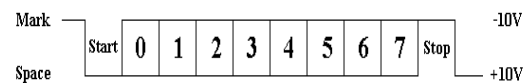
Beberapa hal penting yang perlu diperhatikan dalam protokol ini antara lain:

- Diperlukan *baud rate* yang sama pada sisi pengirim dan penerima;
- Pada pengiriman data, disertakan *start bit* sebagai penanda awal data, dan *stop bit* sebagai penanda akhir data;
- Untuk menjaga integritas data dapat disertakan *parity bit*.

Perbedaan RS-232 dengan komunikasi serial TTL/CMOS adalah pada level tegangannya. Untuk mengubah komunikasi serial asinkron dengan level tegangan TTL/CMOS, biasa digunakan *converter* tegangan. Yang umum dipakai adalah IC MAX-232. Gambar 6 adalah gambaran level tegangan pada TTL/CMOS dibandingkan dengan level tegangan RS-232.



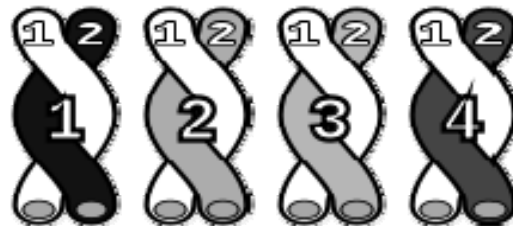
Gambar 6. (a) Level tegangan pada TTL/CMOS



Gambar 6. (b) Level tegangan pada RS-232

Kabel Unshielded Twisted-Pair^[4]

Kabel *Unshielded twisted-pair (UTP)* merupakan kabel yang saling berpasangan tujuannya adalah untuk mengurangi atau meniadakan *interferensi elektromagnetik*. Gambar 7 merupakan gambar kabel UTP.



Gambar 7. Kabel *twisted-pair*

Kabel *UTP* menggunakan bahan dasar tembaga, yang tidak dilengkapi dengan *shield* internal. Kabel *UTP* merupakan jenis kabel yang paling umum yang sering digunakan di dalam jaringan lokal (*LAN*), karena harganya yang rendah, fleksibel, dan kinerja yang ditunjukkannya relatif bagus.

Motor Servo^[4]

Motor yang digunakan adalah motor *servo*. Motor *servo* merupakan jenis motor yang dapat diatur posisi *angular* porosnya. Gambar 8 merupakan bentuk fisik motor servo.



Gambar 8. Bentuk fisik motor servo

Perangkat ini memiliki *output* berupa poros *angular*. Posisi *angular* dari poros ini dapat diatur dengan mengirimkan pulsa dengan lebar tertentu ke jalur *input* pulsa motor *servo*.

Kamera IP^[5]

Kamera *IP* adalah sebuah layanan jaringan yang berdiri sendiri yang mampu melakukan pengawasan dalam bentuk gambar yang dikontrol dari jarak jauh. Kamera *IP* bisa diakses melalui *web browser* dengan cara memasukkan alamat *IP*, *account*, dan *password*.

Teknologi kamera video berbasis *IP* ini sangat sederhana dengan menggabungkan rongga *ethernet* yang dihubungkan ke sistem jaringan, perangkat ini langsung dapat digunakan. Bahkan pada beberapa teknologi kamera video *IP* sudah dimungkinkan untuk menggunakan nirkabel karena teknologi nirkabel memiliki kecepatan *bandwidth* agar dapat mengalirkan aliran tayangan video digital. Gambar 9 merupakan gambar dari kamera *IP*.



Gambar 9. Kamera *IP*

Wireless Router^[6]

Wireless Router merupakan sebuah *router* yang juga berfungsi sebagai *wireless access point (AP)*. *Access point* pada *wireless Local Area Network (LAN)* memiliki fungsi yang mirip seperti *hub*. Tanpa adanya *access point* ini, perangkat yang mempunyai *wireless adapter* (seperti komputer) hanya dapat berkomunikasi lewat *point to point* (dua komputer atau lebih). Jaringan yang menggunakan kabel tipe *point to point* ini mirip dengan sistem jaringan tanpa kabel *hub* atau biasanya disebut dengan *cross link*. *Access point* mengeluarkan sinyal (*code*) *Service Set Identification (SSID)* dan pada semua komputer yang akan terhubung dengan *access point* tersebut harus dikonfigurasi menggunakan *SSID* yang dikeluarkan *access point* tersebut agar semua komputer dapat berkomunikasi dengan *wireless LAN* yang sama.

Router adalah peralatan utama yang digunakan dalam *Wide Area Network (WAN)*. Informasi/paket data dapat diteruskan ke alamat-alamat yang berjauhan dan berada di *network* yang berlainan. Hal itu tidak dapat dilakukan oleh *hub*, *bridge*, dan *switch*. Pada *router*, proses seleksi dilakukan pada *network layer* dari arsitektur jaringan. Artinya proses seleksi bukan pada *ethernet address*, tetapi pada lapisan yang lebih tinggi seperti: *Internet Protocol Address*.

Wireless router yang digunakan adalah merek *ProLink 3.75G*. *Wireless-N MIMO AP/HSPA Router* dapat dilihat pada Gambar 10. *Wireless route* ini memiliki spesifikasi sebagai berikut:

- Standar: *IEEE 802.11b*, *IEEE 802.11g*, *IEEE 802.11n*;
- Antarmuka: *Port RJ-45*, *5 Port* (1 *WAN/4 LAN*), Tombol *Reset*, Tombol *WPS*, *USB*;
- Jangkauan : *Indoors* bisa sampai 100 meter, dan *Outdoors* bisa sampai 300 meter;
- Power*: 100-240 *VAC*, 12 *VDC* 1,5 *A*;
- Batas Frekuensi : 2,400 – 2,4835 *GHz*.



Gambar 10. *Wireless Router*

METODE PENELITIAN

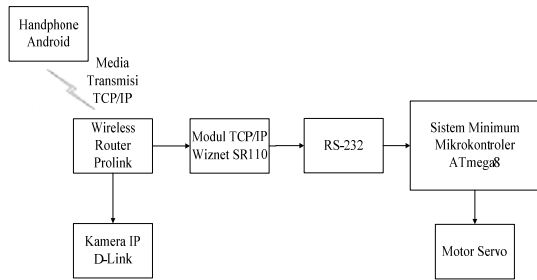
Pada pembuatan alat ini, terdapat dua bagian besar yang saling berhubungan, yaitu

perancangan *hardware*, dan *software*. Kedua bagian ini saling melengkapi untuk melaksanakan fungsi alat.

Pada perancangan *hardware*, meliputi perancangan, dan pembuatan alat yang meliputi rangkaian sistem minimum mikrokontroler *ATmega*, rangkaian *RS-232*, dan rangkaian *regulator* tegangan. Sedangkan perancangan *software* meliputi perancangan, dan pembuatan program yang akan digunakan.

Perancangan Hardware

Perancangan alat ini terbagi menjadi 2 bagian, yaitu perancangan pada bagian sisi jauh, dan bagian pengendali. Bagian sisi jauh terdiri dari kamera *IP*, mikrokontroler, *servo* dan *Wireless Router* sedangkan pada bagian pengendali menggunakan *handphone* berbasis *Android*. Diagram Blok sistem dapat dilihat pada Gambar 11 berikut:



Gambar 11. Diagram blok Alat

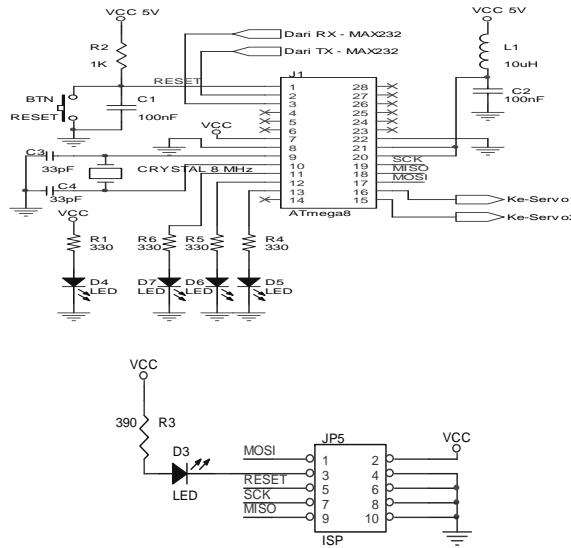
Penjelasan dari Gambar 11:

Pada sistem ini terdapat sebuah kamera *IP* yang dipasang dengan sebuah mekanik dan dapat memantau suatu tempat/area tertentu,

kemudian kamera *IP* tersebut dapat bergerak kiri-kanan, dan juga atas-bawah. Gerakan dari kamera *IP* tersebut dikendalikan sesuai dengan tombol yang akan ditekan melalui *handphone* berbasis *Android*. Pada *handphone* tersebut terdapat 5 buah tombol yang digunakan untuk menggerakkan kamera *IP* yaitu tombol atas, bawah, kiri, kanan, dan *connect*. Empat tombol berbentuk arah panah yang terdapat pada tampilan *handphone*. Apabila salah satu tombol tersebut ditekan, maka *handphone* tersebut akan mengirim data berupa *TCP/IP* melalui media transmisi *TCP/IP*, kemudian data tersebut diterima oleh modul *TCP/IP* melalui suatu alamat *IP*, dan diubahnya menjadi data serial oleh *RS-232* kemudian dikirim ke mikrokontroler. Dari mikrokontroler akan mengolah data serial tersebut untuk menentukan arah gerak dari motor *servo*, sehingga kamera *IP* yang terpasang pada motor *servo* tersebut dapat bergerak sesuai dengan pergerakan *servo*, sedangkan data dari kamera *IP* dikirim lewat *TCP/IP* ke *handphone* melalui alamat *IP* yang berbeda dengan jalur kontroler. Pada alat ini terdapat beberapa perancangan perangkat keras yang terdiri dari perancangan elektronik seperti: rangkaian sistem minimum mikrokontroler, rangkaian *RS-232*, dan rangkaian regulator tegangan.

• Rangkaian Sistem Minimum Mikrokontroler ATmega8

Mikrokontroler yang digunakan pada alat ini adalah *ATmega8*. Jumlah pin dari *ATmega8* adalah 28 pin namun pada alat ini hanya

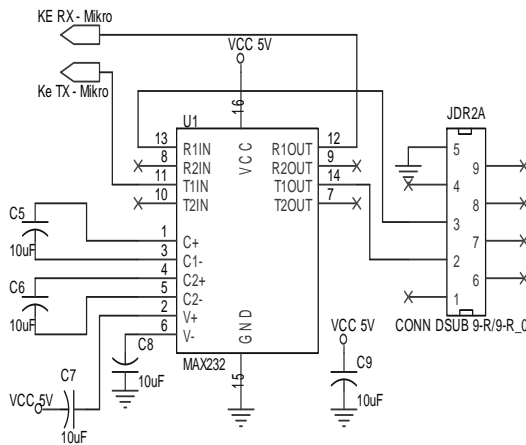


Gambar 12. Rangkaian sistem mikrokontroler *ATmega8*

menggunakan beberapa pin untuk melakukan pemrosesan data yaitu: *PORT D.0*, dan *D.1* untuk menerima data serial dari *USART RS-232*, *PORT B.1*, dan *B.2* digunakan untuk mengontrol motor *servo*. Gambar 12 merupakan gambar rangkaian sistem minimum mikrokontroler.

Perancangan Rangkaian RS-232

Sistem komunikasi antara mikrokontroler dan modul *TCP/IP* menggunakan sistem komunikasi serial *RS-232*. Oleh karena itu dibutuhkan rangkaian *RS-232*. Rangkaian ini menggunakan *IC MAX 232* yang mempunyai 2 *receiver* yang berfungsi sebagai pengubah level tegangan dari level *RS-232* ke level *Transistor Transistor Logic (TTL)*, dan mempunyai 2 *driver* yang berfungsi mengubah level tegangan dari level *TTL* ke level *RS-232*. Pasangan *driver/receiver* ini digunakan untuk *transmitter (TX)* dan *receiver (RX)*, sedangkan pasangan lainnya digunakan untuk *Clear to Send (CTS)* dan *Request to Send (RTS)*. Gambar 13 merupakan rangkaian *RS-232* di mana rangkaian ini membutuhkan tegangan sebesar 5 V untuk dapat bekerja. Pin 12, dan pin 13 pada *IC MAX 232* merupakan *receiver (RX)* yang masing-masing menerima data dari mikrokontroler dan dari modul *TCP/IP*, sedangkan pin 11, dan 14 sebagai *transmitter (TX)* yang masing-masing berfungsi untuk mengirim data dari mikrokontroler, dan dari modul *TCP/IP*.



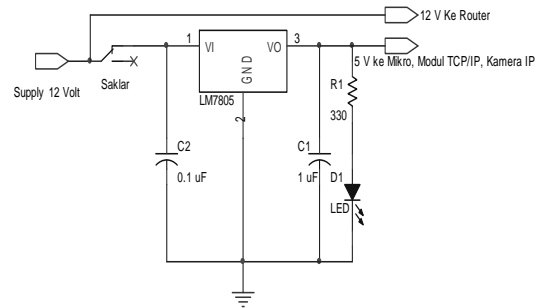
Gambar 13. Rangkaian RS-232

• **Rangkaian Regulator Tegangan**

Fungsi pemasangan regulator tegangan pada catu daya adalah untuk menyetabilkan tegangan keluaran apabila terjadi perubahan tegangan masukan pada catu daya. Fungsi lain

dari regulator tegangan adalah untuk perlindungan dari terjadinya hubung singkat pada beban.

Pada alat ini menggunakan *Power Adaptor* sebagai catu daya dengan tegangan sebesar 12 Volt dan arusnya sebesar 3 Ampere. Namun oleh karena rangkaian mikrokontroler, modul *TCP/IP*, dan kamera *IP* membutuhkan tegangan sebesar 5 Volt, sehingga menggunakan rangkaian regulator tegangan untuk menurunkan tegangan dari 12 Volt menjadi 5 Volt. Salah satu tipe regulator tegangan yang digunakan adalah *LM7805*. Regulator tegangan tipe *LM7805* adalah salah satu regulator tegangan tetap dengan tiga terminal, yaitu: terminal *VIN*, *GND*, dan *VOUT* yang menghasilkan tegangan *output* sebesar 5 Volt. *Wireless router* membutuhkan tegangan sebesar 12 Volt. Gambar 14 merupakan rangkaian regulator tegangan yang digunakan pada alat ini.



Gambar 14. Regulator Tegangan

Rangkaian regulator tegangan di atas terdapat beberapa komponen yaitu *LM7805* sebagai penyetabil tegangan, kapasitor masukan *C2* dibutuhkan untuk perata tegangan, sedangkan kapasitor keluaran *C1* memperbaiki tanggapan peralihan, dan sebuah *LED* sebagai indikator dari rangkaian.

• **Koneksi Antara Pengendali (Handphone) Dan Sisi Yang Jauh (Kamera IP dan Modul TCP/IP)**

Pada sisi yang jauh terdapat dua buah perangkat yang terhubung dengan *wireless router* yaitu kamera *IP*, dan Modul *TCP/IP*. Agar ketiga perangkat (*wireless router*, kamera *IP*, modul *TCP/IP*) tersebut dapat berkomunikasi dengan baik, maka ketiga perangkat ini harus diatur *IP*-nya agar memiliki *IP* yang berada dalam satu kelas, di mana *wireless router* memiliki *IP* 192.168.13.90, kamera *IP* memiliki *IP* 192.168.13.10, dan modul *TCP/IP* memiliki *IP* 192.168.13.2.

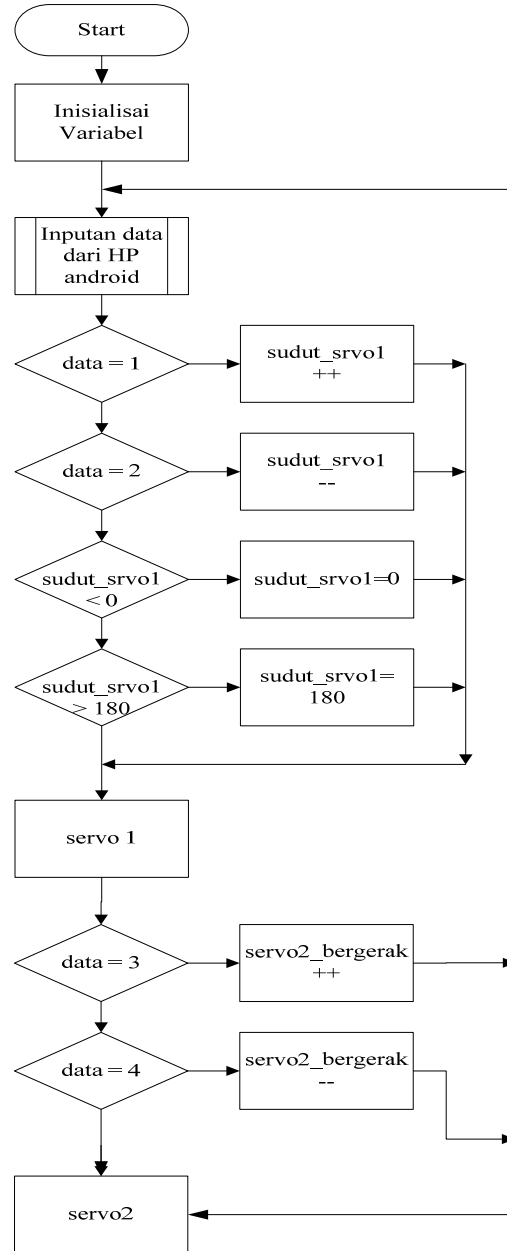
Pengaturan *IP* pada modul *TCP/IP* bertujuan agar perangkat lunak di sisi pengendali dapat mengetahui ke mana perintah pengendali harus dikirim, sedangkan pengaturan *IP* pada kamera *IP* bertujuan agar perangkat lunak di sisi pengendali dapat mengetahui di mana alamat pasti untuk pengambilan gambar pada kamera *IP* tersebut.

Perancangan Perangkat Lunak (*Software*)

Perancangan perangkat lunak (*software*) merupakan perancangan pembuatan program untuk memproses data baik untuk mengirim data maupun menerima data. Bahasa yang digunakan dalam pembuatan program adalah bahasa *C*, *Java*, dan *XML*. Bahasa *C* digunakan pada mikrokontroler *ATMega8* untuk mengendalikan motor *servo*, sedangkan bahasa *Java*, dan *XML* digunakan pada *handphone* berbasis Android untuk mengirim perintah pengendali, dan menampilkan gambar yang diambil oleh kamera *IP*. Gambar 15 merupakan diagram alir program utama dari sistem yang dibuat yaitu pada mikrokontroler. Penjelasan diagram alir utama pada Gambar 15 adalah sebagai berikut. Pada Gambar 15 terdapat 2 buah motor *servo* yaitu motor *servo* 1 dan motor *servo* 2. Motor *servo* 1 digunakan untuk menggerakkan kamera *IP* ke atas, dan bawah, sedangkan motor *servo* 2 digunakan untuk menggerakkan kamera *IP* ke kiri, dan kanan. Motor *servo* 1 menggunakan servo dengan sudut putaran berkisar 0-180 derajat, dan motor *servo* 2 menggunakan servo dengan sudut putaran berkisar 0-360 derajat.

Mula-mula mikrokontroler akan melakukan inialisasi *port I/O (input/output)*. Setelah itu, mikrokontroler menunggu data masukan dari *handphone* Android, apabila:

- Selama masukan data = 1 (menekan tombol arah panah bawah pada tampilan *handphone*), maka motor *servo* 1 akan berputar terus, hingga batas putaran sudutnya yaitu sudut 180 dan sebaliknya jika selama data masukan = 2 (menekan tombol arah panah atas pada tampilan *handphone*), maka motor servo akan berputar kembali hingga batas putaran sudutnya yaitu sudut 0 derajat.
- Sedangkan, selama masukan data = 3 (menekan tombol arah panah kiri pada tampilan *handphone*), maka motor *servo* 2 akan berputar hingga 360 derajat, dan juga masukan data = 4 (menekan tombol arah panah kanan pada tampilan *handphone*), maka motor *servo* 2 berputar kembali hingga 0 derajat.

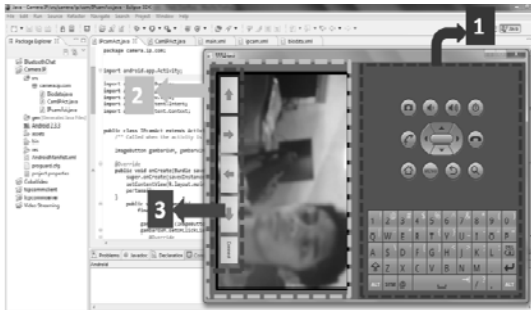


Gambar 15. Diagram alir utama sistem mikrokontroler

Perancangan Program Bagian Pengendali

Pada bagian pengendali menggunakan *handphone* berbasis Android dengan versi 2.3 (*Gingerbread*). Untuk membuat suatu aplikasi Android untuk *handphone* diperlukan *software* yang dinamakan *Eclipse*. Tipe *Eclipse* yang digunakan adalah *Eclipse Indigo* (versi 3.7.2). *Eclipse* adalah sebuah *Integrated Development Environment (IDE)* untuk mengembangkan perangkat lunak, dan dapat dijalankan di semua *platform (platform-independent)*. Agar bisa membuat aplikasi Android, maka perlu

menginstal *software-software* pada *PC personal computer (PC)* yang digunakan untuk pengembangan aplikasi Android di antaranya: menginstal *Android Software Development Kit* atau yang biasa disebut *Android SDK*, *Eclipse* (menggunakan *eclipse* versi 3.7.2 *Indigo*), *ADT plugin*, dan *Java Development Kit (JDK)*. Dengan adanya *software-software* tersebut, maka dapat digunakan untuk mensimulasi aplikasi yang telah dibuat. Apabila program telah siap untuk ditransfer ke *handphone*, maka terlebih dahulu program tersebut diekspor ke format *APK* menggunakan *software eclipse*, setelah itu *file APK* tersebut ditransfer ke *handphone* berbasis Android untuk diinstal. Gambar 16 berikut ini merupakan tampilan simulasi program di *PC* menggunakan *software eclipse*.



Keterangan Gambar:

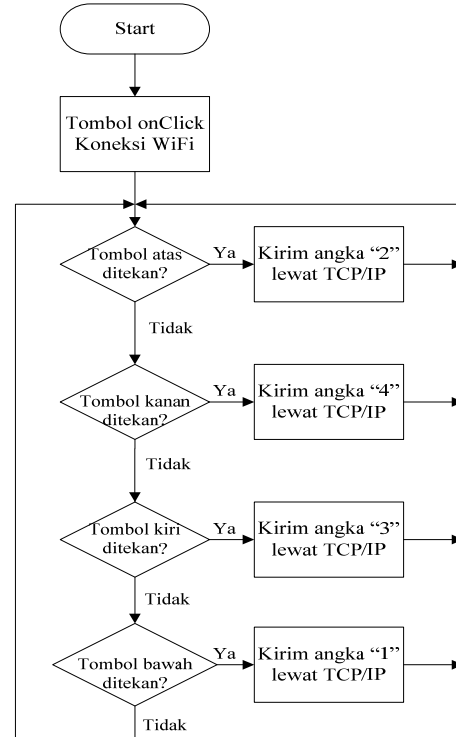
1. Keyboard pada tampilan emulator
2. Tampilan gambar yang diambil oleh kamera *IP*
3. Tombol *connect*, dan tombol arah sebagai tombol pengendali.

Gambar 16. Simulasi program pada *software eclipse*

Bahasa yang digunakan pada pemrograman adalah *Java*, dan *XML*. Bahasa *Java* digunakan untuk memproses data pada *handphone* berbasis Android seperti mengkoneksikan *handphone* dengan modul *TCP/IP* lewat *Wifi*, dan juga untuk mengirim data ketika tombol ditekan, sedangkan bahasa *XML* digunakan untuk membuat tampilan/desain aplikasi Android pada layar *handphone* seperti membuat tombol.

Pada sisi pengendali ini, *handphone* sebagai tampilan utama di mana terdapat sebuah tampilan untuk menampilkan gambar yang diambil oleh kamera *IP*, dan 5 komponen tombol yaitu 4 tombol sebagai tombol pengendali yang terdiri atas tombol atas, bawah, kiri, kanan, dan tombol ke-5 sebagai tombol *connect*. Secara garis besar, diagram alir perancangan program menggunakan *software eclipse* untuk mengirim data pengendali serta

menerima data dari kamera *IP* dapat dilihat pada Gambar 17.



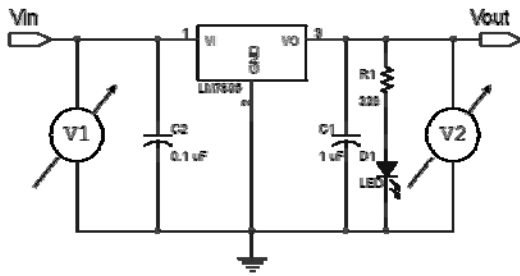
Gambar 17. Diagram alir pengendali dengan menggunakan *software eclipse*

HASIL PENELITIAN DAN PEMBAHASAN

Pengukuran dilakukan untuk mengetahui kinerja dari alat yang telah didesain dan dibuat. Pengukuran, dan pengujian yang dilakukan antara lain meliputi:

- **Pengukuran tahanan *input* dan *output* pada rangkaian regulator tegangan**

Regulator tegangan ini menggunakan *IC LM7805* dengan keluaran tegangan sebesar 5 Volt. Regulator tegangan 5 Volt ini mendapat masukan tegangan dari *power adaptor* sebesar 12 Volt. Regulator tegangan tersebut yang akan menyuplai seluruh sistem pada alat ini kecuali *wireless router*. Menurut teori regulator tegangan *LM7805* ini harusnya menghasilkan tegangan sebesar 5 Volt, namun pada saat dilakukan pengukuran terdapat perbedaan secara teori, dan melalui pengukuran menggunakan multimeter digital. Pengukuran tegangan dilakukan dengan cara mengukur tegangan *input* dan *output* pada rangkaian, dapat dilihat pada Gambar 18.



Gambar 18. Pengukuran tegangan *input* dan *output* rangkaian regulator

Berikut ini merupakan tabel regulator tegangan tanpa beban (Tabel 3), dan dengan beban (Tabel 4).

Tabel 3. Pengukuran regulator tegangan tanpa beban

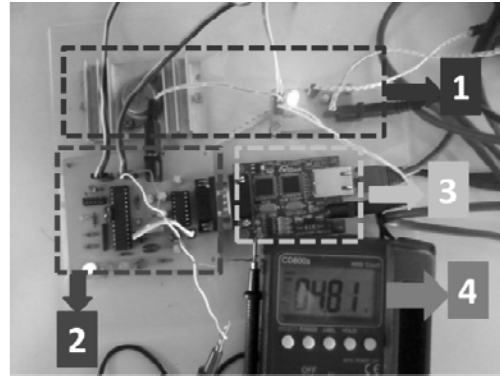
<i>Vin</i> (Volt)	<i>Vout</i> (Volt)
12,63	5,02

Tabel 4. Pengukuran regulator tegangan dengan beban

<i>Vin</i> (Volt)	<i>Vout</i> (Volt)
11,3	4,81

Untuk menghasilkan tegangan 5 Volt dengan menggunakan IC LM7805, maka tegangan masukan (*Vin*) harus lebih besar daripada 5 Volt, apabila tegangan masukan kurang daripada 5 Volt, maka IC LM7805 tersebut akan menghasilkan tegangan di bawah 5 Volt.

Gambar 19 merupakan gambar yang diambil pada saat pengukuran. Pengukuran dilakukan dengan menggunakan AVO meter digital, di mana terdapat probe positif, dan negatif pada AVO meter yang akan dihubungkan ke *output* regulator tegangan secara paralel sesuai polaritasnya. Skala yang digunakan adalah *voltmeter*. Pada Gambar 18 ditunjukkan cara pengukuran tegangan pada rangkaian regulator di mana V1 digunakan untuk mengukur tegangan masukan, dan V2 untuk mengukur tegangan keluaran. Sedangkan Gambar 19 adalah gambar yang diambil pada saat pengukuran.



Keterangan dari **Gambar 19** sebagai berikut :

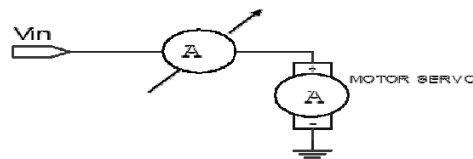
1. Rangkaian regulator tegangan;
2. Rangkaian sistem minimum mikrokontroler;
3. Rangkaian modul TCP/IP Wiznet110SR
4. Hasil pengukuran tegangan dengan menggunakan AVO meter

Gambar 19. Gambar yang diambil pada saat pengukuran

Berdasarkan pengujian di atas, maka dapat diambil kesimpulan bahwa: adanya perbedaan tegangan masukan, dan keluaran pada rangkaian regulator tegangan dikarenakan adanya penambahan beban pada rangkaian.

• **Pengukuran arus motor servo**

Pengukuran arus pada motor *servo* ini bertujuan untuk mengetahui besar arus yang mengalir pada motor *servo*. Pengukuran pada motor *servo* ini dilakukan 2 tahap yaitu pada saat motor *servo* diam, dan pada saat motor *servo* bergerak. Pengukuran dilakukan dengan menggunakan AVO meter digital. Untuk lebih jelas dapat melihat Gambar 20.



Gambar 20. Pengukuran arus pada motor *servo*

Tabel 5, dan Tabel 6 merupakan pengukuran arus pada motor *servo*.

Tabel 5. Pengukuran arus motor *servo* saat tidak bergerak

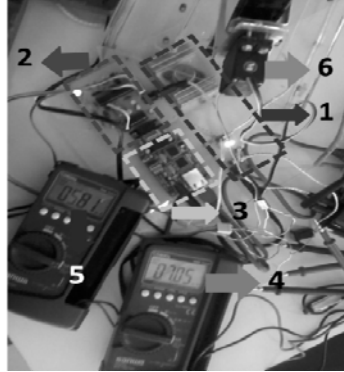
Motor <i>servo</i>	Arus (mA)
Motor <i>servo</i> 1	7,05
Motor <i>servo</i> 2	5,81

Tabel 6. Pengukuran arus motor *servo* saat bergerak

Motor <i>servo</i> 1	Arus (mA)
Bergerak atas	39
Bergerak bawah	36
Motor <i>servo</i> 2	Arus (mA)
Bergerak kiri	37
Bergerak kanan	31

Dari hasil pengukuran pada Tabel 5, dan Tabel 6 di atas terlihat bahwa terjadi perbedaan arus pada saat motor *servo* tidak bergerak, dan motor *servo* bergerak, dan kedua motor *servo* tersebut cukup kuat serta membutuhkan arus yang tidak terlalu banyak dalam melakukan tugasnya.

Berikut ini merupakan Gambar 21 yang diambil pada saat melakukan pengukuran arus pada motor *servo*.



Keterangan dari Gambar 21:

1. Rangkaian regulator tegangan
2. Rangkaian sistem minimum mikrokontroler
3. Rangkaian modul *TCP/IP Wiznet110SR*
4. Pengukuran arus pada motor *servo* 1 dengan menggunakan *AVO* meter digital
5. Pengukuran arus pada motor *servo* 2 dengan menggunakan *AVO* meter digital
6. Motor *servo* 1

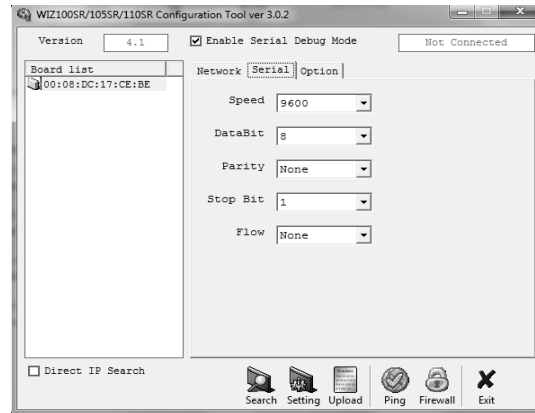
Gambar 21. Pada saat pengukuran arus motor *servo*

Pengukuran dilakukan dengan menggunakan *AVO* meter digital, di mana terdapat *probe* positif dan negatif. *Probe* tersebut dihubungkan secara seri dengan salah satu kabel *VCC* atau *GND* pada motor *servo*. Skala yang digunakan adalah ampere meter. Gambar 20 merupakan cara pengukuran arus pada motor *servo* di mana terdapat sebuah ampere meter yang digunakan untuk melakukan pengukuran arus. Sedangkan, Gambar 21 merupakan gambar yang diambil pada saat melakukan pengukuran arus motor *servo*.

• Pengujian pengiriman perintah pengendali

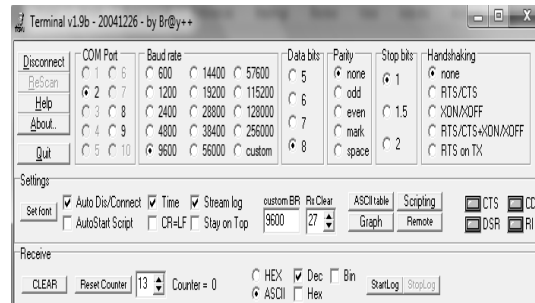
Pada bagian ini akan dilakukan pengiriman perintah dari *handphone* menuju modul *TCP/IP Wiznet110SR* melalui media transmisi *TCP/IP* dengan tujuan mengetahui data yang berasal dari *handphone* telah mencapai pada tujuan atau tidak, serta untuk menguji modul *TCP/IP* tersebut berfungsi dengan baik atau tidak. Sebelum melakukan pengiriman perintah, terlebih dahulu mengatur (*setting*)

parameter-parameter serial (seperti: *baudrate*, data bit, *parity*, *stopbit*) pada modul *TCP/IP Wiznet110SR*, dan *software* terminal pada *Personal Computer (PC)*. Untuk pengaturan pada modul *TCP/IP Wiznet110SR* menggunakan *software Configuration Tool Program*. Untuk lebih jelas dapat dilihat pada Gambar 22 dan Gambar 23.



Gambar 22. Pengaturan parameter serial pada modul *TCP/IP Wiznet110SR*

Gambar 22 di atas digunakan untuk melakukan pengaturan parameter komunikasi serial pada modul *TCP/IP*, di mana *baudrate*: 9600, data bit: 8, *parity*: none, *stopbit*: 1, *flow control*: none.



Gambar 23. Parameter serial pada *PC* menggunakan *SoftwareTerminal.exe*

Gambar 23 di atas digunakan untuk melakukan pengaturan parameter serial pada *software terminal.exe*, di mana *com*: 2, *baudrate*: 9600, data bit: 8, *parity*: none, *stop bit*: 1, *handshaking*: none.

Setelah mengatur parameter-parameter serial pada kedua *software*, maka selanjutnya melakukan pengiriman perintah dari *handphone*. Perintah yang dikirim berupa angka yaitu 1, 2, 3 dan 4. Keempat angka tersebut dapat dikirim apabila menekan tombol arah panah pada *handphone* yaitu tombol bawah kirim angka 1, tombol atas kirim angka 2, tombol kiri kirim angka 3, dan tombol kanan kirim angka 4. Gambar 24 merupakan tampilan

hasil perintah pada *software terminal* yang dikirim dari *handphone*.



Gambar 24. Hasil perintah yang dikirim dari *handphone* menggunakan *Software Terminal* pada *PC*

Gambar 24 di atas dapat dijelaskan bahwa kotak yang paling atas menunjukkan terjadinya koneksi antara *handphone* dengan modul *TCP/IP* di mana *software terminal* akan memberitahu bahwa kedua perangkat tersebut telah terhubung dengan menampilkan: “Listen : OK”, pada kotak tengah menunjukkan aplikasi yang dijalankan telah terhubung dengan modul *TCP/IP*, sehingga pada *software terminal* terdapat pemberitahuan yaitu: “Connected”, sedangkan angka-angka (1, 2, 3, 4) yang paling bawah menunjukkan perintah yang diterima oleh *software terminal*.

Berdasarkan hasil pengujian di atas dapat disimpulkan bahwa modul *TCP/IP* dapat bekerja dengan baik, dan modul tersebut dapat menerima data yang berasal dari *handphone* kemudian ditampilkan di *software terminal* pada *PC*.

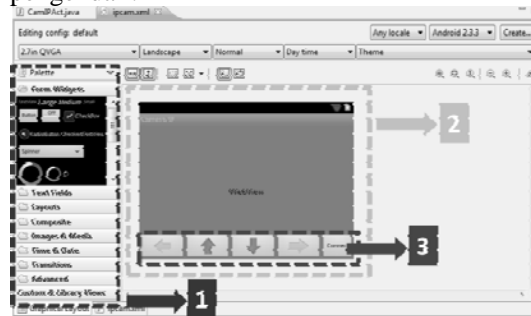
• Simulasi program pengendali

Sebelum program (aplikasi) dijalankan pada *handphone* berbasis Android terlebih dahulu disimulasikan untuk mengetahui apakah programnya berjalan atau tidak. Simulasi program pengendali menggunakan 2 *software* yaitu *software eclipse* sebagai editor untuk menulis kode program (aplikasi), dan *software Android SDK (Software Development Kit)* sebagai *tools* dan emulator untuk menjalankan simulasi aplikasi yang telah dibuat. Gambar 25 merupakan tampilan *software Android SDK*, di mana tampilan 1 adalah layar emulator, sedangkan tampilan 2 adalah *keyboard* emulator.



Gambar 25. Tampilan *software Android SDK*

Pada pembuatan aplikasi Android ini terlebih dahulu membuat sebuah tampilan dan berbagai macam komponen di dalamnya dengan menggunakan bahasa pemrograman *XML*. Komponen-komponen tersebut berupa *TextView*, *EditText*, *Button*, *RadioGroup*, *Radio Button*, *Spinner*, *CheckBox*, *WebView*, dan *ImageView*. Aplikasi ini menggunakan komponen *Button* sebagai tombol untuk mengirim perintah, dan komponen *WebView* digunakan untuk menampilkan gambar yang diambil oleh kamera *IP*. Gambar 26 berikut ini adalah tampilan komponen yang dibuat pada *XML layout*, di mana terdapat 5 buah tombol untuk melakukan pengiriman perintah pengendali.



Keterangan dari Gambar 26:

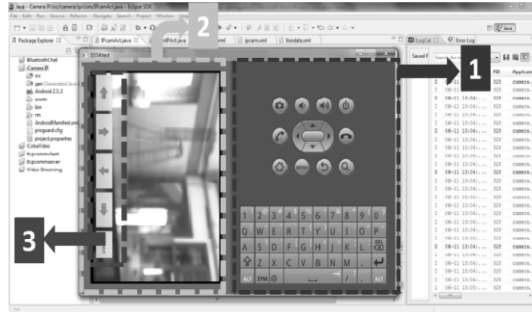
1. Komponen-komponen yang akan digunakan untuk membuat tombol;
2. *XML layout*: tempat untuk membuat tombol
3. Tombol-tombol yang digunakan untuk mengontrol motor *servo*

Gambar 26. Hasil simulasi program pengendali

Dari hasil pengujian program di atas dapat disimpulkan bahwa program/aplikasi yang telah dibuat dapat disimulasikan dengan menggunakan *software eclipse*, dan *software Android SDK*.

Jika semua program telah siap, dan tidak ada *error*, maka selanjutnya adalah mensimulasi atau menjalankan program pada *software eclipse*. Secara otomatis layar emulator (*software Android SDK*) akan muncul

untuk menampilkan hasil simulasi program seperti disajikan pada Gambar 27.



Keterangan dari Gambar 27:

1. Keyboard emulator
2. Tampilan gambar yang di ambil oleh kamera IP
3. Tombol *connect* dan tombol arah sebagai tombol pengendali.

Gambar 27. Hasil Simulasi Program

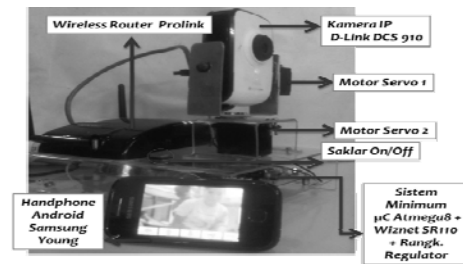
Dari hasil pengujian program di atas dapat disimpulkan bahwa program/aplikasi yang telah dibuat dapat disimulasikan dengan menggunakan *software eclipse*, dan *software Android SDK*.

• Pengujian alat secara keseluruhan

Pada bagian ini akan dilakukan pengujian alat secara keseluruhan seperti yang tampak pada Gambar 28. Langkah langkah dalam menguji alat sebagai berikut:

1. Menghidupkan alat dengan menekan tombol *on/off* pada alat tersebut, dan menghubungkan alat dengan *handphone Android Samsung Galaxy Young* melalui *wifi*. Setelah itu, menjalankan program aplikasi pada *handphone* dengan menekan tombol "*connect*". Gambar yang diambil oleh kamera IP akan langsung muncul pada layar *handphone*;
2. Selanjutnya, pada *handphone* terdapat 4 tombol arah panah yakni: atas, bawah, kiri, dan kanan yang digunakan untuk menggerakkan motor *servo* 1, dan motor *servo* 2, sehingga kamera IP yang terpasang secara mekanik dengan kedua motor *servo* tersebut dapat bergerak untuk mengambil gambar. Motor *servo* 1 menggerakkan kamera IP atas, dan bawah, sedangkan motor *servo* 2 menggerakkan kamera IP ke kiri, dan ke kanan. Rangkaian mikrokontroler digunakan untuk mengontrol motor *servo*. Rangkaian regulator tegangan untuk mencatu daya untuk seluruh sistem;
3. Gambar yang diambil oleh kamera IP tersebut akan dikirim menggunakan *wireless router* melalui *wifi*.

Pada Gambar 28 disajikan gambar pengujian alat secara keseluruhan.



Gambar 28. Pengujian keseluruhan alat

Dengan demikian dapat disimpulkan bahwa alat tersebut dapat bekerja dengan baik.

KESIMPULAN

Dari hasil perancangan, pembuatan, dan pengujian alat yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

- Alat yang telah dibuat dapat dikendalikan menggunakan *handphone* berbasis Android dengan versi 2.3 (*Gingerbread*) melalui *wireless* berbasis *TCP/IP*;
- Motor *servo* dapat menggerakkan kamera IP dengan konsumsi arus hingga 39 mA;
- Regulator tegangan dengan keluaran 4,81 mA mampu mencatu daya seluruh rangkaian pada sistem;
- *Software terminal.exe* dapat menampilkan perintah yang dikirim dari *handphone*. Perintah yang dikirim berupa : "1", "2", "3", dan "4".

DAFTAR PUSTAKA

- [1] Reto, M. *Professional Android™ Application Development*, Hlm. 1- 271 , Wiley Publishing, Canada, 2009
- [2] Wiznet, *Innovative embedded Networking*http://www.wiznet.co.kr/sub_modules/en/product/Product_Detail.asp? Diakses 10 November 2011
- [3] Atmel, *Microcontroller*, <http://www.atmel.com/>, Diakses 8 Juli 2011
- [4] Tantra, T., *Telerobot Dengan Menggunakan Media Transmisi TCP/IP*, Hlm. 6-23, Skripsi Jurusan Teknik Elektro Universitas Katolik Widya Mandala, Surabaya, 2010
- [5] Dlink, *Fast Ethernet Network Camera*, <http://www.dlink.com/products/?pid=665>. Diakses 9 Februari 2012
- [6] ProLink, *Wireless router*, www.prolink2u.com Diakses 20 Nopember 2011
- [7] Cinar, O., *Android Apps With Eclipse*, Hlm. 1-350, Apress, New York, 2012.
- [8] Texas Instrument, *Voltage Regulator LM7805*, <http://www.ti.com/product/lm78m05#description>, Diakses 20 Desember 2011