

Perancangan Sistem Pengontrolan Instalasi Listrik Apartemen Berbasis

Website



TUGAS AKHIR

*Disusun dalam rangka memenuhi salah satu persyaratan
Untuk menyelesaikan program Strata-1 Jurusan Elektro
Universitas Hasanuddin
Makassar*

Disusun Oleh :

RIDHA ANHAR

D411 07 019

NIRMALASARI A. RERANG

D411 07 050

**PROGRAM STUDI TEKNIK ENERGI LISTRIK
JURUSAN ELEKTRO FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN
MAKASSAR
2013
HALAMAN PENGESAHAN**

Perancangan Sistem Pengontrolan Instalasi Listrik Apartemen Berbasis

Website

oleh:

RIDHA ANHAR
D41107019

NIRMALASARI A. RERANG
D41107050

Tugas Akhir ini diajukan sebagai salah satu persyaratan untuk menyelesaikan program Strata Satu (S1) pada Sub Program Studi Teknik Energi Listrik, Jurusan Elektro Fakultas Teknik Universitas Hasanuddin
Makassar

Makassar, 1 February 2014

Ketua

Sekretaris

Prof. Dr. Ir. H. Ansar Suyuti,
MT
NIP. 19690124 199303 1 001

Ir. Hj. Zaenab
Muslimin, MT
NIP. 19660201 199203
2 002

Menyetujui,

Mengetahui,
Ketua Jurusan Teknik Elektro
Fakultas Teknik Universitas Hasanuddin

Dr. Ir. H. Andani Achmad, MT
NIP. 19601211 198703 1 002

ABSTRAK

Perkembangan teknologi dan pemukiman adalah dua dari banyaknya faktor yang mengakibatkan perubahan. Salah satu perilaku hidup modern yang mulai dilakukan oleh masyarakat perkotaan saat ini adalah beralihnya pola hidup dari segi pemukiman. Banyaknya masyarakat yang mulai memilih untuk tinggal di apartemen dibanding rumah tinggal. Tidak jarang para penghuni apartemen ini melupakan hal-hal kecil yang berdampak besar bagi diri sendiri dan orang lain. Salah satu contoh yang paling banyak bermunculan seperti menghidupkan pendingin ruangan atau menyalakan pemanas air beberapa menit sebelum sampai di apartemen agar saat tiba dapat langsung menikmati fasilitas yang dimiliki dan bersantai sejenak. Hal tersebut membutuhkan sistem pengontrolan jarak jauh yang akhir-akhir ini mulai banyak bermunculan. Dan salah satu sistem pengontrolan yang mulai berkembang dan banyak penggunanya adalah sistem pengontrolan berbasis website via Android.

Dalam penelitian ini akan dipaparkan tentang bagaimana merancang interface (antar muka) untuk pengontrolan perangkat elektronik, dimana interface tersebut dikontrol menggunakan komputer dan telepon selular dengan bantuan software pengontrolan untuk client dan server yang dirancang berbasis menggunakan NetBeans IDE 7.2 dan Eclipse Juno. Sehingga didapatkan sistem pengontrolan berbasis jaringan komputer yang nantinya akan dihubungkan dengan sistem koneksi Virtual Private Network (VPN), sehingga mendapatkan sistem pengontrolan peralatan elektronik yang lebih fleksibel dengan jangkauan yang luas. Sistem pengontrolan dilengkapi dengan beberapa bagian yang mendukung keamanannya sehingga tidak bias diakses oleh orang lain.

Software yang dibuat untuk server dan client dirancang lebih simpel untuk memudahkan client dalam penggunaannya. Aplikasi yang dibuat dapat bekerja dengan baik dan melakukan transfer data yang cepat dengan menggunakan jaringan internet. Transfer data dipengaruhi oleh cepat atau lambatnya koneksi dari jaringan internet yang digunakan serta komunikasi dari client dan server ke openvpn.

Kata Kunci : Piranti listrik, pengendalian, mikrokontroler, Android, ponsel

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
ABSTRAK.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah.....	4
I.3 Tujuan Penelitian.....	4
I.4 Batasan Masalah.....	4
I.5 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	7
II.1 Apartemen.....	7
II.1.1 Pengertian Apartemen.....	7
II.1.2 Karakteristik Apartemen.....	9
II.1.3 Klasifikasi Apartemen.....	10
II.2 Teori Mikrokontroller.....	12
II.2.1 AVR ATMega 16.....	13

II.2.1.1	Konfigurasi Pin ATMega 16.....	14
II.2.2	MAX 232.....	16
II.3	Relay DC.....	20
II.3.1	Prinsip Kerja Relay.....	22
II.4	NetBeans.....	23
II.4.1	Platform NETBeans.....	23
II.4.2	NETBeans IDE 7.2.....	24
II.4.3	Paket-Paket Tambahan NETBeans IDE.....	25
II.5	Code Vision AVR.....	28
II.6	Display (indikator).....	30
II.7	Android.....	32
II.7	Eclipse Juno.....	38
II.8	Open VPN.....	40
 BAB III PERANCANGAN DAN PEMBUATAN ALAT.....		43
III.1	Tempat
	dan Waktu Penelitian.....	43
III.2	Perancangan.....	43
III.3	Perancangan Perangkat Keras.....	48
III.3.1	Rangkaian Power Supply.....	48
III.3.2	Rangkaian Mikrokontroler ATMega 16.....	49
III.3.3	Rangkaian Driver Relay.....	51
III.4	Perancangan Perangkat Lunak.....	53

III.4.1 Perangkat Lunak Mikrokontroller.....	54
III.4.2 Perangkat Lunak Server.....	57
III.4.3 Perangkat Aplikasi Android (<i>Client</i>).....	61
III.5 Perancangan Koneksi Server dengan Aplikasi Android (<i>Client</i>).....	63
BAB IV HASIL DAN PEMBAHASAN.....	69
IV.1 Pengujian Perangkat Keras.....	69
IV.1.1 Pengujian Power Supply.....	69
IV.1.2 Pengujian Mikrokontroller.....	71
IV.1.3 Pengujian Driver Relay.....	72
IV.2 Pengujian Perangkat Lunak.....	74
IV.2.1 Pengujian Fungsional.....	74
IV.2.2 Pengujian Kecepatan Pengiriman Perintah Pengontrolan.....	76
IV.2.2.1 Pengendalian Melalui Server Terhadap Respon Waktu.....	77
IV.2.2.2 Pengendalian Melalui Client terhadap Respon Waktu.....	80
BAB V KESIMPULAN DAN SARAN.....	85
V.1 Kesimpulan.....	85
V.2 Saran.....	86

DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 2.1	Apartemen.....	10
Gambar 2.2	Konfigurasi Pin ATmega 16.....	15
Gambar 2.3	Rangkaian Max 232.....	17
Gambar 2.4	Konektor DB-9 Pada PC.....	18
Gambar 2.5	Konfigurasi Pin Max 232.....	20
Gambar 2.6	Relay DC.....	22
Gambar 2.7	Prinsip Kerja Relay.....	22
Gambar 2.8	Tampilan NetBeans IDE 7.2.....	28
Gambar 2.9	Tampilan Code Vision AVR.....	30
Gambar 2.10	Susunan Alamat Pada LCD.....	31
Gambar 2.11	LCD (Liquid Crystal Display).....	32
Gambar 2.12	Tampilan Android Versi 4.2 Jelly Bean.....	34
Gambar 2.13	Tampilan Eclipse Juno.....	40
Gambar 2.14	Sistem Kerja OpenVPN.....	41
Gambar 3.1	Alur Kerja Rangkaian.....	44
Gambar 3.2	Diagram Kerja Rangkaian.....	45
Gambar 3.3	Skematik Apartemen Untuk Perancangan Sistem Kontrol.....	47
Gambar 3.4	Rangkaian Skematik <i>Power Supply</i>	48
Gambar 3.5	Rangkaian <i>Power Supply</i>	48
Gambar 3.6	Rangkaian Skematik Mikrokontroler dan LCD.....	49

Gambar 3.7	Rangkaian Mikrokontroler dan LCD.....	50
Gambar 3.8	Rangkaian Skematik <i>Driver Relay</i>	52
Gambar 3.9	Rangkaian <i>Driver Relay</i>	52
Gambar 3.10	Tampilan CodeVision AVR 2.1.....	55
Gambar 3.11	<i>Flowchart</i> Program Mikrokontroler ATmega 16.....	56
Gambar 3.12	Tampilan NetBEANS IDE 7.2.....	58
Gambar 3.13	Halaman Log-In <i>Server</i>	59
Gambar 3.14	Halaman Pengaturan <i>Server</i>	60
Gambar 3.15	Halaman Pengontrolan <i>Server</i>	60
Gambar 3.16	Halaman Log-In <i>Client</i>	62
Gambar 3.17	Halaman Pengontrolan <i>Client</i>	62
Gambar 3.18	Tampilan OpenVPN GUI.....	63
Gambar 3.19	<i>Flowchart</i> Koneksi <i>Client</i> ke <i>Server</i>	64
Gambar 3.20	<i>Flowchart</i> Koneksi <i>Server</i> ke <i>Client</i> dan Mikrokontroler.....	65
Gambar 3.21	<i>Flowchart</i> Status Balikan dari <i>Server</i> ke <i>Client</i>	66
Gambar 3.22	Tampilan OpenVPN For Android.....	67
Gambar 3.23	Koneksi OpenVPN dengan <i>Server</i> dan <i>Client</i>	67
Gambar 3.24	Pemasangan Rangkaian.....	68
Gambar 3.25	Pemasangan Prototipe dengan Beban.....	68
Gambar 4.1	Titik Pengukuran Rangkaian Power Supply.....	69
Gambar 4.2	Titik Pengukuran <i>Driver</i> Beban.....	73
Gambar 4.3	Perangkat Listrik Dalam Keadaan OFF.....	76
Gambar 4.4	Perangkat Listrik Dalam Keadaan ON.....	76

Gambar 4.5	Pengujian Kinerja Melalui Server.....	77
Gambar 4.6	Pengujian Kinerja Melalui Client.....	81

DAFTAR TABEL

Tabel 2.1	Fungsi Khusus Port B ATmega 16.....	15
Tabel 2.2	Fungsi Khusus Port D ATMEga 16.....	16

Tabel 2.3	Fungsi Khusus Port C ATmega 16.....	16
Tabel 2.4	Konfigurasi Pin RS 232.....	18
Tabel 3.1	Karakter String Mikrokontroler.....	54
Tabel 4.1	Hasil Pengukuran Rangkaian Power Supply.....	70
Tabel 4.2	Hasil Pengukuran Mikrokontroler Keadaan OFF.....	71
Tabel 4.3	Hasil Pengujian Mikrokontroler Keadaan ON.....	72
Tabel 4.4	Pengujian Driver Beban Keadaan ON.....	73
Tabel 4.5	Pengujian Driver Beban Keadaan OFF.....	74
Tabel 4.6	Hasil Pengujian Fungsional Tombol Kendali Server.....	75
Tabel 4.7	Hasil Pengujian Fungsional Tombol Kendali Client.....	75
Tabel 4.8	Hasil Pengujian Melalui Server	79
Tabel 4.9	Hasil Pengujian Melalui Client.....	82

BAB 1

PENDAHULUAN

I.1 Latar Belakang

Pola hidup masyarakat di zaman modern ini sudah sangat jauh berbeda dengan pola hidup masyarakat beberapa tahun terakhir. Banyak faktor yang mengakibatkan terjadinya transformasi pola hidup masyarakat. Beberapa orang beranggapan bahwa kehidupan perkotaan tidak boleh terlepas dari kemajuan zaman dan perilaku modern. Salah satu perilaku hidup modern yang mulai dilakukan oleh masyarakat perkotaan saat ini adalah beralihnya pola hidup dari segi pemukiman. Banyaknya masyarakat yang mulai memilih untuk tinggal di apartemen dibanding rumah tinggal.

Apartemen adalah blok bangunan yang di dalamnya terbagi-bagi dalam sejumlah ruang atau unit, yang dipasarkan secara *strata-title* atau disewakan. Sebagian orang berfikir bahwa keputusan untuk tinggal di apartemen dibanding rumah tinggal adalah hal yang paling tepat. Pasalnya, kebanyakan orang yang tinggal di perkotaan adalah orang-orang yang masih sangat produktif. Efektifitas dan mobilitas adalah hal yang utama.

Tidak jarang para penghuni apartemen ini melupakan hal-hal kecil yang berdampak besar bagi diri sendiri dan orang lain. Lupa mematikan peralatan listrik apartemen adalah salah satu hal yang paling banyak terjadi. Keinginan untuk

menikmati kemudahan dan kenyamanan tempat tinggal dengan fasilitas serba otomatis dan dapat dikendalikan dari jauh tanpa perlu membawa banyak perangkat tambahan seperti remote kontrol adalah salah satu hal yang paling diinginkan oleh orang-orang dengan aktifitas yang padat. Salah satu contoh yang paling banyak bermunculan seperti menghidupkan pendingin ruangan atau menyalakan pemanas air beberapa menit sebelum sampai di apartemen agar saat tiba dapat langsung menikmati fasilitas yang dimiliki dan bersantai sejenak. Hal tersebut menjadi incaran utama para penghuni apartemen dikarenakan padatnya jadwal kerja yang harus dilakukan sehingga sekedar untuk menunggu pendingin ruang menjadi dingin atau sekedar kembali ke rumah untuk mematikan lampu disaat lupa dianggap menjadi hal yang membuang waktu.

Hal tersebut membutuhkan sistem pengendali jarak jauh yang akhir-akhir ini mulai banyak bermunculan. Para pengembang tersebut mengikuti perkembangan teknologi yang ada. Teknologi kontrol telah banyak berkembang penggunaannya dan memanfaatkan berbagai media transmisi, mulai dari media infra merah dan bluetooth yang penggunaannya terbatas oleh jarak, internet hingga jaringan telepon. Internet contohnya, merupakan salah satu kebutuhan pokok masyarakat modern saat ini. Hampir semua orang di dunia ini sudah mengenal dengan baik internet dan menjadi gaya hidup tersendiri yang tentunya sangat mudah untuk mengaksesnya dikarenakan hampir semua telepon selular yang beredar di pasaran memiliki aplikasi untuk melakukan akses ke internet.

Salah satu piranti yang harganya cukup terjangkau, banyak dimiliki oleh masyarakat modern pada umumnya dan memiliki akses paket data yang cukup cepat

untuk mengakses jaringan internet adalah telepon seluler yang berbasis pada sistem operasi Android. Berbagai kemudahan ditawarkan pada sistem ini dikarenakan Android merupakan Operating Sistem berbasis *open source*. *Open Source* adalah sistem pengembangan yang tidak dikoordinasi oleh suatu individu / lembaga pusat, tetapi oleh para pelaku yang bekerja sama dengan memanfaatkan kode sumber (*source-code*) yang tersebar dan tersedia bebas (biasanya menggunakan fasilitas komunikasi [internet](#)). Berdasarkan pengertian di atas ini berarti memungkinkan untuk dilakukan pembuatan aplikasi baru untuk digunakan sebagai aplikasi pengontrolan pada perangkat Android yang kita miliki.

Beberapa keuntungan lain dari sistem ini yaitu setelah kita melakukan beberapa perubahan pada perangkat manual dan membuat antarmuka untuk server dan penggunaannya, maka kita dapat memperoleh status penggunaan dari perangkat listrik yang terhubung dengan antarmuka. Untuk itu kami mencoba menjadikan sistem pengontrolan berbasis *website* ini sebagai bahan penelitian kami dengan memberikan judul :

“Perancangan Sistem Pengontrolan Instalasi Listrik Apartemen Berbasis Website”

I.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini antara lain:

1. Bagaimana melakukan pengontrolan jarak jauh untuk mengontrol sistem instalasi listrik apartemen berbasis website.

2. Bagaimana menganalisa dan mengevaluasi penggunaan modul antarmuka dan aplikasi pengontrolan.
3. Bagaimana pengaruh jarak terhadap waktu pengontrolan menggunakan antarmuka dan aplikasi pengontrolan.

I.3 Tujuan Penelitian

Adapun tujuan penulisan penelitian ini adalah :

1. Merancang alat serta antarmuka untuk melakukan pengontrolan sistem instalasi listrik berbasis website..
2. Merancang sistem komunikasi antara *server*, *client* dan OpenVPN.
3. Mengevaluasi pengaruh jarak terhadap waktu pengontrolan menggunakan aplikasi server dan client.

I.4 Batasan Masalah

1. Pembahasan teori mikrokontroler dan pengontrolan sistem instalasi listrik dengan sistem *switching*.
2. Perancangan antarmuka dan aplikasi pengontrolan sistem instalasi listrik berbasis *website*.
3. Perancangan antarmuka dengan menggunakan program NETBEANS IDE 7.2 dan ECLIPSE JUNO serta pembuatan sistem pengontrolan ATmega 16 dengan menggunakan CodeVision AVR 2.1.

4. Hasil perancangan berupa antarmuka dan aplikasi pengontrolan sistem instalasi listrik berbasis *website*.

I.5 Sistematika Penelitian

Laporan penelitian ini disusun dengan menggunakan sistematika sebagai berikut :

BAB I PENDAHULUAN

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

BAB II TINJAUAN PUSTAKA

Bab ini membahas teori-teori dasar yang digunakan dalam penelitian ini meliputi pengertian apartemen, mikrokontroler, android, dan bahasa-bahasa pemrograman yang digunakan.

BAB III METODE PENELITIAN

Bab ini membahas tentang perancangan modul pengontrolan, modul antarmuka dan aplikasi pengontrolan sistem *switching* menggunakan aplikasi desktop dan aplikasi android.

Bab IV HASIL DAN PEMBAHASAN

Berisi tentang hasil dan pembahasan implementasi alat, antarmuka dan aplikasi pengontrolan perangkat listrik menggunakan aplikasi android.

BAB V KESIMPULAN DAN SARAN

Merupakan bagian terakhir yang berisikan kesimpulan dan saran-saran dari hasil pembahasan yang telah diperoleh.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas tentang teori dasar apartemen, mikrokontroller, relay dc bahasa pemrograman NETBeans, CodeVision AVR 2.1 dan Eclipse Juno, Android dan OpenVPN. Penjelasan tentang apartemen mencakup pengertian dan klasifikasi apartemen, penjelasan tentang mikrokontroller mencakup tentang teori mikrokontroller, ATmega 16, Konfigurasi Pin ATmega, dan Max 232. Relay DC membahas prinsip kerja relay dc. Pada NetBeans, CodeVision AVR dan Eclipse Juno dijelaskan tentang kelebihan masing-masing bahasa pemrograman, kegunaan dan sifat-sifatnya. Selain itu, pada penjelasan mengenai android dijelaskan beberapa versi dari sistem operasi tersebut. Di bab ini juga dibahas OpenVPN yang berfungsi sebagai konektor antara sebuah aplikasi.

II.1 Apartemen

II.1.1 Pengertian Apartemen

Beberapa pengertian dari kata Apartemen sebagai berikut :

- Menurut buku Site Planning (1984 : 252), apartemen didefinisikan sebagai *“several dwelling units share a common (usually anindoor) acces and are*

enclosed by a common structural envelope”, yang berarti beberapa bagian unit hunian yang saling berbagi akses yang sama dan dilengkapi oleh struktur kulit bangunan yang sama.

- Menurut Kamus Besar Bahasa Indonesia (1993:51), apartemen didefinisikan sebagai tempat tinggal (terdiri atas kamar duduk, kamar tidur, kamar mandi, dapur dsb.) yang berada pada satu lantai bangunan bertingkat, rumah, flat, rumah pangsa: bangunan bertingkat terbagi dalam beberapa tempat tinggal.
- Suatu kompleks hunian dan bukan rumah tinggal yang berdiri sendiri (Joseph Dechiara, Time Saver Standard for Building Types).
- Sebuah ruangan atau beberapa susunan dalam beberapa jenis yang memiliki kesamaan dalam suatu bangunan yang digunakan sebagai rumah tinggal (Sein, 1967).
- Gedung bertingkat yang dibangun dalam suatu lingkungan, terbagi atas bagian-bagian yang distrukturkan secara fungsional dalam arah vertikal dan horizontal dan merupakan satuan-satuan yang dapat dimiliki dan digunakan secara terpisah yang dilengkapi dengan bagian bersama, tanah bersama dan benda bersama (Pasal UURS no. 16 Tahun 1985).

Secara umum, apartemen dapat didefinisikan sebagai suatu bangunan bertingkat lebih dari satu yang didalamnya terdapat kumpulan dari beberapa unit hunian, dengan tiap unit hunian memiliki ruang untuk hidup yang lengkap dimana para penghuninya saling berbagi fasilitas yang sama.

Apartemen merupakan salah satu variasi jenis hunian yang diminati masyarakat terutama yang tinggal di kota-kota besar. Jika dahulu rumah biasa (*landed house*) menjadi primadona pilihan tempat tinggal, kini kecenderungan ini sedikit demi sedikit mulai bergeser. Hal tersebut bukan disebabkan oleh faktor tren, melainkan timbul masalah permukiman di perkotaan yang kian pelik. Hal tersebut menyebabkan, apartemen yang merupakan hunian vertikal menjadi alternatif yang layak bagi pengembang perumahan di wilayah pusat kota untuk dapat memenuhi kebutuhan masyarakat terhadap tempat tinggal. Bagi masyarakat kota, tinggal di apartemen sebenarnya bukanlah hal yang istimewa. Tinggal di apartemen sama seperti tinggal di kompleks perumahan bahkan fasilitas yang tersedia pun hampir sama. Yang menjadi perbedaan adalah bentuknya. Apartemen berbentuk vertikal sehingga penggunaan lahan lebih efisien dan merupakan solusi yang paling ideal untuk menyelesaikan masalah pemukiman di kota.

II.1.2 Karakteristik Apartemen

Ada beberapa hal yang membedakan antara satu apartemen dengan apartemen lainnya, misalnya tinggi bangunan, penampilan fisik, fasilitas yang disediakan, struktur yang digunakan, dan kelas apartemen, namun secara garis besar apartemen memiliki ciri-ciri sebagai berikut :

- Memiliki jumlah lantai lebih dari satu.
- Terdiri dari beberapa unit hunian dalam satu lantai.
- Setiap unit terdiri atas minimal 3 macam ruang yaitu ruang tidur, dapur dan kamar mandi.

- Setiap penghuni akan saling berbagi fasilitas yang ada pada apartemen.
- Sirkulasi vertikal berupa tangga atau lift, sementara sirkulasi horizontalnya berupa koridor.
- Setiap unit akan mendapatkan jendela yang menghadap keluar.



Gambar 2.1 Apartemen (PT. Mahardika Propertindo)

II.1.3 Klasifikasi Apartemen

Berdasarkan kategori jenis dan besar bangunan Apartemen terdiri atas “

1. *High-Rise* Apartemen. Bangunan apartemen yang terdiri atas lebih dari sepuluh lantai. Dilengkapi area parkir bawah tanah, sistem keamanan dan servis penuh. Struktur apartemen lebih kompleks sehingga desain unit apartemen cenderung standar. Jenis ini banyak dibangun dipusat kota.
2. *Mid-Rise* Apartemen. Bangunan apartemen yang terdiri dari tujuh sampai dengan sepuluh lantai. Jenis ini lebih sering dibangun di kota satelit.

3. *Low-Rise* Apartemen. Apartemen dengan ketinggian kurang dari tujuh lantai dan menggunakan tangga sebagai alat transportasi vertikal. Biasanya untuk golongan menengah ke bawah.
4. *Walked-Up* Apartemen. Bangunan apartemen yang terdiri atas tiga lantai sampai dengan enam lantai. Apartemen ini kadang-kadang memiliki lift, tetapi bisa juga tidak. Jenis apartemen ini disukai oleh keluarga yang besar (keluarga inti ditambah dengan orang tua). Gedung apartemen hanya terdiri dari dua atau tiga unit apartemen.
5. *Garden* Apartemen. Bangunan apartemen dua sampai empat lantai. Apartemen ini memiliki halaman dan taman disekitar bangunan. Apartemen ini sangat cocok untuk keluarga inti yang memiliki anak kecil karena anak-anak dapat mudah mencapai ke taman. Biasanya untuk golongan menengah ke atas.

Klasifikasi Apartemen berdasarkan kepemilikan.

1. Apartemen Sewa. Pemilik membangun dan membiayai operasi serta perawatan bangunan, penghuni membayar uang sewa selama jangka waktu tertentu.
2. Apartemen Kondominium. Penghuni membeli dan mengelola unit yang menjadi haknya, tidak ada batasan bagi penghuni untuk menjual kembali atau menyewakan kembali unit miliknya. Penghuni biasanya membayar uang pengelolaan ruang bersama yang dikelola oleh pemilik gedung.
3. Apartemen Koperasi. Apartemen dimiliki oleh koperasi, penghuni memiliki saham didalamnya. Sesuai dengan unit yang ditematinya. Bila

penghuni pindah, ia dapat menjual sahamnya kepada koperasi atau calon penghuni baru dengan persetujuan koperasi. Biaya operasional dan pemeliharaan ditanggung oleh koperasi.

II.2 Teori Mikrokontroler

Mikrokontroler, sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, hadir memenuhi kebutuhan pasar (*market need*) dan teknologi baru. Sebagai teknologi baru, yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara massal (dalam jumlah banyak) membuat harganya menjadi lebih murah (dibandingkan mikroprosesor).

Tidak seperti sistem komputer, yang mampu menangani berbagai macam program aplikasi (misalnya pengolah kata, pengolah angka dan lain sebagainya), mikrokontroler hanya bisa digunakan untuk suatu aplikasi tertentu saja (hanya satu program saja yang bisa disimpan). Perbedaan lainnya terletak pada perbandingan RAM dan ROM. Pada sistem komputer perbandingan RAM dan ROM-nya besar, artinya program-program *user* disimpan dalam ruang RAM yang relatif besar, sedangkan program-program antarmuka perangkat keras disimpan dalam ruang ROM yang kecil. Sedangkan pada mikrokontroler, perbandingan ROM dan RAM yang besar artinya program kontrol disimpan dalam ROM (bisa *Masked ROM* atau *Flash PEROM*) yang ukurannya relatif lebih besar, sedangkan RAM digunakan sebagai

tempat penyimpanan sementara, termasuk register-register yang digunakan pada mikrokontroler yang bersangkutan.

Adapun kelebihan dari mikrokontroler adalah sebagai berikut :

1. Penggerak pada mikrokontroler menggunakan bahasa pemrograman *assembly* dengan berpatokan pada kaidah digital dasar sehingga pengoperasian sistem menjadi sangat mudah dikerjakan sesuai logika sistem.
2. Mikrokontroler tersusun dalam suatu *chip* dimana prosesor, memori, dan I/O terintegrasi menjadi suatu kesatuan kontrol sistem sehingga mikrokontroler dapat dikatakan sebagai komputer mini yang dapat berkerja secara inovatif sesuai dengan kebutuhan sistem.
3. Sistem *running* bersifat berdiri sendiri tanpa tergantung dengan komputer sedangkan parameter komputer hanya digunakan untuk *download* perintah instruksi atau program. Langkah-langkah untuk *download* komputer dengan mikrokontroler sangat mudah digunakan karena tidak menggunakan banyak perintah.
4. Pada mikrokontroler tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem.
5. Harga untuk memperoleh alat ini lebih murah dan lebih mudah didapat.

II.2.1 AVR Atmega16

Salah satu mikrokontroler yang banyak digunakan saat ini yaitu mikrokontroler AVR. AVR adalah mikrokontroler RISC (*Reduce Instruction Set Compute*) 8 bit berdasarkan arsitektur Harvard. Secara umum mikrokontroler AVR

dapat dikelompokkan menjadi 3 kelompok, yaitu keluarga AT90Sxx, ATmega, dan ATtiny. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fiturnya.

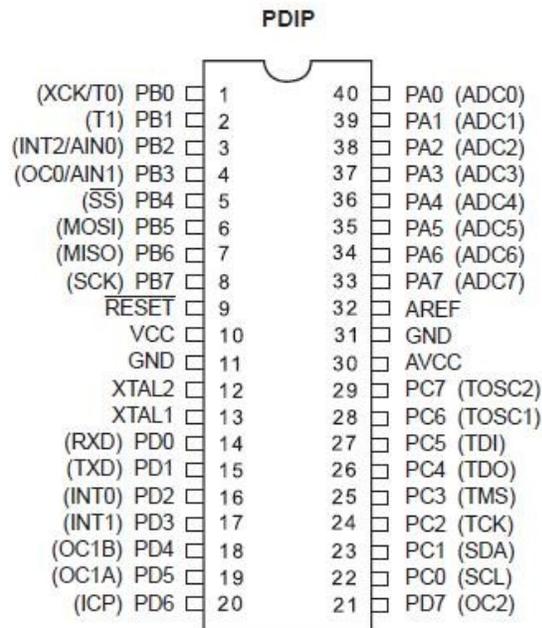
Seperti mikroprosesor pada umumnya, secara internal mikrokontroler ATmega16 terdiri atas unit-unit fungsionalnya *Arithmetic and Logical Unit (ALU)*, himpunan register kerja, register dan *decoder* instruksi dan pewaktu beserta komponen kendali lainnya. Berbeda dengan mikroprosesor, mikrokontroler menyediakan memori dalam unit yang sama dengan prosesornya (*in chip*).

II.2.1.1 Konfigurasi Pin ATmega 16

ATmega16 memiliki 32 pin yang digunakan untuk *input/output*, pin-pin tersebut terdiri dari 8 pin sebagai *port A*. 8 pin sebagai *port B*. 8 pin sebagai *port C*. 8 pin sebagai *port D*. Dalam komunikasi serial, maka hanya *port D* yang dapat digunakan karena fungsi khusus yang dimilikinya. Untuk lebih jelas akan ditunjukkan pada tabel-tabel fungsi khusus *port*. Susunan pin Mikrokontroler ATmega16 diperlihatkan pada gambar 2.2.

- Pin 1 sampai 8 (**Port B**) merupakan *port* paralel 8 bit dua arah (*input/output*) dan pin fungsi khusus.
- Pin 9 (**Reset**) merupakan pin yang digunakan untuk mereset mikrokontroler.
- Pin 10 (**VCC**) merupakan pin masukan catu daya.
- Pin 11 & 31 (**GND**) merupakan pin *ground*.
- Pin 12 (**XTAL2**) & Pin 13 (**XTAL1**) merupakan pin masukan *clock* eksternal.
- Pin 14 sampai 21 (**Port D**) merupakan *port* paralel 8 bit dua arah (*input/output*) dan pin fungsi khusus.

- Pin 22 sampai 29 (**Port C**) merupakan *port* paralel 8 bit dua arah (*input/output*) dan pin fungsi khusus.
- Pin 30 (**AVCC**) merupakan pin masukan tegangan untuk ADC.
- Pin 32 (**AREF**) merupakan pin masukan tegangan referensi ADC.
- Pin 33 sampai 40 (**Port A**) merupakan pin I/O dua arah dan pin masukan ADC



Gambar 2.2 Konfigurasi Pin ATmega16 (Budiharto, 2008)

Tabel 2.1 Fungsi *Port B* ATmega16

Port Pin	Alternate Functions
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	\overline{SS} (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input/Output)

Tabel 2.2 Fungsi *Port D* ATmega16

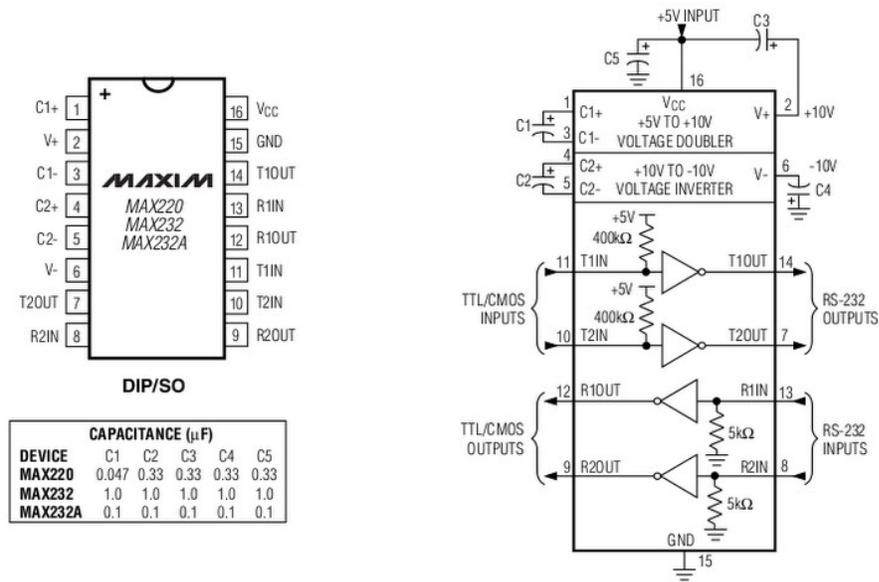
Port Pin	Alternate Function
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

Tabel 2.3 Fungsi Khusus *Port C* ATmega16

Port Pin	Alternate Function
PC7	TOSC2 (Timer Oscillator Pin 2)
PC6	TOSC1 (Timer Oscillator Pin 1)
PC5	TDI (JTAG Test Data In)
PC4	TDO (JTAG Test Data Out)
PC3	TMS (JTAG Test Mode Select)
PC2	TCK (JTAG Test Clock)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)

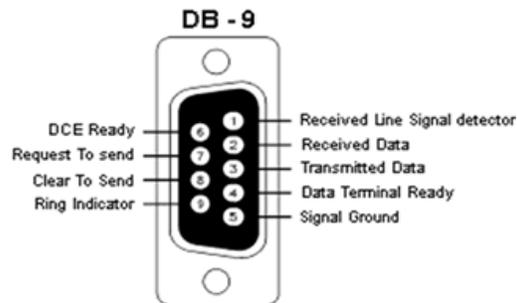
II.2.2 Max 232

MAX232 adalah saluran *driver/receiver* ganda yang termasuk pembangkit tegangan kapasitif yang menyediakan level tegangan RS232 dari sebuah sumber tegangan 5V. Setiap *receiver* pada IC MAX232 ini mengkonversikan level tegangan RS232 ke level tegangan TTL/CMOS sebesar 5V. Dan setiap *receiver* ini mempunyai ambang batas sebesar 1.3 V, dan histeresis sebesar 0.5 V, serta dapat menerima masukan level tegangan ± 30 V. Sedangkan untuk setiap *driver* pada IC MAX232 ini mengkonversikan level tegangan masukan TTL/CMOS menjadi level tegangan RS232.



Gambar 2.3 Rangkaian Max 232 (Retna dkk, 2004)

Pada dasarnya *port serial* komputer menggunakan level RS-232, RS (*Recommended Standard*) dikeluarkan oleh EIA (*Electronics Industry Association*). Dalam logika RS-232 logika 1 dinyatakan sebagai *mark* dengan level tegangan antara -3V dan -25V (negatif), sedangkan logika 0 dinyatakan sebagai *space* dengan level tegangan antara 3V dan 25V (positif). Konektor DB-9 (gambar 2.4) pada bagian belakang komputer PC adalah *serial port* RS-232 yang biasa dinamai COM1 atau COM2.



Gambar 2.4 Konektor RS 232 pada PC (Mujahidin, 2009)

Tabel 2.4 Konfigurasi Pin RS-232

Pin	Nama Pin	Fungsi	Level
1	GND	<i>Ground</i>	
2	Vcc	Tegangan input	+3.3~5.5V
3	RXD/TTL	Input Serial data	TTL
4	TXD/TTL	Output Serial data	TTL
5	DGND	Digital grounding	
6	A(TXD)	A of RS-485 or TXD of RS-232	
7	B(RXD)	B of RS-485, RXD of RS-232	
8	SLEEP	Sleep control (input)	TTL
9	RESET	Reset (input)	TTL

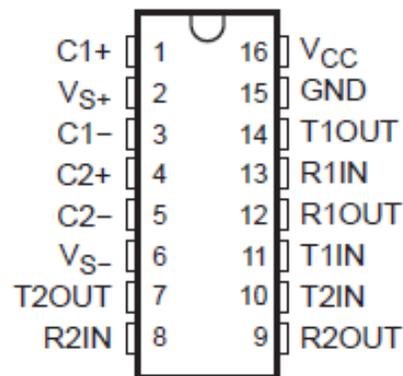
Keterangan mengenai saluran RS-232 pada konektor DB9 adalah sebagai berikut :

1. *Received Line Signal Detect*, dengan saluran ini DCE memberitahukan ke DTE bahwa terminal masukan ada data yang masuk.
2. *Received Data*, digunakan DTE menerima data dari DCE.
3. *Transmite Data*, digunakan DTE mengirimkan data ke DCE.
4. *Data Terminal Ready*, pada saluran ini DTE memberitahukan kesiapan terminalnya.
5. *Signal Ground*, saluran *Ground*.
6. *Ring Indicator*, pada saluran ini DCE memberitahukan ke DTE bahwa stasiun menghendaki hubungan dengannya.
7. *Clear to Send*, dengan saluran ini DCE memberitahukan ke DTE boleh mengirimkan data.
8. *Request to Send*, dengan saluran ini DCE diminta mengirimkan data oleh DTE.
9. *DCE Ready*, sinyal aktif pada saluran ini menunjukkan bahwa DCE sudah siap.

Dengan adanya perbedaan level tegangan TTL mikrokontroler dengan level tegangan RS-232, maka *port* serial mikrokontroler tidak bisa secara langsung dihubungkan dengan *port input*, oleh sebab itu diperlukan sebuah pengubah level logika dari TTL ke RS-232.

IC MAX232 dari Maxim Incoporation adalah IC pengubah level TTL menjadi RS-232 atau sebaliknya, yang memiliki sebuah *charge pump* yang bisa menghasilkan tegangan +10V dan -10V dari tegangan catu daya 5V. Tegangan tegangan ini dihasilkan dengan proses pengisian dan pembuangan empat kapasitor luar yang dihubungkan dengan rangkaian pengganda tegangan internal yang dimiliki IC ini. MAX232 mempunyai 2 penerima (RS-232 ke TTL) dan 2 pengirim (TTL ke RS232), cukup untuk menghubungkan pin TXD dan RXD mikrokontroler.

Pada umumnya hanya 1 pengirim dan 1 penerima yang dipakai, baik untuk level RS232 atau TTL. Pin yang digunakan diantaranya Pin 11 (T1IN) sebagai *Input* (TTL), Pin 12 (R1OUT) sebagai *Output* (TTL), Pin 14 (T1OUT) sebagai *Input* (RS-232) dan Pin 13 (R1IN) digunakan sebagai *Output* (RS-232). Konfigurasi pin MAX 232 sebagai berikut :



Gambar 2.5 Konfigurasi Pin MAX 232 (Texas Instruments Incorporated,2002)

II.3 Relay DC

Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip, relay merupakan tuas saklar dengan lilitan kawat pada batang besi (solenoid) di dekatnya. Ketika solenoid dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada solenoid sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka.

Relay biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4 Ampere 220 V_{DC}) dengan memakai arus/tegangan yang kecil (misalnya 0.1 Ampere 12 V_{DC}). Relay yang paling sederhana ialah relay elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik. Secara sederhana relay elektromekanis ini didefinisikan sebagai berikut :

- Alat yang menggunakan gaya elektromagnetik untuk menutup (atau membuka) kontak saklar.
- Saklar yang digerakkan (secara mekanis) oleh daya/energi listrik.

Dalam pemakaiannya biasanya relay yang digerakkan dengan arus DC dilengkapi dengan sebuah dioda yang di-paralel dengan lilitannya dan dipasang terbalik yaitu anoda pada tegangan (-) dan katoda pada tegangan (+). Ini bertujuan untuk mengantisipasi sentakan listrik yang terjadi pada saat relay berganti posisi dari ON ke OFF agar tidak merusak komponen di sekitarnya. Konfigurasi dari kontak-kontak relay ada tiga jenis, yaitu:

- *Normally Open* (NO), apabila kontak-kontak tertutup saat relay dicatu

- *Normally Closed (NC)*, apabila kontak-kontak terbuka saat relay dicatu.
- *Change Over (CO)*, relay mempunyai kontak tengah yang normal tertutup, tetapi ketika relay dicatu kontak tengah tersebut akan membuat hubungan dengan kontak-kontak yang lain.

Penggunaan relay perlu memperhatikan tegangan pengontrolnya serta kekuatan relay men-*switch* arus/tegangan. Biasanya ukurannya tertera pada *body* relay. Misalnya relay $12V_{DC}/4A$ 220V, artinya tegangan yang diperlukan sebagai pengontrolnya adalah $12V_{DC}$ dan mampu men-*switch* arus listrik (maksimal) sebesar 4 ampere pada tegangan 220 Volt. Sebaiknya relay difungsikan 80% saja dari kemampuan maksimalnya agar aman, lebih rendah lagi lebih aman.

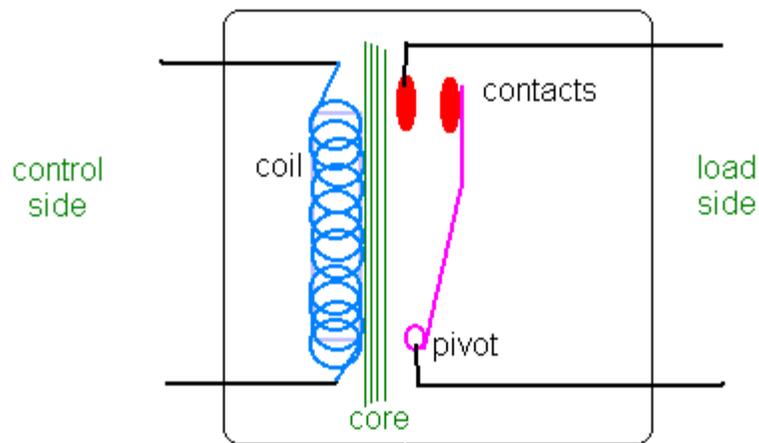
Relay jenis lain ada yang namanya *reedswitch* atau relay lidi. Relay jenis ini berupa batang kontak terbuat dari besi pada tabung kaca kecil yang dililitin kawat. Pada saat lilitan kawat dialiri arus, kontak besi tersebut akan menjadi magnet dan saling menempel sehingga menjadi saklar yang ON. Ketika arus pada lilitan dihentikan medan magnet hilang dan kontak kembali terbuka (OFF).



Gambar 2.6 Relay DC ([Yueqing Haboo Automation Co., Ltd.](#))

II.3.1 Prinsip Kerja Relay

Relay terdiri dari *Coil & Contact*. *Coil* adalah gulungan kawat yang mendapat arus listrik, sedang *contac* tadalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*. *Contact* ada 2 jenis : *Normally Open* (kondisi awal sebelum diaktifkan *open*), dan *Normally Closed* (kondisi awal sebelum diaktifkan *close*). Secara sederhana berikut ini prinsip kerja dari relay : ketika *Coil* mendapat energi listrik (*energized*), akan timbul gaya elektromagnet yang akan menarik armatur yang berpegas, dan *contact* akan menutup.



Gambar 2.7 Prinsip Kerja Relay (Kilian, Christopher 1996)

II.4 NetBeans

Mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop java, dan sebuah Integrated Development Environment (IDE) yang dibangun menggunakan platform NetBeans. NetBeans mengacu pada dua hal, yakni platform untuk pengembangan aplikasi *desktop* java, dan sebuah *Integrated Development Environment* (IDE) yang dibangun menggunakan platform NetBeans.

Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak modular yang disebut 'modul'. Modul adalah suatu arsip Java (*Java archive*) yang memuat 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*.

II.4.1 Platform NetBeans

Platform NetBeans adalah *framework* yang dapat digunakan kembali (*reusable*) untuk menyederhanakan pengembangan aplikasi *desktop*. Ketika aplikasi berbasis platform NetBeans dijalankan, kelas *main* dari platform dieksekusi. Modul-modul yang tersedia ditempatkan di sebuah *registry* di dalam memori, dan tugas *startup* modul dijalankan. Secara umum, kode modul dimuatkan ke dalam memori hanya ketika ia diperlukan.

Aplikasi dapat menginstal modul secara dinamis. Aplikasi dapat memasukkan modul *Update Center* untuk memungkinkan *user* men-download *digitally-signed upgrade* dan fitur-fitur baru secara langsung ke dalam aplikasi yang berjalan. Penginstalan kembali sebuah *upgrade* atau rilis baru tidak memaksa *user* untuk men-download keseluruhan aplikasi lagi.

Platform NetBeans menawarkan layanan-layanan yang umum bagi aplikasi *desktop*, memungkinkan pengembang untuk fokus ke logika yang spesifik terhadap aplikasi. Fitur-fitur yang disediakan oleh platform NetBeans:

- Manajemen antarmuka (misal: *menu & toolbar*)
- Manajemen pengaturan *user*
- Manajemen penyimpanan (menyimpan dan membuka berbagai macam data)
- Manajemen jendela
- *Wizard framework* (mendukung dialog langkah demi langkah)

II.4.2 NetBeans IDE 7.2

NetBeans IDE 7.2 adalah IDE *open source* yang ditulis sepenuhnya dengan bahasa pemrograman Java menggunakan platform NetBeans. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (J2SE, web, EJB, dan aplikasi *mobile*). Fitur lainnya adalah sistem proyek berbasis *Ant*, kontrol versi, dan *refactoring*.

Modularitas: Semua fungsi IDE disediakan oleh modul-modul. Tiap modul menyediakan fungsi yang didefinisikan dengan baik, seperti dukungan untuk bahasa pemrograman Java, editing, atau dukungan bagi CVS. NetBeans memuat semua modul yang diperlukan dalam pengembangan Java dalam sekali *download*, memungkinkan *user* untuk mulai bekerja sesegera mungkin. Modul-modul juga memungkinkan NetBeans untuk bisa dikembangkan. Fitur-fitur baru, seperti dukungan untuk bahasa pemrograman lain, dapat ditambahkan dengan menginstal modul

tambahan. Sebagai contoh, *Sun Studio*, *Sun Java Studio Enterprise*, dan *Sun Java Studio Creator* dari *Sun Microsystems* semuanya berbasis NetBeans IDE.

II.4.3 Paket-Paket Tambahan NetBeans IDE

NetBeans Mobility Pack

NetBeans Mobility Pack adalah alat untuk mengembangkan aplikasi yang berjalan pada perangkat bergerak (*mobile*), umumnya telepon seluler, tetapi juga mencakup PDA, dan lain-lain. *NetBeans Mobility Pack* dapat digunakan untuk menulis, menguji, dan *debugging* aplikasi untuk perangkat bergerak yang menggunakan teknologi berplatform *Java Micro Edition* (platform Java ME). Paket ini mengintegrasikan dukungan terhadap *Mobile Information Device Profile* (MIDP) 2.0, *Connected Limited Device Configuration* (CLDC) 1.1, dan *Connected Device Configuration* (CDC). Emulator dari pihak ketiga dapat diintegrasikan dengan mudah untuk lingkungan pengujian yang lebih kokoh. *NetBeans Mobility Pack* saat ini tersedia dalam dua kelas yang berbeda, yang satu memuat CDC dan yang lainnya CLDC.

NetBeans Profiler

NetBeans Profiler adalah alat untuk mengoptimalkan aplikasi Java, membantu menemukan kebocoran memori dan mengoptimalkan kecepatan. *Profiler* ini berdasarkan sebuah proyek riset *Sun Laboratories* yang dahulu bernama *Jfluid*. Riset tersebut mengungkap teknik tertentu yang dapat digunakan untuk menurunkan *overhead* proses *profiling* aplikasi Java. Salah satu dari teknik tersebut adalah instrumentasi *i code byte* dinamis, yang berguna untuk *profiling* aplikasi Java yang

besar. Dengan menggunakan instrumentasi *code byte* dinamis dan algoritma-algoritma tambahan, *Netbeans Profiler* mampu mendapatkan informasi *runtime* aplikasi yang terlalu besar atau kompleks bagi *profiler* lain. NetBeans IDE 6.0 akan mendukung *Profiling Point* yang memungkinkan untuk memprofilkan titik yang tepat dari eksekusi dan mengukur waktu eksekusi.

NetBeans C/C++ Pack

NetBeans C/C++ Pack menambahkan dukungan terhadap pengembang C/C++ ke NetBeans IDE 5.5. Paket ini memperbolehkan pengembang menggunakan sekumpulan kompiler dan alat sendiri bersama dengan NetBeans IDE untuk membangun aplikasi *native* untuk MS Windows, Linux, dan Solaris. Paket ini membuat editor mengenali bahasa C/C++ dan menyediakan *project template*, *browser* kelas yang dinamis, dukungan pembuatan file dan fungsionalitas *debugger*. Para pengembang juga dapat mengembangkan paket tersebut dengan fungsionalitas tambahan mereka sendiri.

NetBeans Enterprise Pack

NetBeans Enterprise Pack memperluas dukungan terhadap pengembangan aplikasi perusahaan dan *web service* di NetBeans IDE 7.2. Enterprise Pack ini mengembangkan kemampuan untuk menulis, menguji, dan *debug* aplikasi dengan arsitektur berorientasi layanan (*Service-Oriented Architecture*) menggunakan XML, BPEL, dan Java *web service*. Paket ini menambahkan alat desain visual untuk pemodelan UML, skema XML, dan *web service orchestration*, juga dukungan untuk *web service* dengan menggunakan identitas yang aman. Paket ini juga menginstal dan

mengkonfigurasi *runtime* yang diperlukan, termasuk mesin BPEL dan *server* manajemen identitas yang terintegrasi dengan *Sun Java Sistem Application Server*.

NetBeans JavaScript Editor

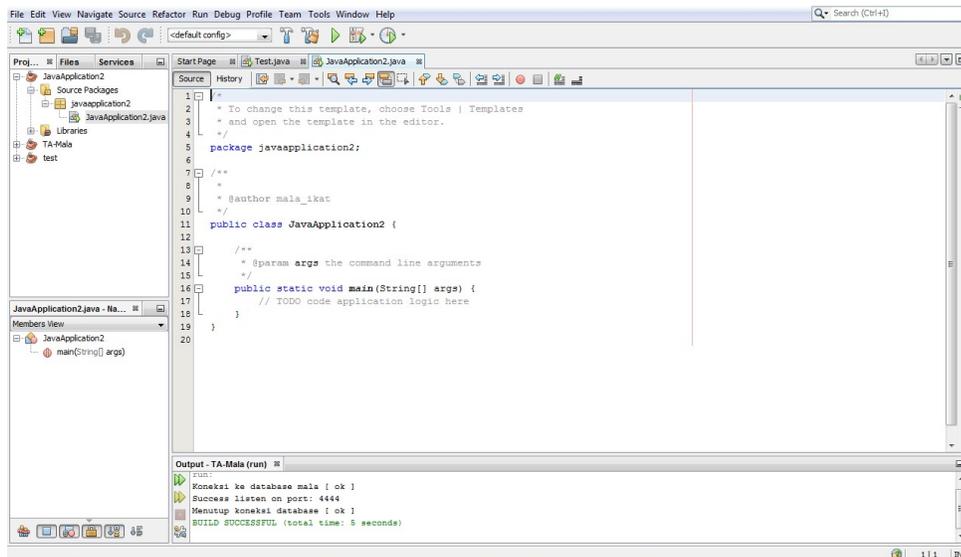
NetBeans JavaScript Editor menyediakan perluasan dukungan terhadap *JavaScript* dan *CSS*. Fitur-fiturnya antara lain:

1. Editor JavaScript

- * *syntax highlighting*
- * pelengkapan kode untuk objek dan fungsi *native*
- * semua fitur dalam editor NetBeans
- * pembuatan kerangka kelas *JavaScript* secara otomatis
- * pembuatan pemanggilan *AJAX* dari *template*

2. Ekstensi editor CSS

- * pelengkapan kode untuk nama-nama *style*
- * navigasi cepat melalui panel navigator
- * penampilan deklarasi aturan *CSS* di *List View*
- * penampilan struktur file di *Tree View*
- * mengurutkan *outline view* berdasarkan nama, tipe, atau urutan deklarasi (*List & Tree*)
- * pembuatan deklarasi aturan (hanya *Tree*)
- * pemfaktoran kembali sebagian nama *rule* (hanya *Tree*)



Gambar 2.8 Tampilan NETBeans IDE 7.2 (Oracle Corporation, 2012)

II.5 Code Vision AVR 2.1

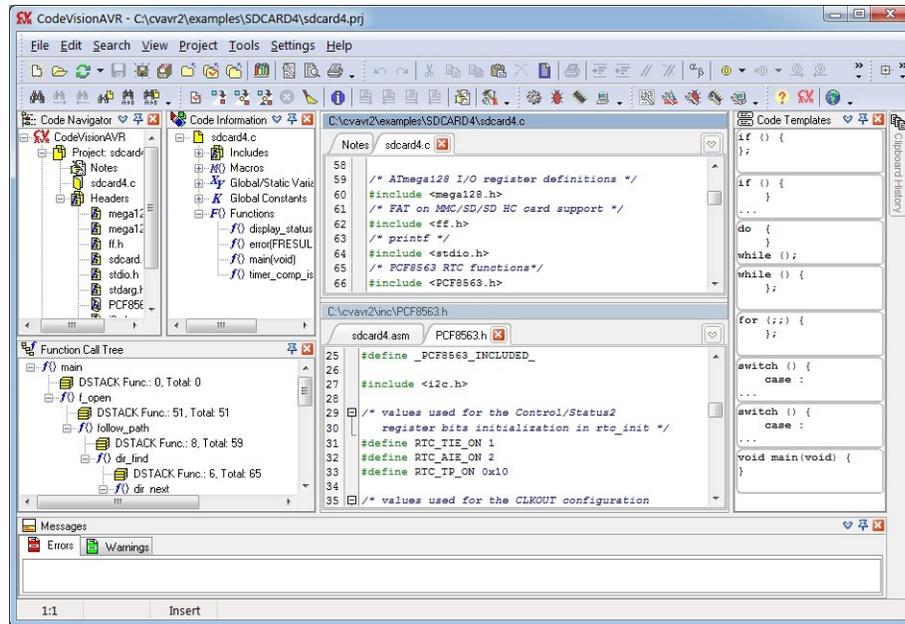
CodeVision AVR adalah sebuah *cross-compiler C*, *Integrated Development Environment (IDE)*, dan *Automatic Program Generator* yang didesain untuk mikrokontroler buatan Atmel seri AVR. *CodeVisionAVR* dapat dijalankan pada sistem operasi *Windows 95, 98, Me, NT4, 2000, dan XP*. *Cross-compiler C* mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada sistem *embedded*. *File object COFF* hasil kompilasi dapat digunakan untuk keperluan *debugging* pada tingkatan C, dengan pengamatan variabel, menggunakan *debugger Atmel AVR Studio*.

CodeVision AVR merupakan yang terbaik bila dibandingkan dengan *compiler-compiler* yang lain karena beberapa kelebihan yang dimiliki oleh CodeVision AVR antara lain :

1. Menggunakan IDE (*Intergrated Development Environment*).
2. Fasilitas yang disediakan lengkap (mengedit program, meng*compile* program, men-*download* program) serta tampilannya yang terlihat menarik dan mudah dimengerti. Setingan editor dapat diatur sedemikian rupa sehingga membantu memudahkan dalam penulisan program.
3. Mampu membangkitkan kode program secara otomatis dengan menggunakan fasilitas *CodeWizard* AVR.
4. Memiliki fasilitas untuk men-*download* program langsung dari CodeVision AVR dengan menggunakan *hardware* khusus seperti Atmel STK500, Sistem STK 200+/300 dan beberapa *hardware* lain yang telah didefinisikan oleh CodeVision AVR.
5. Memiliki fasilitas *debugger* sehingga dapat menggunakan *software compiler* lain untuk mengecek kode *assembler*-nya, contohnya AVRStudio.
6. Memiliki terminal komunikasi serial yang terintegrasi dalam CodeVision AVR sehingga dapat digunakan untuk membantu pengecekan program yang telah dibuat khususnya yang menggunakan fasilitas komunikasi serial UART.

Selain itu, CodeVision AVR juga menyediakan sebuah *tool* yang dinamakan dengan *Code Generator* atau CodeWizard AVR. CodeWizard AVR merupakan salah satu fasilitas yang disediakan oleh CodeVision AVR yang dapat digunakan untuk

mempercepat penulisan listing program. Dengan *CodeWizard AVR* secara otomatis akan dibuatkan kerangka program melalui menu – menu yang disediakan.



Gambar 2.9 Tampilan CodeVision AVR (Dellcomp, 2011)

Fasilitas ini sangat membantu terutama apabila pengguna lupa dengan nama *register* yang akan digunakan untuk mengatur mode kerja fitur – fitur yang ada dalam mikrokontroler. Jadi fasilitas ini akan mudah dimengerti oleh *user*, minimal sudah pernah mempelajari *register – register* kontrol dalam mikrokontroler ATmega8535. Atau dengan kata lain, fasilitas ini hanya digunakan untuk membantu mempercepat penulisan program serta mengingat kembali bagaimana penggunaan *register – register*.

II.6 Display (Indikator)

LCD (*Liquid Crystal Display*) adalah modul penampil yang banyak digunakan karena tampilannya menarik. LCD yang paling banyak digunakan saat ini

ialah LCD M1632 *refurbish* karena harganya cukup murah. LCD M1632 merupakan modul LCD dengan tampilan 2x16 (2 baris x 16 kolom) dengan konsumsi daya rendah. Modul tersebut dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD.

Mikrokontroler HD44780 buatan Hitachi yang berfungsi sebagai pengendali LCD memiliki CGROM (*Character Generator Read Only Memory*), CGRAM (*Character Generator Random Access Memory*), dan DDRAM (*Display Data Random Access Memory*). LCD yang umum, ada yang panjangnya hingga 40 karakter (2x40 dan 4x40), dimana digunakan DDRAM untuk mengatur tempat penyimpanan karakter tersebut.

Display	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16						
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	...
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	...

Gambar 2.10 Susunan Alamat pada LCD (Zeppelinstraße, 2008)

Fungsi dari pin pada LCD :

1. Pin 1 : VSS/GND (Tegangan 0 volt atau *ground*)
2. Pin 2 : VCC (Tegangan VCC)
3. Pin 3 : VEE/*Vcontrast* (Tegangan pengatur kontras pada LCD)
4. Pin 4 : RS (*Register Select*) ; “0” = Instruksi dan “1” = Data
5. Pin 5 : R/W (Signal yang digunakan untuk memilih mode membaca atau menulis) ; “0” = Menulis dan “1” = Membaca
6. Pin 6 : E (*Enable*) (Untuk mulai pengiriman data atau instruksi)

7. Pin 7 : Sampai dengan Pin 14 ; DB 0 s/d DB 7 (Untuk mengirimkan data karakter)
8. Pin 15 dan Pin 16 : Untuk mengatur cahaya pada *background* LCD atau instruksi.



Gambar 2.11 LCD (*Liquid Crystal Display*) (Anonim^b,2010)

Alamat awal di baris kedua dimulai dari 40H. Untuk meletakkan suatu karakter pada baris ke-2 kolom pertama, maka harus di-set pada alamat 40H. Jadi, meskipun LCD yang digunakan 2x16 atau 2x24, atau bahkan 2x40, maka penulisan programnya sama saja.

CGRAM merupakan memori untuk menggambarkan pola sebuah karakter, dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan. Namun, memori akan hilang saat *power supply* tidak aktif sehingga pola karakter akan hilang. (Widodo Budiharto & Gamayel Rizal 2007, P.51-52).

II.7 Android

Android adalah sistem 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, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan 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 sistem 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)*.

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010). Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*mobile*) yang merupakan modifikasi *kernel Linux 2.6*. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan *debug* dan penambahan fitur baru.



Gambar 2.12 Tampilan Android versi 4.2 Jelly Bean (Android Inc.)

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android. Berikut adalah beberapa versi dari Android :

- **Android versi 1.1**

I.1.1 Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan [Gmail](#), dan pemberitahuan [e-mail](#).

- **Android versi 1.5 (Cupcake)**

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). 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.

- **Android versi 1.6 (Donut)**

Donut (versi 1.6) dirilis pada bulan September 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.

- **Android versi 2.0/2.1 (Eclair)**

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), 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.

- **Android versi 2.2 (Froyo: Frozen Yoghurt)**

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan *auto update* dalam aplikasi *Android Market*.

- **Android versi 2.3 (Gingerbread)**

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. 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.

- **Android versi 3.0/3.1 (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. Perangkat tablet dengan *platform* Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus.

- **Android versi 4.0 (ICS: Ice Cream Sandwich)**

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb 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. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

- **Android versi 4.1 (Jelly Bean)**

Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan *input keyboard*, desain baru fitur pencarian, UI yang baru dan pencarian melalui *Voice Search* yang lebih cepat. Tak ketinggalan Google Now juga menjadi bagian yang diperbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7.

- **Android versi 4.2 (Jelly Bean)**

Fitur *photo sphere* untuk panorama, *daydream* sebagai *screensaver*, *power control*, *lock screen widget*, menjalankan banyak *user* (dalam tablet saja), *widget* terbaru. Android 4.2 Pertama kali dikenalkan melalui LG Google Nexus 4

II.8 Eclipse Juno

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

- *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

- *Multit-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*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh *user* dengan komponen yang dinamakan *plug-in*.

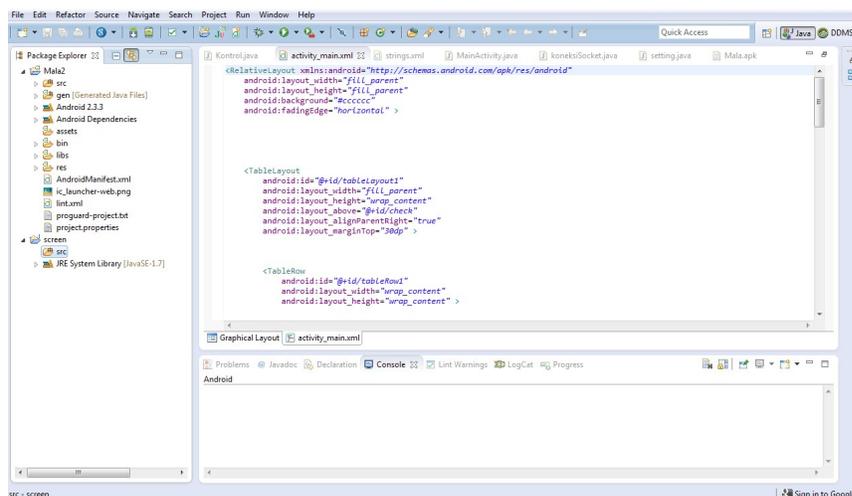
Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah *kernel*, yang mengangkat *plug-in*. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Berikut ini adalah komponen yang membentuk RCP:

- *Core platform*
- OSGi
- SWT (*Standard Widget Toolkit*)
- JFace

- *Eclipse Workbench*

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.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya. Salah satu situs yang menawarkan *plug-in* secara gratis seperti Eclipse download by project.

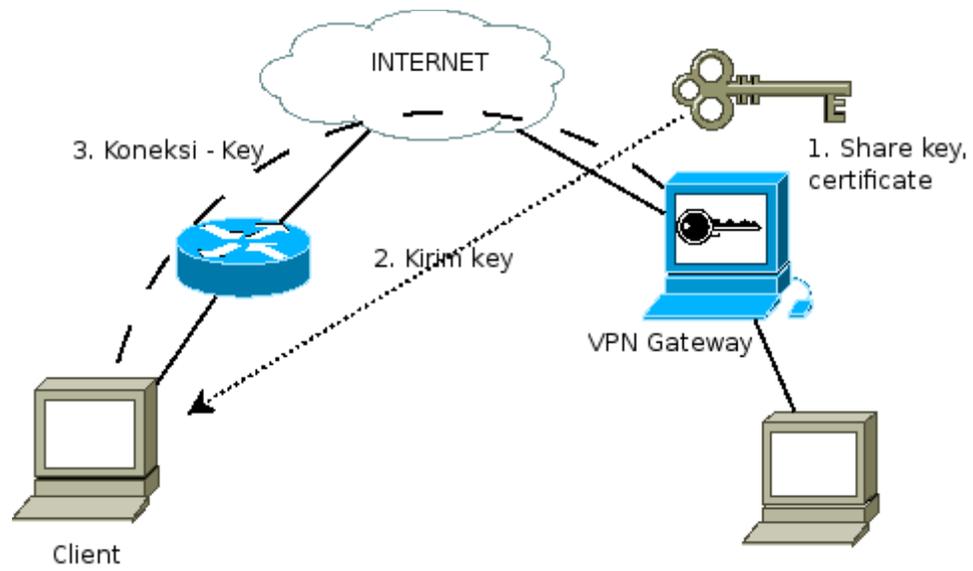


Gambar 2.13 Tampilan Eclipse Juno (Oracle, 2011).

II.9 OpenVPN

OpenVPN adalah sebuah aplikasi perangkat lunak *open source* yang mengimplementasikan *Virtual Private Network* (VPN). OpenVPN ini menggunakan teknik *point-to-point* dalam konfigurasi koneksi jembatan ataupun fasilitas akses remot. Aplikasi ini menggunakan sistem modifikasi IP sebagai sistem keamanannya. Sistem ini mampu menembus jaringan internet dan sistem *firewall*. Open VPN memungkinkan untuk saling berinteraksi menggunakan kode rahasia, sertifikasi ataupun *username dan password*. Ketika digunakan dalam konfigurasi multi *client-server*, memungkinkan server untuk melepaskan otentifikasi untuk setiap klien menggunakan *username ataupun password*.

OpenVPN menggunakan OpenSSL (open Secure Socket Layers). OpenSSL berfungsi untuk menyediakan enkripsi dari kedua data dan saluran kontrol. Ini memungkinkan OpenSSL melakukan semua enkripsi dan otentifikasi kerja, sehingga OpenVPN dapat menggunakan semua jaringan yang tersedia dalam paket OpenSSL. Hal ini juga dapat menggunakan fitur paket otentikasi HMAC untuk menambahkan lapisan tambahan keamanan untuk koneksi (disebut sebagai "*Firewall HMAC*" oleh sang pencipta). Hal ini juga dapat menggunakan akselerasi hardware untuk mendapatkan kinerja yang lebih baik.



Gambar 2.14 Sistem Kerja OpenVPN (Dhoto, 2010)

OpenVPN memiliki beberapa cara untuk mengotentikasi rekan-rekan satu sama lain. OpenVPN menawarkan sistem berbagi kunci, berbasis sertifikat, dan *username / password-based authentication*. Kunci rahasia sistem berbagi adalah yang paling mudah, dengan sertifikat berdasarkan yang paling kuat dan kaya fitur. Dalam versi *username / password* otentikasi 2.0 dapat diaktifkan, baik dengan atau tanpa sertifikat. Namun untuk menggunakan *username / password* otentikasi, OpenVPN tergantung pada modul pihak ketiga.

Keuntungan yang dapat diperoleh dengan menggunakan service VPN :

1. Akses global ke jaringan server/client melalui internet.
2. Kemudahan mengakses resources yang berada di dalam jaringan server/client yang tidak dapat diakses dari mana saja secara langsung.
3. Keamanan dalam transfer data karena adanya enkripsi data.

4. Keamanan jaringan sebab hanya user yang telah melewati proses autentikasi dan otorisasi saja yang dapat menggunakan service VPN.

BAB III

METODE PENELITIAN

III.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Jurusan Elektro Fakultas Teknik Universitas Hasanuddin, mulai pada bulan April 2012 hingga Januari 2013.

III.2 Perancangan

Perancangan sistem meliputi perancangan perangkat keras dan perangkat lunak.

Perancangan perangkat keras meliputi :

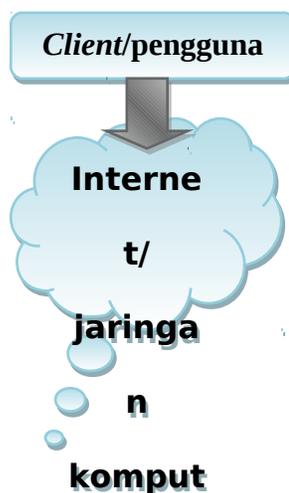
- A. Perancangan *driver* relay
- B. Perancangan *power supply*

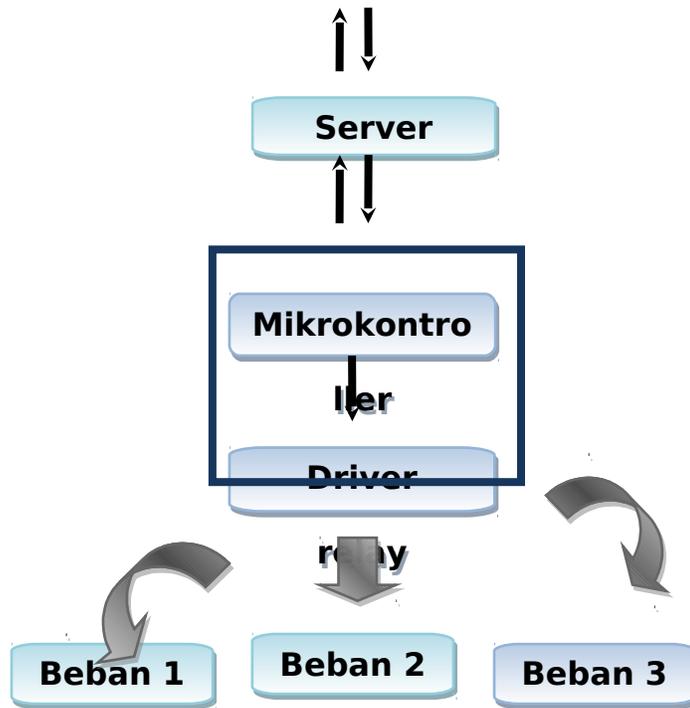
C. Perancangan rangkaian mikrokontroller

Perancangan perangkat lunak meliputi :

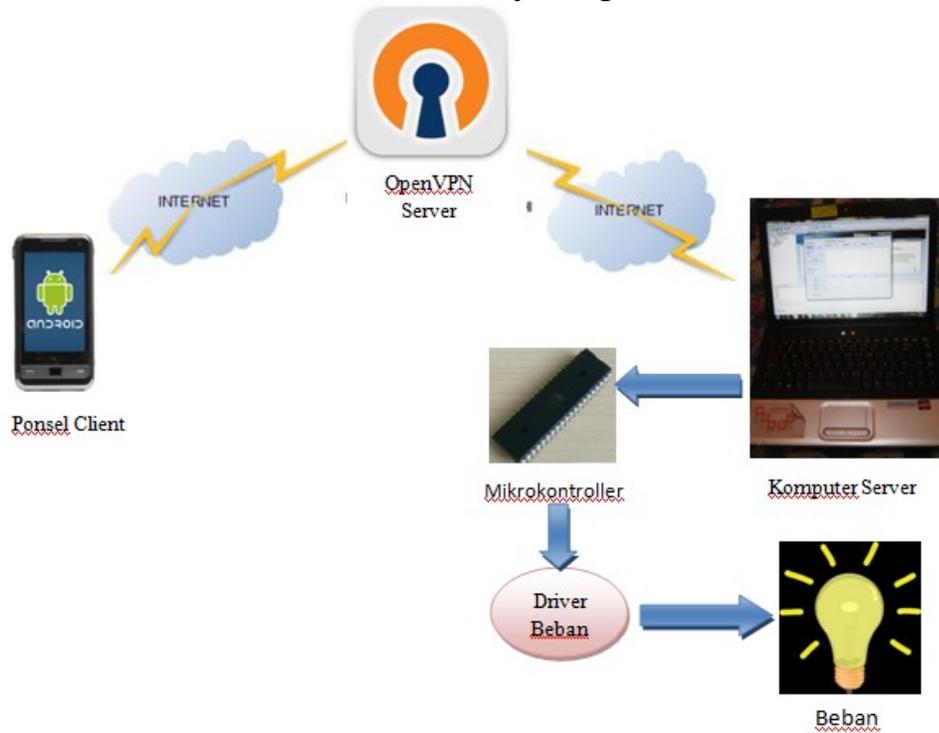
- A. Pembuatan program *desktop*
- B. Pembuatan program aplikasi seluler
- C. Pembuatan *flowchart* dan *listing* program
- D. *Compile* dan *load* program ke dalam mikrokontroller ATmega 16

Secara garis besar, prinsip kerja dari sistem ini menggunakan metode *switching*, dan yang menjadi asal mula pengendalian pada sistem ini berada pada panel kendali yang terdapat pada aplikasi *desktop* dan aplikasi android. Kedua aplikasi ini masing-masing disimpan pada perangkat yang berbeda. Aplikasi *desktop* yang dibuat menggunakan Bahasa Pemrograman NETBeans IDE 7.2 berfungsi sebagai aplikasi untuk komputer *server* yang terhubung langsung dengan perangkat keras *switching*. Sedangkan aplikasi seluler berbasis android dibuat menggunakan Bahasa Pemrograman Eclipse Juno yang berfungsi sebagai aplikasi untuk *client* / pengguna yang di instalkan langsung pada handphone/tablet dengan OS Android versi 2.3 atau sesuai dengan *update* dari Android SDK yang dimiliki. Untuk lebih jelasnya cara kerja sistem dari penelitian ini dapat dilihat pada gambar 3.1.





Gambar 3.1 Alur Kerja Rangkaian



Gambar 3.2 Diagram Kerja Rangkaian

Sistem pengontrolan peralatan rumah tangga via telepon seluler merupakan sebuah sistem yang dirancang untuk mengendalikan peralatan listrik rumah tangga menggunakan aplikasi *desktop* sebagai pengirim isyarat kendalian pada keseluruhan sistem. Aplikasi *desktop* untuk komputer *server* ini diawali dengan sebuah halaman aplikasi yang menampilkan halaman LOG-IN. Halaman Log-IN ini berfungsi sebagai sistem keamanan awal pada komputer *server* administrator. Halaman selanjutnya berisi halaman awal dari aplikasi *desktop* ini yang berisi panel “kontrol” yang berfungsi sebagai panel untuk masuk ke halaman pengendalian. Panel lainnya adalah panel “*manage user*”, panel ini berfungsi sebagai panel kontrol untuk *administrator* pada saat ada penambahan ID *client*.

Setiap piranti yang terhubung dengan alat pengendali sebelumnya telah diprogramkan di komputer *server* dimana komputer *server* ini terhubung langsung dengan aplikasi android yang ter-install dalam telepon seluler ataupun perangkat lain yang berbasis android. Selain itu, pada mikrokontroller juga terdapat beberapa perintah yang mewakili keadaan setiap piranti yang akan dikendalikan. Setiap piranti ini diwakili oleh masing-masing relay yang terhubung elektrik dengan mikrokontroller.

Setiap relay yang akan dihidupkan telah diprogramkan sebelumnya pada mikrokontroller menggunakan karakter *string* untuk dapat memberikan perintah menyala atau mematikan piranti listrik tersebut, dan karakter *string* tersebut dikirimkan dari halaman aplikasi *server* yang digunakan. Setiap pengiriman karakter yang berbeda dari aplikasi *server* ke mikrokontroller mengirimkan umpan balik pada

aplikasi *server* berupa status relay yang sedang aktif sehingga peralatan listrik tersebut dapat dikontrol sesuai kebutuhan penggunaanya.

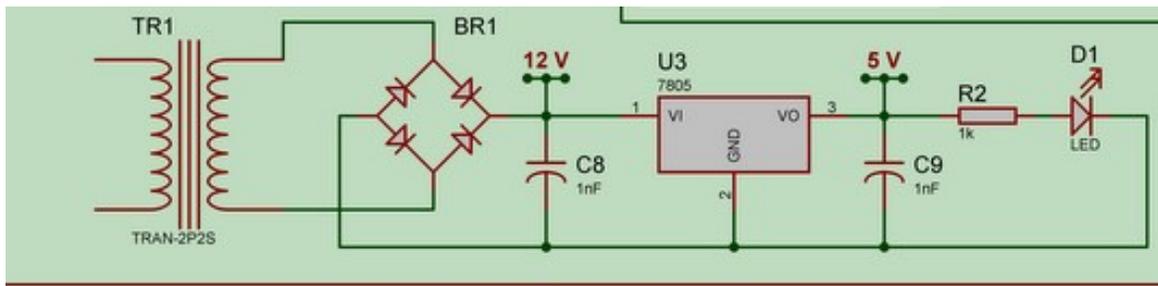
Karakter *string* yang dikirimkan dari *server* selanjutnya akan diproses oleh mikrokontroller untuk diolah dan dieksekusi. Selanjutnya, keluaran dari mikrokontroller terhubung langsung dengan peralatan listrik rumah tangga. Relay terhubung dengan pin-pin keluaran mikrokontroller dan peralatan listrik yang berfungsi sebagai pengaman dan saklar otomatis yang penggunaannya diatur oleh mikrokontroller dengan tegangan $220V_{AC}$. Hasil pembacaan ATMEGA 16 yang akan menggerakkan relay yang mengakibatkan mati/menyalanya beban. Sehingga relay memiliki fungsi sebagai saklar beban.

Gambar 3.3 Skematik Apartemen untuk perancangan sistem control

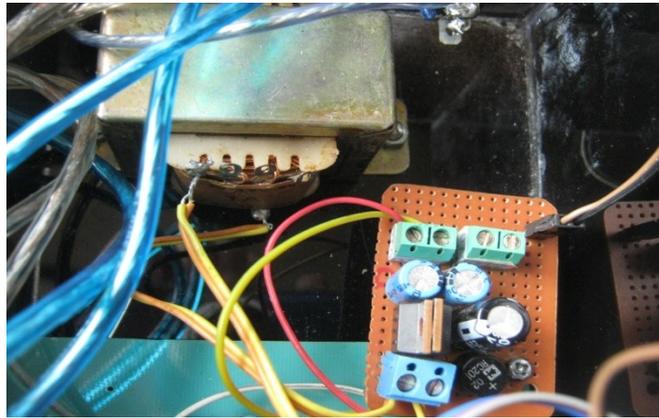
III.3 Perancangan Perangkat Keras

III.3.1 Perancangan Rangkaian *Power supply*

Power supply dirancang untuk menghasilkan *supply* tegangan DC, untuk rangkaian mikrokontroler dibutuhkan tegangan $5V_{DC}$ dan untuk rangkaian *driver relay* dibutuhkan $12V_{DC}$. Sebagai penyearah, rangkaian ini menggunakan sistem rangkaian penyearah jembatan seperti yang terlihat pada gambar 3.4 Berikut adalah gambar rangkaian skematik.



Gambar 3.4 Rangkaian Skematik *Power supply*



Gambar 3.5 Rangkaian *Power supply*

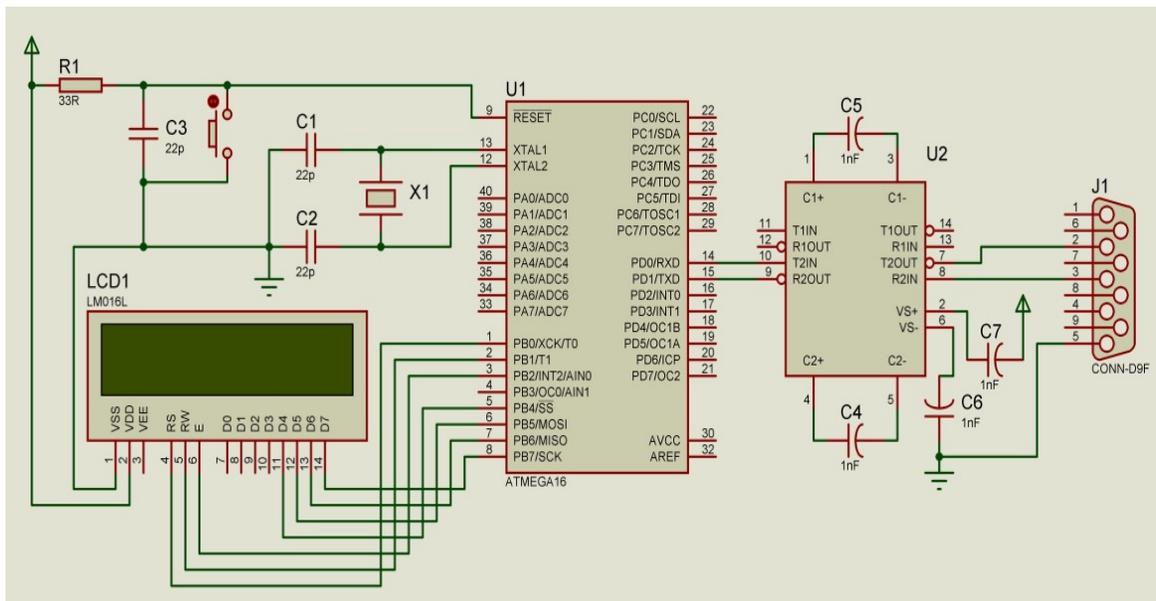
Pada rangkaian *power supply* dilengkapi dengan dua buah kapasitor yang berfungsi sebagai pengaman dan menghasilkan arus DC dengan riak yang lebih kecil untuk menghasilkan tegangan rata-rata yang lebih baik. Selain itu, rangkaian juga dilengkapi dengan dua buah regulator LM7805 yang berfungsi menghasilkan tegangan keluaran $5V_{DC}$. Disamping itu regulator LM7805 juga berfungsi menjaga kestabilan tegangan keluaran dari trafo agar sistem dapat terus bekerja dengan baik.

Adapun komponen-komponen yang digunakan pada rangkaian *power supply* pada sistem pengendali ini adalah :

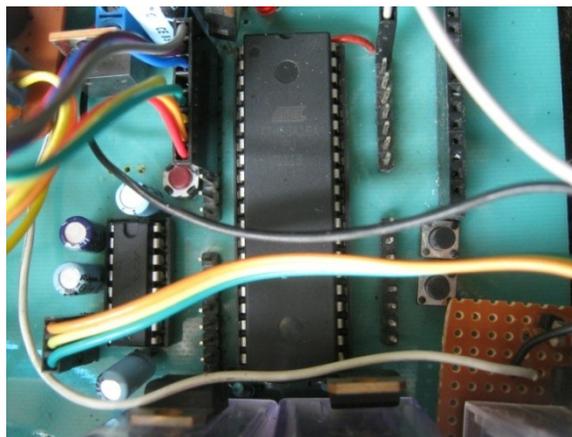
- Trafo $220V_{AC}$
- Dioda RC201
- LM 7805
- Kapasitor $330\mu F$
- Kapasitor $1000\mu F$

III.3.2 Rangkaian Mikrokontroller ATmega 16

Rangkaian dari mikrokontroller ATmega 16 dapat dilihat pada gambar 3.6. Rangkaian mikrokontroller ini memiliki sistem *reset* dan *clock*. Crystal 11,0592MHz digunakan sebagai *clock* pada minimum sistem *power supply*. Untuk *supply* rangkaian digunakan adaptor 5Volt.



Gambar 3.6 Rangkaian Skematik Mikrokontroller dan LCD



Gambar 3.7 Rangkaian Mikrokontroller dan LCD

Komputer *server* dihubungkan dengan mikrokontroler menggunakan *USB-to-serial* yang kemudian mengubah data yang dikirimkan komputer *server* dari *string* menjadi logika “1” atau “0”. Data yang dikirimkan ini juga menjadi masukan bagi mikrokontroler ATmega 16 sebagai pemicu kerja dari mikrokontroler. Logika “1” atau “0” yang dihasilkan dari *port* keluaran ATmega 16 bergantung pada perintah pengguna (*client*). Logika “1” yang dikirimkan ke mikrokontroler akan mengaktifkan relay, sehingga peralatan listrik yang terhubung akan mengalami kondisi ON. Sebaliknya, saat logika “0” dikirimkan pada mikro, akan menonaktifkan relay, sehingga peralatan listrik yang terhubung akan mengalami kondisi OFF.

Adapun komponen-komponen yang digunakan pada rangkaian minimum sistem rangkaian mikrokontroler ini adalah :

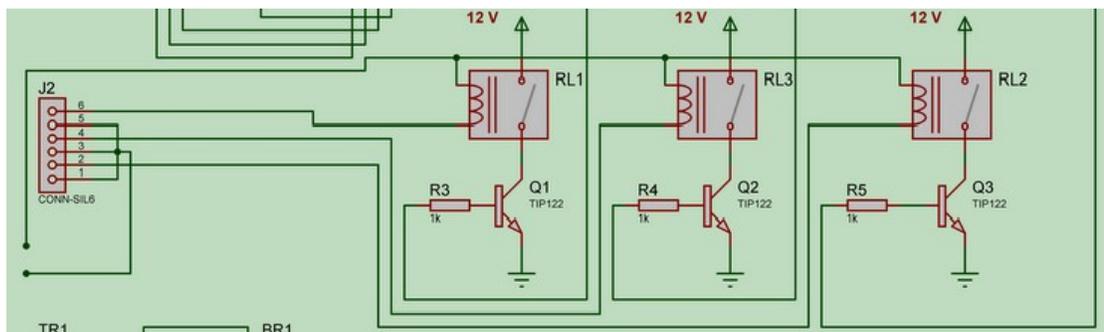
- Mikrokontroler ATmega 16
- Kristal 11,0592MHz
- Kapasitor 22 μ F
- Resistor
- Max 232
- Serial Port
- *Push button*

III.3.3 Rangkaian *Driver Relay*

Relay berfungsi sebagai saklar otomatis yang digunakan untuk memutus dan menyambungkan *supply* tegangan 220V_{AC} dari rangkaian *power supply* ke beban peralatan listrik. Relay juga banyak digunakan dalam rangkaian pensaklaran dan perangkat penguji otomatis. Relay berisi suatu kumparan yang apabila dialiri arus searah akan membangkitkan medan magnet yang akan menyambungkan atau memutuskan satu atau lebih kontak mekanis.

Pensaklaran *supply* beban dilakukan oleh relay yang dikendalikan oleh mikrokontroller. Rangkaian ini dirancang sesuai dengan rancangan program yang terdapat pada mikrokontroller, dimana terdapat sinyal kendali dari mikrokontroller. Bila sinyal (± 5 volt) yang berlogika “1” dikirimkan dari *antarmuka* ke mikrokontroller, maka beban yang dikendalikan akan terhubung dengan *line AC*, dan sebaliknya apabila sinyal (± 0 volt) yang berlogika “0” dikirimkan dari *antarmuka* ke mikrokontroller, maka beban yang dikendalikan akan terputus dengan *line AC*.

Pada perancangan pengendali jarak jauh peralatan listrik rumah tangga kali ini digunakan tiga buah relay yang masing-masing memiliki tugas untuk mengontrol dan mengatur penggunaan peralatan listrik yang nantinya dihubungkan dengan modul *antarmuka*. Setiap relay memiliki fungsi ON dan OFF. Masing-masing relay juga terhubung dengan TIP yang akan menghasilkan arus yang melewati kumparan relay dan akan memicu relay.



Gambar 3.8 Rangkaian Skematik *Driver Relay*



Gambar 3.9 Rangkaian *Driver Relay*

Komponen-komponen yang digunakan pada rangkaian *driver relay* untuk dapat men-*drive* beban $220V_{AC}$ adalah :

- Resistor
- Relay
- TIP 122

III.4 Perancangan Perangkat Lunak

Perangkat lunak adalah program yang berisi kumpulan instruksi untuk melakukan proses pengolahan data. *Software* sebagai penghubung antara manusia

(*client*) dengan perangkat keras computer (*server*), berfungsi menerjemahkan bahasa manusia ke dalam bahasa mesin sehingga perangkat keras dapat melaksanakan kerja sesuai dengan instruksi ataupun keinginan dari pengguna dan menjalankan instruksi yang diberikan yang selanjutnya menghasilkan perintah yang diinginkan oleh pengguna. Pada penelitian ini, terdapat sebuah metode yang digunakan untuk mengendalikan peralatan listrik (ON/OFF) sesuai dengan keinginan pengguna. Pengendalian dilakukan dengan cara mengirimkan isyarat kendali, dimana *client* mengirimkan karakter *string* dari aplikasi android dimana karakter *string* tersebut mewakili perintah untuk menyalakan ataupun mematikan piranti listrik yang akan dikendalikan.

Tabel 3.1 Karakter String Mikrokontroler

ISYARAT KENDALI	
KARAKTER STRING	FUNGSI STRING
1	Menyalakan Saklar 1
2	Memadamkan Saklar 1
3	Menyalakan Saklar 2
4	Memadamkan Saklar 2
5	Menyalakan Saklar 3
6	Memadamkan Saklar 3
7	Menyalakan Saklar 1,2 & 3
8	Memadamkan Saklar 1,2 & 3

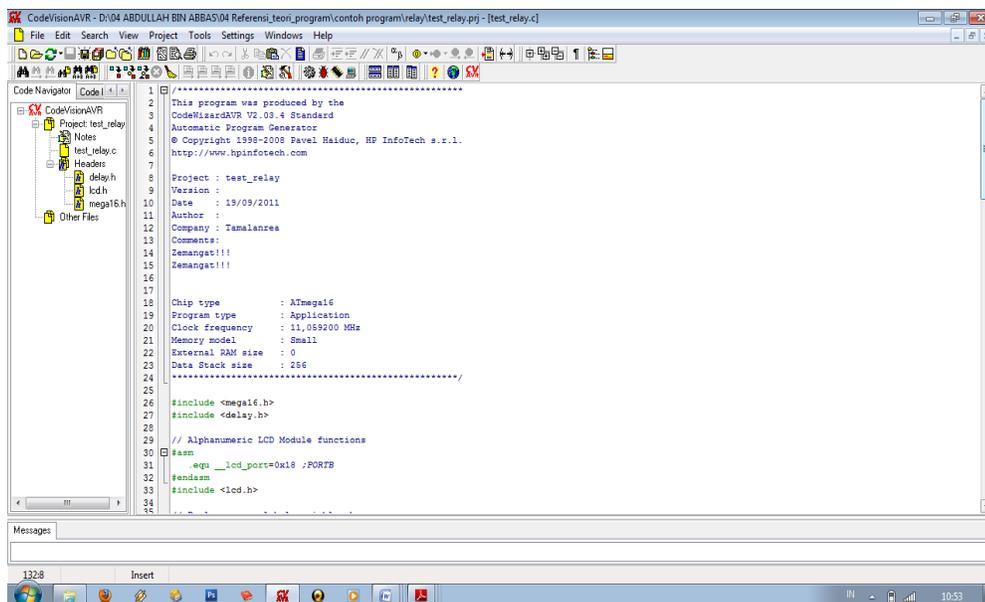
Karakter *string* yang dikirimkan dari aplikasi android adalah sama dengan karakter *string* yang digunakan pada aplikasi server sehingga keduanya bisa mengendalikan kerja dari peralatan listrik.

III.4.1 Perancangan Perangkat Lunak Mikrokontroler



Pembuatan perangkat lunak mikrokontroler ini dibuat berdasarkan pada pengendali utamanya yaitu ATmega 16. Perangkat lunak ini dibuat menggunakan bahasa pemrograman Codevision AVR. Bahasa pemrograman ini merupakan kumpulan baris-baris perintah register yang secara otomatis ditampilkan saat mengetikkan langkah-langkah program. File program yang telah dibuat kemudian di *compile* lalu di *upload* ke mikrokontroler menggunakan *usb downloader*. File-file yang *ter-upload* inilah yang mengatur kerja dari mikrokontroler sebagai pengendali utama sistem.

Perancangan perangkat lunak mikrokontroler ini bertujuan mengenali dan mengolah data yang dikirim oleh komputer *server* yang selalu terhubung dengan mikrokontroler melalui kabel *usb-to-serial*. Setiap data yang dikirimkan ke mikrokontroler memiliki fungsi yang berbeda-beda bergantung dari penggunaannya. Setiap karakter *string* yang dikirim mewakili satu perintah khusus untuk setiap relay yang terhubung.

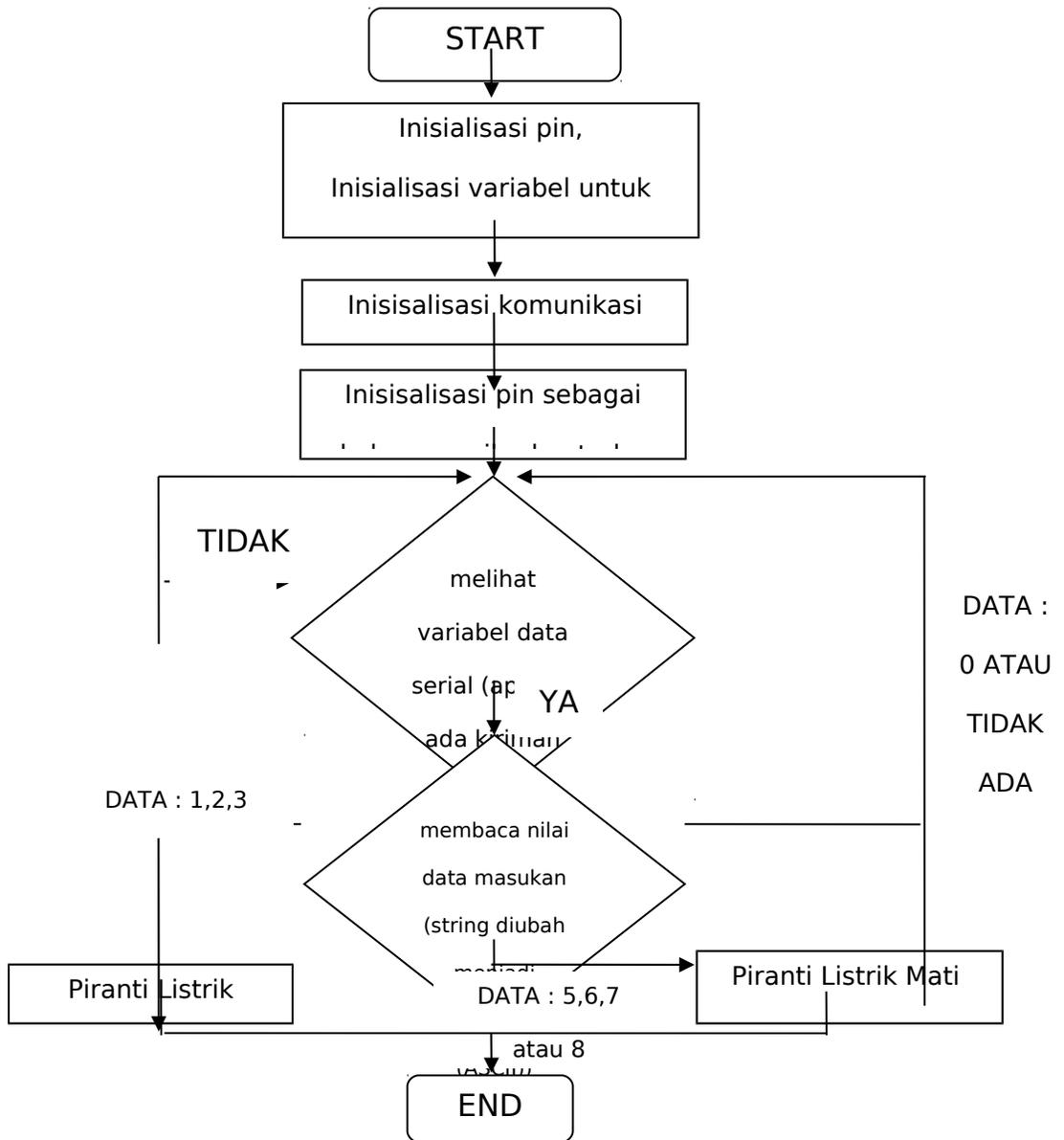


```
CodeVisionAVR - D:\04 ABDULLAH BIN ABBAS\04 Referensi_teori_program\contoh program\relay\test_relay.c - [test_relay.c]
File Edit Search View Project Tools Settings Windows Help
.....
1 This program was produced by the
2 CodeWizardAVR V2.03.4 Standard
3 Automatic Program Generator
4 @ Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.
5 http://www.hpinfootech.com
6
7 Project : test_relay
8 Version :
9 Date : 19/09/2011
10 Author :
11 Company : Tamalanrea
12 Comment:
13 Zemangat!!!
14 Zemangat!!!
15
16
17 Chip type : ATmega16
18 Program type : Application
19 Clock frequency : 11,059200 MHz
20 Memory model : Small
21 External RAM size : 0
22 Data Stack size : 256
23
24 .....
```

```
25
26 #include <mega16.h>
27 #include <delay.h>
28
29 // Alphanumeric LCD Module functions
30 #asm
31 .equ __lcd_port=0x18 ;PORTB
32 #endasm
33 #include <lcd.h>
34
```

Gambar 3.10 Tampilan CodeVision AVR 2.1

Karakter *string* yang digunakan dalam mikrokontroller ini ada delapan *string* mulai dari angka 1 sampai 8 dimana setiap karakter ini nantinya akan di *compile* lagi didalam perangkat lunak *server* yang dikenal dalam *decimal*.



Gambar 3.11 Flowchart Program mikrokontroller ATmega 16

Angka-angka ini kemudian bertindak sebagai perintah bagi mikrokontroller yang terbagi menjadi dua yaitu untuk memadamkan peralatan listrik atau menyalakan peralatan listrik.

Dari *flowchart* pada gambar 3.11 dapat dijelaskan bahwa program mikrokontroller memulai prosesnya lalu dilanjutkan dengan melakukan inisiasi pin mikrokontroller, variable serta komunikasi serial. Pada saat mikrokontroller telah menerima kiriman data dari *server* dimana data tersebut berupa karakter *string* yang dikirim dari *server*, lalu diterjemahkan oleh mikrokontroller sebagai sebuah instruksi. Instruksi ini kemudian terbagi atas dua kondisi yaitu instruksi untuk menyalakan peralatan (*high*) dan instruksi mematikan peralatan listrik (*low*) yang terhubung pada *port/pin output* mikrokontroller tersebut.

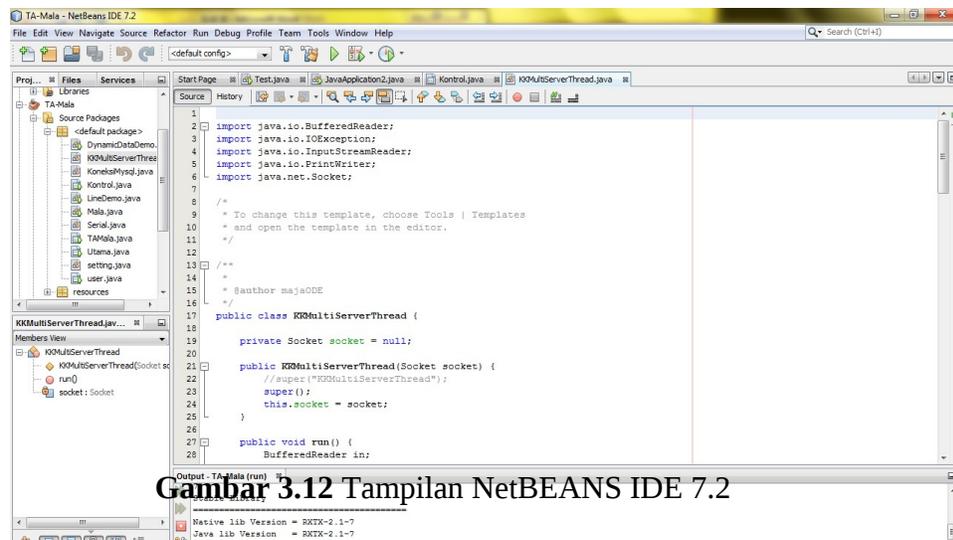
III.4.2 Perancangan Perangkat Lunak Server

Rancangan aplikasi pengontrolan pada penelitian ini berupa aplikasi yang digunakan untuk mengendalikan peralatan listrik. Aplikasi ini harus bisa melakukan komunikasi dengan mikrokontroller untuk mengatur kerja dari perangkat listrik. Selain itu, aplikasi ini juga harus bisa melakukan komunikasi dengan aplikasi seluler (android) baik melalui jaringan internet baik melalui sistem Wi-Fi ataupun sistem jaringan dari provider internet.

Aplikasi pengontrolan berinteraksi dengan mikrokontroller melalui komunikasi serial. Aplikasi ini mengenali mikrokontroller yang dikontrol berdasarkan karakter *string* yang disesuaikan dengan karakter yang terdapat di program mikrokontroller. Komunikasi antara aplikasi pengontrolan dan mikrokontroller adalah komunikasi dua arah. Aplikasi ini mengirimkan karakter

string untuk menyalakan/memadamkan dan mikrokontroller mengirimkan kembali status balikan saat relay bekerja.

Komunikasi antara aplikasi pengontrolan dengan aplikasi seluler (android) dapat menggunakan wi-fi ataupun jaringan internet lainnya. Sistem komunikasi real-time adalah sistem komunikasi yang digunakan untuk menghubungkan antara *server* dengan *client*. Untuk melakukan komunikasi antara *server* dengan *client*, kedua aplikasi ini harus terhubung dengan jaringan internet.



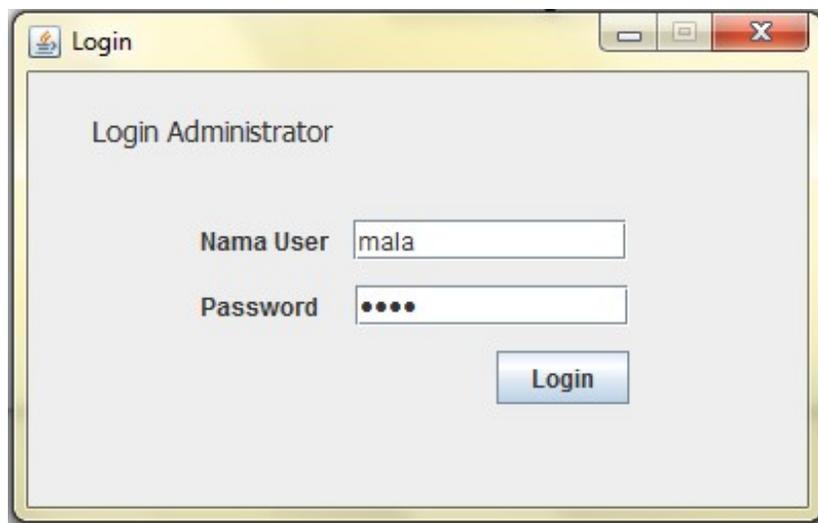
Gambar 3.12 Tampilan NetBEANS IDE 7.2

Bahasa pemrograman yang digunakan dalam pembuatan aplikasi perangkat

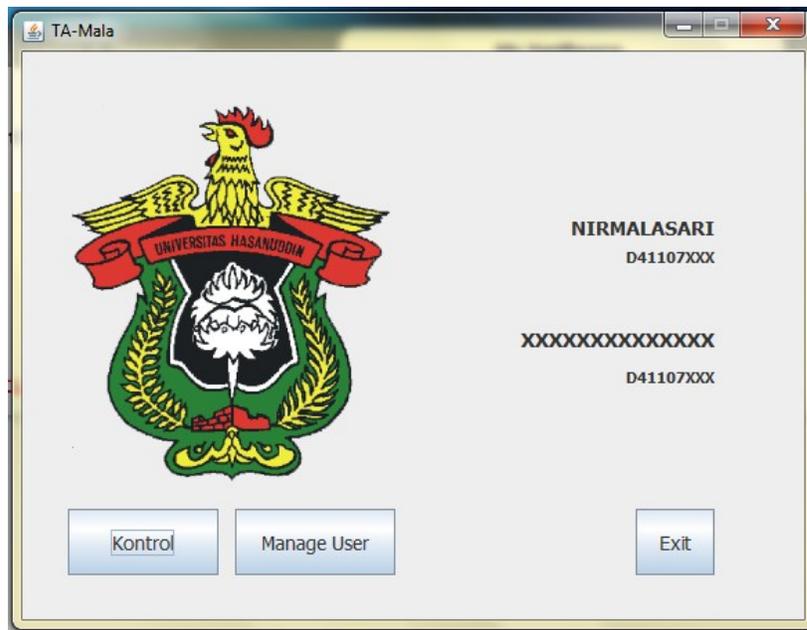
lunak *server* ini adalah NETBeans IDE 7.2. langkah-langkah pembuatannya adalah :

1. Perancangan tampilan
2. Penulisan kode program
3. Pengujian kode program
4. Pengujian dan pengecekan kesalahan
5. *Compile program*

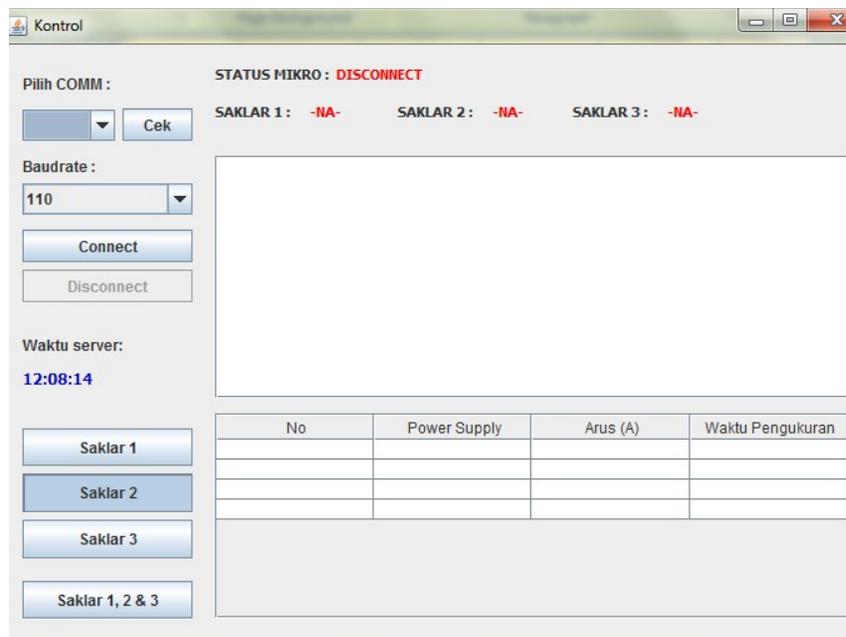
Pada program aplikasi *server*, terdapat beberapa tahapan sebelum memasuki halaman pengontrolan *server*. Halaman pertama adalah halaman LOG-IN, dimana halaman ini adalah halaman pengisian *user name* dan *password*. Halaman ini juga sebagai halaman pengaman pertama dalam program ini. *User name* dan *password* untuk *server* telah diatur sebelumnya di dalam script program NETBeans. Setelah menekan tombol Log-IN, yang muncul selanjutnya adalah halaman pengaturan *server* yang mana pada halaman tersebut, server dapat melakukan penambahan *user name* dan *password* untuk *client*. Selain itu, terdapat tombol untuk langsung masuk ke panel pengontrolan. Berikut adalah halaman tampilan untuk *server* :



Gambar 3.13 Halaman Log-IN *server*



Gambar 3.14 Halaman Pengaturan server



Gambar 3.15 Halaman Pengontrolan server

III.4.3 Perancangan Aplikasi Android / Client

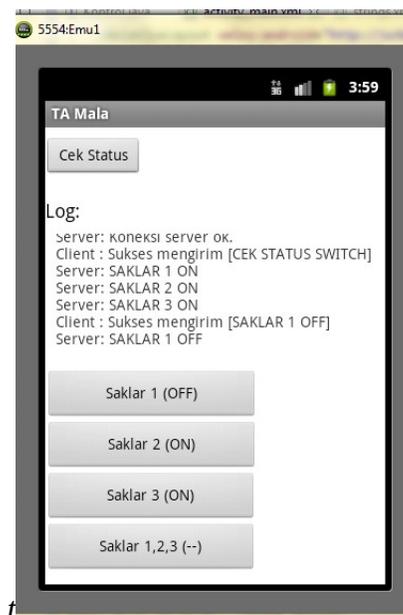
Apliasi seluler merupakan aplikasi utama dari sistem pengendali ini. Aplikasi ini berhubungan langsung dengan pengguna dan dapat terus di *update* oleh pengguna kapan dan dimana saja pengguna berada. Perangkat listrik juga dapat dikendalikan dari aplikasi pengontrolan (*server*/kendali utama). Komunikasi antara aplikasi seluler dengan aplikasi pengontrolan menggunakan media internet.

Aplikasi ini adalah aplikasi *client* yang terhubung langsung dengan internet. Settingan standard yang dibutuhkan adalah *setting*-an yang sama untuk penggunaan semua aplikasi *browsing* untuk internet. Akses dapat dilakukan menggunakan Wi-Fi ataupun internet dari provider telepon seluler. Sama halnya dengan aplikasi *server* sebelumnya, aplikasi *client* ini juga memiliki tingkat keamanan yang cukup dikarenakan sebelum *client* memulai penggunaan dari aplikasi ini, *client* akan diminta memasukkan *user name* dan *password* yang harus didaftar terlebih dahulu pada *server* sehingga tidak sembarang orang bisa mengakses aplikasi ini.

Halaman aplikasi *server* ini terdiri dari beberapa bagian diantaranya, bagian panel pengontrolan dan *update* status terakhir dari alat yang hendak dikontrol. Aplikasi ini sendiri masih dapat ditambah ataupun dikurangi komponen di dalamnya sesuai dengan yang ada pada aplikasi *server*. Untuk pembuatan program ini, digunakan bahasa pemrograman *Eclipse Juno*.



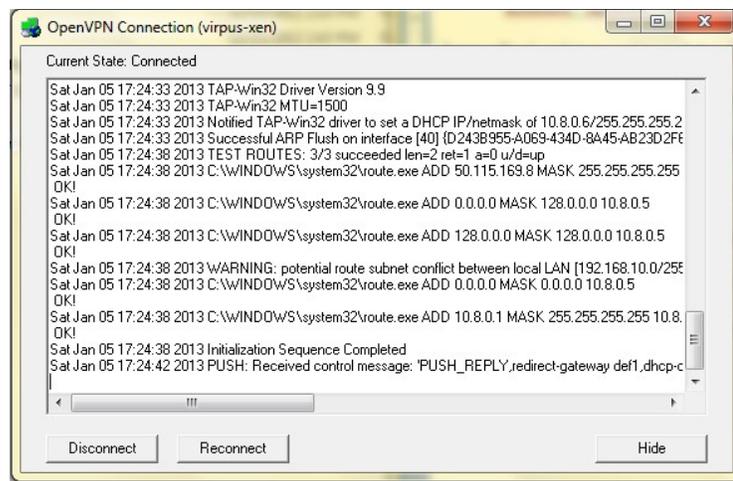
Gambar 3.16 Halaman Log-IN *Clie*n



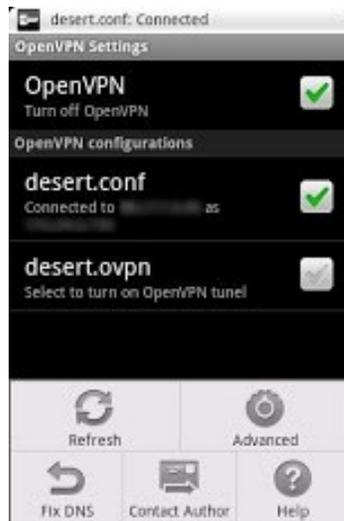
Gambar 3.17 Halaman Pengontrolan Aplikasi Android (*client*)

III.5 Perancangan Koneksi *Server* dengan Aplikasi Android (*Client*)

Koneksi aplikasi android (*client*) dengan *server* menggunakan sistem komunikasi point-to-point melalui OpenVPN untuk membangun komunikasi melalui jaringan internet. Pada komputer *server* terlebih dahulu di install-kan OpenVPN yang sudah dienkripsi dan memiliki konfigurasi tetap yang dapat dijadikan acuan untuk OpenVPN pada aplikasi android (*client*). OpenVPN pada *server* berfungsi sebagai pengatur koneksi utama antara *server* dan *client*. Sedangkan pada ponsel android yang digunakan juga di install-kan aplikasi OpenVPN *Setting* dan *Installer* yang dapat di *download* secara langsung dari Android *Market*. Namun sebelum dilakukan seting, harus dilakukan duplikasi konfigurasi kedalam sistem ponsel yang digunakan.

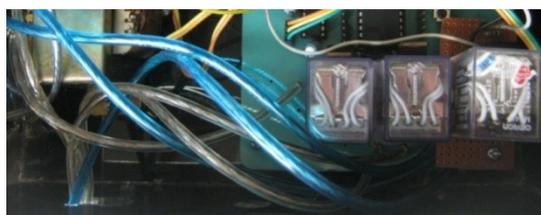
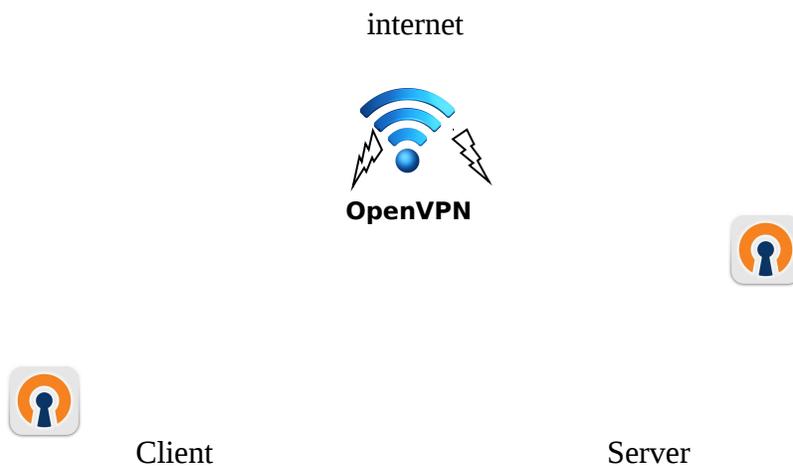


Gambar 3.18 Tampilan OpenVPN GUI (*server*)



Gambar 3.22 Tampilan OpenVPN *for* Adroid

Menghubungkan antara *server* dan *client* pada dasarnya digunakan untuk dapat berkomunikasi dalam melakukan pengontrolan perangkat listrik melalui aplikasi yang telah dibuat. OpenVpn berfungsi sebagai jembatan yang menghubungkan *server* dan *client* dengan jaringan internet. OpenVpn juga berfungsi untuk meminimalisir waktu penundaan yang biasa terjadi saat melakukan komunikasi melalui internet.



Gambar 3.24 Pemasangan Rangkaian



Gambar 3.25 Pemasangan Prototipe dengan Beban

BAB IV

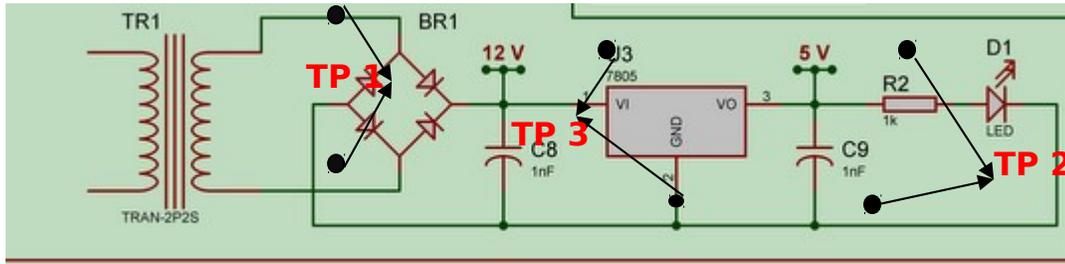
HASIL DAN PEMBAHASAN

Pengujian alat meliputi pengujian perangkat keras dan perangkat lunak sistem. Dimana pengujian perangkat keras meliputi pengujian unjuk kerja perangkat keras terhadap tegangan. Sedangkan pengujian perangkat lunak dibagi atas pengujian fungsi dan tombol-tombol, sedangkan pengujian kinerja berupa pengujian yang menggunakan respon waktu terhadap ON/OFF peralatan listrik.

IV.1 Pengujian Perangkat Keras

IV.1.1 Pengujian *Power supply*

Rangkaian *power supply* digunakan untuk memberi *supply* tegangan bagi mikrokontroller dan *driver* relay. Rangkaian ini harus mensuplai tegangan yang stabil dan mempunyai arus yang cukup untuk mensuplai mikrokontroller sehingga tidak terjadi *drop* tegangan saat mikrokontroller digunakan.



Gambar 4.1 Titik Pengukuran Rangkaian *Power supply*

Rangkaian *power supply* dibuat sedemikian rupa sehingga menghasilkan dua tegangan keluaran yang berbeda yaitu tegangan $12V_{DC}$ untuk mengaktifkan *driver* relay dan tegangan $5V_{DC}$ untuk mengaktifkan mikrokontroller. Pada gambar 4.1 terlihat bahwa terdapat dua titik potong sebagai tempat melakukan pengukuran tegangan keluaran dari *power supply*, dan berikut hasil pengukurannya :

Tabel 4.1 Hasil Pengukuran Rangkain *Power supply*

Titik	Jenis CatuDaya	Tegangan Input (Volt)	Tegangan Output (Volt)
TP 1	Trafo	213 V_{AC}	12,2 V_{AC}
TP 2	Output Regulator 7805	12 V_{DC}	5,09 V_{DC}
TP 3	Tanpa Regulator 7805	12,2 V_{AC}	12,4 V_{DC}

Dari tabel 4.1 didapatkan hasil pengujian di beberapa titik pengukuran. Untuk TP1 dengan trafo sebagai komponen yang diukur tegangan *input* dan *output*, didapatkan $213V_{AC}$ untuk tegangan inputnya dan $12,2V_{AC}$ untuk tegangan output. Sedangkan nilai tegangan yang diharapkan adalah $220V_{AC}$ untuk *input* dan $12V_{AC}$ untuk *output* pada trafo. Untuk TP2 dengan pengukuran dengan regulator 7805, didapatkan $12V_{DC}$ pada tegangan *input* dan $5,09V_{DC}$ pada tegangan *output*. Sedangkan nilai tegangan terukur yang diharapkan adalah tegangan yang sesuai dengan *datasheet* sebesar $12V_{DC}$ untuk tegangan *input* dan $5V_{DC}$ untuk tegangan *output*. Pada TP3 untuk pengukuran tanpa regulator, didapatkan $12,2V_{AC}$ untuk tegangan *input* dan $12,4V_{AC}$ untuk tegangan *output*. Sedangkan nilai tegangan terukur yang diharapkan adalah $12V_{AC}$ untuk tegangan *input* dan $12V_{DC}$ untuk tegangan *output* sesuai dengan *datasheet*. Perbedaan level tegangan yang terjadi pada rangkaian *power supply* ini masih berada pada level normal sesuai dengan yang ditetapkan pada *datasheet*.

IV.1.2 Pengujian Mikrokontroller

Tabel 4.2 diperlihatkan hasil pengukuran tegangan pada *pinout* mikrokontroller pada kondisi *standby* atau tanpa kiriman isyarat kendali (karakter *string*) dari halaman *server*. Tabel pengujian ini hanya menyajikan data terkait dengan *pinout* yang digunakan pada penelitian ini.

Tabel 4.2 Hasil Pengukuran Mikrokontroller Keadaan OFF

<i>Pinout</i>	Keterangan	Tegangan (Volt)
1	PB.0	0
2	PB.1	0
3	PB.2	0

4	PB.3	0
5	PB.4	0
6	PB.5	0
7	PB.6	0
8	PB.7	0
9	RESET	0
10	VCC	0
11	XTAL 2	0
12	XTAL 1	0
13	PD.0 (RX)	0
14	PD.1 (TX)	0
15	PC.3	0
16	PC.4	0
17	PC.5	0

Dari tabel hasil pengukuran 4.2 dapat dilihat bahwa rating tegangan berada pada level normal. Hal ini dikarenakan mikrokontroller bekerja sesuai dengan keadaan yang diharapkan. Adapun hasil pengukuran tegangan pada pinout mikrokontroller pada kondisi isyarat kendali telah dikirim terdapat dari aplikasi server dengan status semua kendalian dalam keadaan hidup, dapat dilihat pada tabel 4.3. Dari tabel hasil pengukuran 4.3 dapat dilihat bahwa rating tegangan berada pada level normal. Hal ini dikarenakan mikrokontroller bekerja sesuai dengan keadaan yang diharapkan dimana tidak ada tegangan yang terukur pada saat mikrokontroller dalam keadaan OFF.

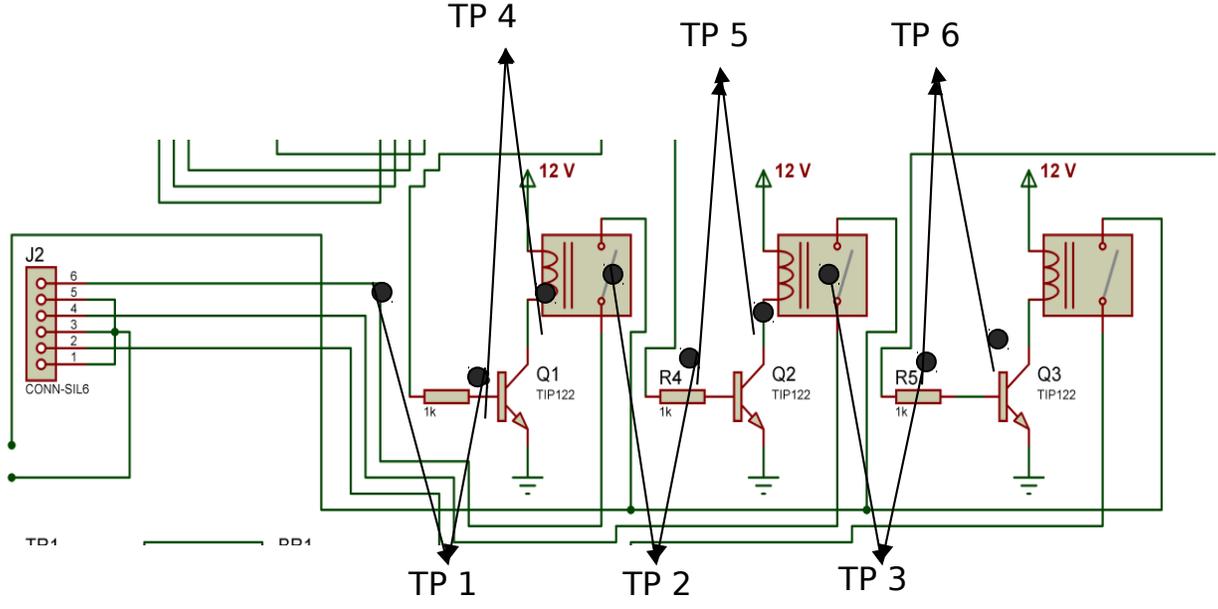
Tabel 4.3 Hasil Pengukuran Mikrokontroller Keadaan ON

<i>Pinout</i>	Keterangan	Tegangan (Volt)
1	PB.0	1,31
2	PB.1	1,17
3	PB.2	1,17
4	PB.3	1,16

5	PB.4	1,16
6	PB.5	1,14
7	PB.6	1,14
8	PB.7	1,14
9	RESET	4,88
10	VCC	4,88
11	XTAL 2	2,43
12	XTAL 1	2,43
13	PD.0 (RX)	0
14	PD.1 (TX)	0
15	PC.3	0,55
16	PC.4	0,50
17	PC.5	0,48

IV.1.3 Pengujian Driver Relay

Pada pengujian rangkaian *driver* beban menggunakan relay ini dilakukan enam kali pengukuran, dan titik pengukuran dapat terlihat pada gambar 4.2



Gambar 4.2 Titik Pengukuran *Driver* Beban

Berikut hasil pengukuran *driver* beban pada saat kondisi dalam keadaan hidup (semua peralata dalam keadaan ON) :

Tabel 4.4 Pengujian *Driver* Beban keadaan ON

Titik	Jenis Catu Daya	Tegangan Terukur (Volt)
TP 1	<i>Output</i> Mikrokontroller Pin 25	4,4 V _{AC}
TP 2	<i>Output</i> Mikrokontroller Pin 26	4,4 V _{AC}
TP 3	<i>Output</i> Mikrokontroller Pin 27	4,4 V _{AC}
TP 4	<i>Output</i> Relay Beban 1	220 V _{AC}
TP 5	<i>Output</i> Relay Beban 2	220 V _{AC}
TP 6	<i>Output</i> Relay Beban 3	220 V _{AC}

Berdasarkan tabel 4.4 didapatkan hasil pengukuran seperti yang diharapkan. Masing-masing pin mikrokontroller yang terhubung dengan relay dapat berfungsi dengan baik yang ditandai dengan nilai tegangan yang terukur dan berfungsinya relay dengan baik. Adapun hasil pengukuran *driver* beban pada saat kondisi dalam keadaan *standby* (semua peralatan dalam keadaan OFF) :

Tabel 4.5 Tabel Pengujian *Driver* Beban keadaan OFF

Titik	Jenis Catu Daya	Tegangan Terukur (Volt)
TP 1	<i>Output</i> Mikrokontroller Pin	0
TP 2	<i>Output</i> Mikrokontroller Pin	0
TP 3	<i>Output</i> Mikrokontroller Pin	0
TP 4	<i>Output</i> Relay Beban 1	0
TP 5	<i>Output</i> Relay Beban 2	0
TP 6	<i>Output</i> Relay Beban 3	0

Berdasarkan tabel 4.5 tegangan yang terukur dalam keadaan *standby* adalah 0V untuk semua *pinout* mikrokontroller. Pada kondisi tersebut relay tidak berfungsi seperti yang diharapkan.

IV.2 Pengujian Perangkat Lunak

IV.2.1 Pengujian Fungsional

Pengujian ini dilakukan untuk menguji fungsi dari perangkat lunak yang telah dibuat khusus pada *server* dan *client*. Dalam pengujian kali ini, aplikasi mikrokontroller tidak dilakukan pengujian dikarenakan pengujian fungsi mikrokontroller dapat dilakukan dengan menguji kualitas kinerja sistem secara keseluruhan, dan pengujian kinerja akan dibahas berikutnya.

Pada aplikasi *server*, pengujian dilakukan dengan melakukan pengujian pada panel dan tombol-tombol yang ada pada halaman kendalian yaitu tombol ON/OFF dari masing-masing perangkat listrik. Pengujian ini ditujukan agar semua tombol-tombol yang tersedia mewakili kondisi kendalian berfungsi dengan baik atau tidak. Berikut adalah hasil dari pengujian dari fungsi tombol-tombol :

Tabel 4.6 Hasil pengujian Fungsional Tombol Kendali *Server*

Jenis Peralatan	Tombol Kendali	Hasil Pengujian
Lampu / Saklar 1	ON	Lampu menyala
	OFF	Lampu padam
AC / Saklar 2	ON	AC menyala
	OFF	AC padam
Water Heater / Saklar 3	ON	<i>Water Heater</i> menyala
	OFF	<i>Water Heater</i> padam

Pada aplikasi *client*, pengujian fungsional dilakukan pada fungsi pembacaan status dan waktu tunda pengaktifan atau penonaktifan perangkat listrik, dimana status ini juga dapat dilihat pada halaman aplikasi *server*. Dan berikut hasil pengujian fungsional dari aplikasi pembaca status kendalian:

Tabel 4.7 Hasil pengujian Fungsional Tombol Kendali Aplikasi Android (*Client*)

Jenis Peralatan	Tombol Kendali	Hasil Pengujian
Lampu / Saklar 1	ON	Lampu menyala
	OFF	Lampu padam
AC / Saklar 2	ON	AC menyala
	OFF	AC padam
Water Heater / Saklar 3	ON	Water Heater menyala
	OFF	Water Heater padam

Dari hasil pengujian pada tabel 4.6 dapat dilihat bahwa ketiga tombol pada aplikasi *server* untuk ketiga saklar berfungsi sebagaimana mestinya. Contohnya pada penggunaan Saklar 1 / Lampu. Saklar 1 berfungsi sebagai tombol kendalian bagi mikrokontroller untuk mengirim perintah agar Lampu yang terhubung dengan saklar 1 berada pada kondisi OFF.



Gambar 4.3
keadaan OFF



Perangkat listrik dalam

Gambar 4.4 Perangkat listrik dalam keadaan OFF

IV.2.2 Pengujian Kecepatan Pengiriman Perintah Pengontrolan

Pengujian ini dimaksudkan untuk menguji seberapa cepat respon kinerja sistem secara keseluruhan dimana pada pengujian ini, menggunakan semua perangkat lunak dan dilakukan per-bagian untuk memudahkan dalam proses pengambilan data.

Bagian-bagian yang diuji adalah :

1. Pengendalian yang dilakukan melalui aplikasi *server* terhadap respon waktu peralatan listrik
2. Pengendalian yang dilakukan melalui aplikasi *client* terhadap respon waktu peralatan listrik

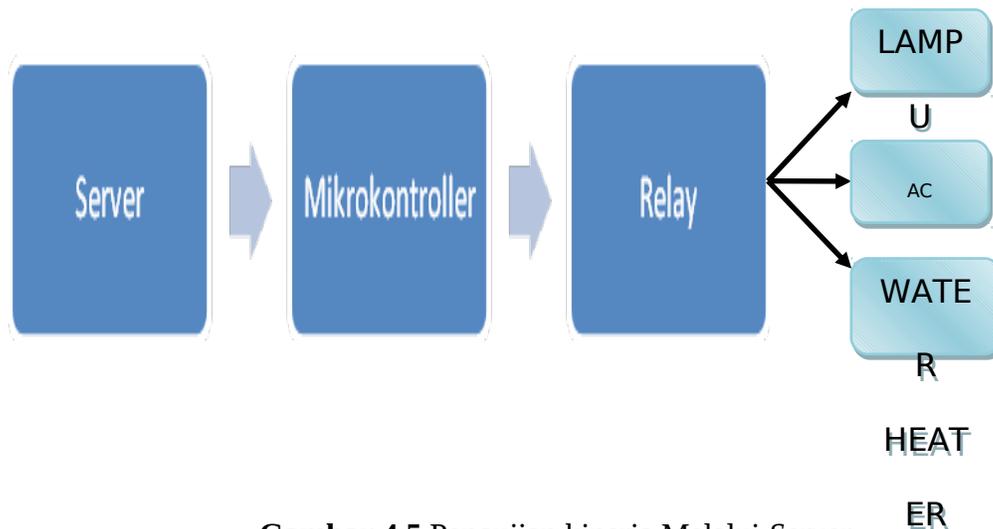
IV.2.2.1 Pengujian melalui *server* terhadap respon waktu

Pengujian ini dilakukan untuk mengetahui seberapa cepat respon beban peralatan listrik terhadap *server*.

- **Komponen yang digunakan**

- Komputer *server*
- *Power supply*
- Rangkaian mikrokontroller
- Rangkaian relay
- Beban peralatan listrik

- **Prosedur pengujian**



Gambar 4.5 Pengujian kinerja Melalui *Server*

Pengujian ini dilakukan dengan merangkai perangkat keras sesuai gambar 4.5, kemudian mengaktifkan *power supply* untuk *driver* beban dan *input 5V* pada rangkaian mikrokontroller. Mengaktifkan aplikasi *server*, kemudian melakukan proses Log-In sebelum memasuki halaman pengontrolan *server*. Setelah memasuki halaman pengontrolan kemudian mengatur port masukan yang digunakan, lalu mengkoneksikan antarmuka *server* dengan mikrokontroller dan menekan tombol ON

untuk saklar 1 (lampu) pada panel piranti beban. Pada saat tombol ON ditekan, secara bersamaan dilakukan penghitungan waktu respon nyala lampu. Kemudian menekan tombol OFF untuk saklar lampu pada panel piranti beban. Pada saat tombol OFF ditekan, secara bersamaan dilakukan penghitungan waktu respon padamnya lampu. Prosedur ini dilakukan berulang untuk mencoba kedua saklar lainnya dengan jarak tetap. Hasil pengujian dapat dilihat pada tabel 4.8.

- **Pembahasan hasil pengujian melalui server**

Dari hasil pengujian pada tabel 4.8 terlihat bahwa pada Saat pengendalian terhadap peralatan listrik rumah tangga diakses langsung oleh server, maka dibutuhkan waktu penundaan rata-rata sekitar 0.33 detik untuk dapat menghidupkan lampu ketika tombol ON telah ditekan, namun ketika tombol OFF pada lampu ditekan, maka dibutuhkan waktu penundaan 0.29 detik untuk dapat mematikan lampu tersebut. Selanjutnya pada AC, jika tombol ON telah ditekan, dibutuhkan waktu penundaan sekitar 0.31 detik untuk dapat menghidupkan AC tersebut melalui server, selanjutnya jika tombol OFF pada AC ditekan, dibutuhkan penundaan sekitar 0.32 detik melalui server untuk mematikan AC tersebut. Kemudian, pada Water Heater, jika tombol ON telah ditekan, dibutuhkan waktu penundaan sekitar 0.25 detik untuk dapat menghidupkan Water Heater tersebut melalui server, selanjutnya jika tombol OFF pada water heater ditekan, dibutuhkan penundaan sekitar 0.33 detik untuk dapat mematikan water heater tersebut.

Tabel 4.8 Hasil Pengujian Melalui Server

Peralatan	Respon terhadap waktu (m/s)
-----------	-----------------------------

	Hidup	Mati
LAMPU	0.21	0.30
	0.39	0.34
	0.40	0.36
	0.48	0.45
	0.21	0.15
	0.39	0.29
	0.14	0.54
	0.30	0.11
	0.45	0.13
AC	0.43	0.20
	0.17	0.37
	0.37	0.11
	0.18	0.58
	0.38	0.44
	0.44	0.11
	0.15	0.47
	0.49	0.29
	0.21	0.38
WATER HEATER	0.39	0.13
	0.15	0.57
	0.13	0.42
	0.29	0.11
	0.44	0.29
	0.15	0.41
	0.18	0.33
	0.19	0.34
	0.34	0.41

Setelah mengolah hasil pengujian sistem menggunakan sistem pada *server*, terlihat bahwa waktu penundaan terkecil terdapat pada pengontrolan peralatan listrik yang dilakukan menggunakan komputer *server* jika dibandingkan dengan pengendalian yang dilakukan oleh *client*. Hal tersebut terjadi karena *server* melakukan eksekusi program secara langsung melalui aplikasi desktop pada komputer *server* yang data isyarat kendalian berupa karakter *string* yang dikirim menuju aplikasi pembaca status lalu kemudian diteruskan menuju mikrokontroler. Kecepatan transfer antara *server* dengan alat kontrol berlangsung sangat cepat dikarenakan,

antara *server* dan alat pengendali terhubung secara langsung melalui perantara port *usb-to-serial*.

IV.2.2.2 Pengendalian melalui *client* terhadap respon waktu

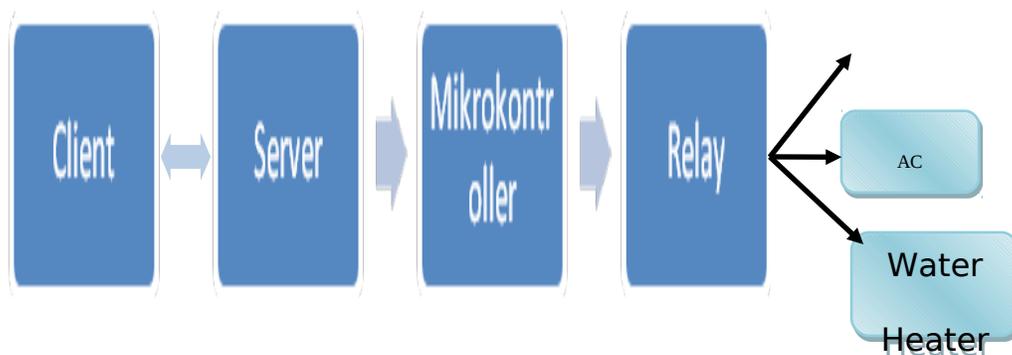
Pengujian ini dilakukan untuk mengetahui seberapa cepat respon beban peralatan listrik terhadap *waktu*.

- **Peralatan yang digunakan :**

- Komputer *server*
- Smartphone / perangkat dengan OS Android.
- Catu Daya
- Rangkaian mikrokontroller
- Rangkaian relay
- Beban peralatan listrik

- **Langkah-langkah Pengujian**





Gambar 4.6 Pengujian kinerja melalui *client*

Pengujian ini dilakukan dengan merangkai perangkat keras sesuai gambar 4.6 untuk pengujian melalui *client*, modul pengontrolan dan *server* harus dalam keadaan ON dan terkoneksi satu sama lain. *Server* harus terkoneksi ke jaringan internet agar dapat dilakukan akses dari *client* ke *server*. *Client* juga harus dikoneksikan ke jaringan internet baik melalui wi-fi ataupun jaringan internet yang didukung provider internet. Setelah terkoneksi ke internet, pengujian tombol-tombol pada panel pengontrolan *client* dapat dilakukan.

Setelah memasuki halaman pengontrolan *client* kemudian menekan tombol ON untuk saklar 1 (lampu) pada panel piranti beban. Pada saat tombol ON ditekan, secara bersamaan dilakukan penghitungan waktu respon hidup lampu. Kemudian menekan tombol OFF untuk saklar lampu pada panel piranti beban. Pada saat tombol OFF ditekan, secara bersamaan dilakukan penghitungan waktu respon padamnya lampu. Prosedur ini dilakukan berulang untuk mencoba kedua saklar lainnya dengan jarak tetap. Hasil pengujian dapat dilihat pada tabel 4.9.

Tabel 4.9 Hasil Pengujian Melalui *Client*

Peralatan	Respon terhadap waktu (m/s)	
	Nyala	Padam
LAMPU	0.75	0.45
	0.17	0.39
	0.39	0.28
	0.59	0.52
	0.17	0.79
	0.98	0.17
	0.49	0.24
	0.66	0.19
	0.13	0.27
AC	0.19	0.13
	0.30	0.21
	0.73	0.16
	0.64	0.37
	0.35	0.23
	0.42	0.28
	0.68	0.64
	0.79	0.37
	0.31	0.40
WATER HEATER	0.29	0.39
	0.19	0.75
	0.55	0.89
	0.78	0.94
	0.74	0.29
	0.31	0.44
	0.21	0.20
	0.31	0.75
0.10	0.13	

- **Pembahasan hasil pengujian melalui *client***

Dari hasil pengujian pada tabel 4.9 terlihat bahwa pada Saat pengendalian terhadap peralatan listrik rumah tangga diakses langsung oleh komputer *client*, maka dibutuhkan waktu penundaan rata-rata sekitar 0.48 detik untuk dapat menghidupkan lampu ketika tombol ON telah ditekan, namun ketika tombol OFF pada lampu

ditekan, maka dibutuhkan waktu penundaan 0.36 detik untuk dapat mematikan lampu tersebut. Selanjutnya pada AC, jika tombol ON telah ditekan, dibutuhkan waktu penundaan sekitar 0.49 detik untuk dapat menghidupkan AC tersebut melalui *client*, selanjutnya jika tombol OFF pada AC ditekan, dibutuhkan penundaan sekitar 0.31 detik melalui *client* untuk mematikan AC tersebut. Kemudian, pada *Water Heater*, jika tombol ON telah ditekan, dibutuhkan waktu penundaan sekitar 0.38 detik untuk dapat menghidupkan *Water Heater* tersebut melalui *client*, selanjutnya jika tombol OFF pada *water heater* ditekan, dibutuhkan penundaan sekitar 0.53 detik untuk dapat mematikan *water heater* tersebut.

Berbeda dengan pengendalian peralatan listrik yang dilakukan oleh *server*, selain menunggu *server* mengeksekusi program dan membaca status, *client* juga melakukan komunikasi dengan jaringan internet untuk dapat mengakses komputer *server*. Hal tersebut menyebabkan waktu penundaan yang mengakibatkan bertambahnya waktu penundaan ketika *client* mengendalikan peralatan listrik rumah tangga. Sedangkan untuk kecepatan transfer data antara *client* ke *server* sangat tergantung dari kecepatan koneksi internet dan kecepatan transfer dengan *open vpn* yang digunakan sebagai penghubung aplikasi dengan koneksi internet karena, *client* terlebih dahulu mengirimkan data ke *openvpn* lalu *openvpn* mengirimkan data ke *server* dan *server* mengirimkan data ke mikrokontroler lalu mikrokontroler mengirimkan status balikan ke *server* dan *client*.

Dalam sistem pengontrolan melalui *server* dan *client* ini, jarak tidak menjadi faktor utama tertundanya transfer data dari *server* ke mikrokontroler maupun *client*

ke mikrokontroller. Yang paling berpengaruh besar dalam sistem pengontrolan ini adalah kecepatan jaringan internet yang digunakan pada *server* dan *client*.

BAB V

KESIMPULAN DAN SARAN

V.1 Kesimpulan

Berdasarkan hasil perancangan dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan antara lain :

1. Pengontrolan sistem instalasi listrik apartemen untuk mengendalikan ON/OFF peralatan listrik dilakukan dengan menggunakan telepon seluler berbasis Android. Telepon seluler ini berperan sebagai *remote control*, dengan menggunakan jaringan internet sebagai media pengirim data.
2. OpenVPN berfungsi sebagai penghubung antara jaringan internet ke *server* dan jaringan internet ke *client* yang mempercepat proses pengiriman perintah dari *server* ke *client* tanpa terpengaruh oleh cepat lambatnya koneksi internet baik dari *server* maupun *client*. *Client* mengirimkan perintah ke *server* melalui openvpn dan openvpn inilah yang mengirimkan perintah dari *client* ke *server* yang kemudian dari *server* mengirimkan perintah ke driver beban dan setelah perintah di eksekusi, *server* menerima umpan balik yang dikirim kembali ke *client* melalui openvpn.
3. Waktu pengontrolan tidak dipengaruhi oleh pengontrolan melalui *server* dikarenakan *server* terhubung langsung dengan modul pengontrolan. Sedangkan untuk *client*, yang mempengaruhi waktu pengontrolan adalah cepat lambatnya transfer data dari *client* ke openvpn dan openvpn ke *server* lalu *server* mengirimkan kembali perintah ke *client*. Rata-rata waktu penundaan untuk pengujian melalui *server* untuk menyalakan peralatan listrik adalah 0,29m/s, sedangkan waktu penundaan pada saat memadamkan adalah 0.31m/s/. Sedangkan rata-rata waktu penundaan untuk menguji *client* menyalakan peralatan listrik adalah 0.45m/s dan waktu penundaan pada saat memadamkan adalah 0.35m/s.

V.2 Saran

Besar harapan kami bahwa penelitian ini dapat memberikan inspirasi dalam pengembangan penelitian untuk keperluan penelitian mahasiswa elektro. Untuk itu, dalam rangka pengembangan penelitian penelitian ini lebih lanjut, kami memberikan beberapa saran sebagai berikut :

1. Sistem pengendalian ini masih menggunakan PC sebagai perantara antara aplikasi seluler dengan antarmuka pengontrolan. Kami harap nantinya dapat dikembangkan sehingga antarmuka pengontrolan dapat langsung terhubung ke jaringan internet.
2. Parameter lampu, ac dan *water heater* yang dimonitor pada sistem pengendalian ini hanya parameter waktu saja. Kami berharap nantinya ada pengembangan sehingga parameter-parameter lain seperti penggunaan sensor cahaya ataupun suhu dapat dipergunakan.

DAFTAR PUSTAKA

Budiharto, Widodo dan Rizal, Gamayel. 2007. “BelajarSendiri 12 Proyek Mikrokontroler untukPemula”. Jakarta: PT. Elex Media Komputindo.

“DT-AVR Low Cost Micro System”. Diakses November 20, 2012 dari http://innovativeelectronics.com/innovative_electronics/pro_dtavr1cm.htm

Malvino, Albert Paul, 1989. "*Elektronika Komputer Digital*". Erlangga, Jakarta.

"Mujahuddin, Max 232 pada komunikasi serial" Diakses November 13 2012 dari <http://journal.mercubuana.ac.id/data/lecKK-012325-5-1.pdf>.

Huda, Miftakhul dan Bunafit Komputer, 2010. "Membuat Aplikasi Database dengan JAVA, MySQL dan NETBeans". Jakarta: PT. Elex Media Komputindo.

Suprianto, Dudit dan Agustina, Rini, 2012. "Pemrograman Aplikasi ANDROID". Mediakom, Yogyakarta.

Noviandisti, Agria Tri, 2012. "Perencanaan dan Perancangan Segreen Apartemen". Bandung: Univeristas Pendidikan Indonesia.