

APLIKASI MIKROKONTROLER ATMEL 90S8515 SEBAGAI PENGHITUNG PADA KALKULATOR RESISTOR E12

Oleh: Argo Budi Utomo (L2F099576)

Abstrak — Resistor yang tersedia di pasaran mempunyai nilai tertentu yang dibagi berdasarkan nilai toleransinya yaitu seri e24, e12, e6 dan e3. Di mana seri e12 adalah seri resistor yang paling umum tersedia di pasaran. Jika nilai resistor yang dibutuhkan tidak terdapat dalam cakupan standar seri e12. Maka harus dirangkai beberapa resistor e12 agar nilai resistansinya sesuai dengan yang dibutuhkan

Pada tugas akhir ini dibuat alat kalkulator resistor e12 berbasis AT90S8515 sebagai alat bantu untuk merangkai beberapa resistor e12 untuk mendapatkan nilai resistansi yang mendekati nilai resistansi yang dibutuhkan. Keunggulan yang dimiliki mikrokontroler AVR AT90S8515 adalah dapat diprogram dengan menggunakan bahasa C. Dimana untuk perhitungan variabel yang agak “rumit”, lebih mudah menggunakan bahasa C daripada bahasa assembler. Selain itu juga menjadi lebih praktis karena dengan mikrokontroler maka penggunaan komputer bisa digantikan hanya dengan menggunakan IC yang berukuran kecil sehingga mudah untuk dibawa-bawa.

Hasil pengujian dan analisa menunjukkan bahwa kalkulator resistor yang dibuat dapat bekerja dengan baik. Untuk nilai-nilai resistor tertentu terdapat perbedaan antara hasil perhitungan kalkulator resistor dan resistor yang diinginkan. Tetapi perbedaan yang dihasilkan ini sangat kecil. Hal ini disebabkan karena adanya program untuk memberikan error pada resistor yang dimasukkan. Program ini untuk mempercepat waktu processing dan untuk menghindari tidak dapat ditemukannya kombinasi resistor untuk resistor yang dimasukkan.

Kata Kunci: AVR AT90S8515, bahasa C, resistor seri e12.

1 Pendahuluan

1.1 Latar Belakang

Biasanya resistor dipakai dalam rangkaian pembagi tegangan, rangkaian pengurang arus (sesuai fungsi dasarnya), rangkaian filter (sebagai penentu frekuensi cut off), rangkaian penguat dengan op-amp (penentu nilai penguatan) dan rangkaian pembangkit frekuensi/clock (penentu frekuensi resonansi). Resistor yang tersedia di pasaran mempunyai nilai tertentu yang dibagi berdasarkan nilai toleransinya yaitu seri e24, e12, e6 dan e3. Di mana seri e12 adalah seri resistor yang paling umum tersedia di pasaran. Jika nilai resistor yang dibutuhkan tidak terdapat dalam cakupan standar seri e12. Maka kita harus merangkai beberapa resistor e12 agar nilai resistansinya sesuai dengan yang dibutuhkan.

Keunggulan yang dimiliki mikrokontroler AVR AT90S8515 adalah dapat diprogram dengan menggunakan bahasa C. Dimana untuk perhitungan variabel yang agak “rumit”, lebih mudah menggunakan bahasa C daripada bahasa assembler. Selain itu juga menjadi lebih praktis karena dengan mikrokontroler maka penggunaan komputer bisa digantikan hanya dengan menggunakan IC yang berukuran kecil sehingga mudah untuk dibawa-bawa. Dengan pertimbangan-pertimbangan tersebut maka dibuat alat bantu sebagai penghitung resistor e12 berbasis mikrokontroler AT90S8515.

1.2 Tujuan

Tujuan yang hendak dicapai pada Tugas Akhir ini, yaitu :

1. Membuat perangkat keras dan lunak kalkulator resistor berbasis mikrokontroler AT90S8515.
2. Mempermudah mencari kombinasi resistor dengan besar resistansi yang dikehendaki yang tidak tersedia dalam seri e12.
3. Agar dapat dibawa dan digunakan secara langsung. Karena jika hanya berbentuk perangkat lunak, harus menggunakan komputer untuk menjalankannya.

1.3 Pembatasan Masalah

Batasan masalah dalam Tugas Akhir ini adalah:

- Perangkat keras yang digunakan berbasis mikrokontroler AT90S8515

- Pada tugas akhir ini yang dibahas hanya pada sisi teknisnya saja.
- Resistor kalkulator ini menggunakan resistor-resistor e12 dengan resistansi antara 10 Ω sampai dengan 1 M Ω .
- Resistor kalkulator ini hanya dapat mencari besar resistor antara 1 sampai dengan 3 M Ω .

2 Dasar Teori

2.1 Resistor Seri E12

The Electronic Industries Association (EIA) menetapkan standar manufaktur untuk nilai resistor. Standar manufaktur nilai resistor yang ditetapkan oleh EIA memiliki beberapa variasi toleransi yaitu antara lain: E3 dengan toleransi 50%, E6 dengan toleransi 20%, E12 dengan toleransi 10% (resistor yang paling banyak dipakai dan dijual dipasaran), E24 dengan toleransi 5%, E48 dengan toleransi 2%, E96 dengan toleransi 1% dan E192 dengan toleransi .5% atau lebih tinggi. Nilai-nilai resistor seri e12 diperlihatkan pada Tabel 2.1 di bawah ini

Tabel 2.1 Resistor seri E12

10R	100R	1K0	10K	100K	1M0
12R	120R	1K2	12K	120K	
15R	150R	1K5	15K	150K	
18R	180R	1K8	18K	180K	
22R	220R	2K2	22K	220K	
27R	270R	2K7	27K	270K	
33R	330R	3K3	33K	330K	
39R	390R	3K9	39K	390K	
47R	470R	4K7	47K	470K	
56R	560R	5K6	56K	560K	
68R	680R	6K8	68K	680K	
82R	820R	8K2	82K	820K	

2.2 Mikrokontroler AT90S8515

Mikrokontroler AT90S8515 adalah sebuah sistem mikrokontroler 8 bit dan memiliki 8 Kbyte flash. Instruksi-instruksi maupun pin-nya kompatibel dengan standar MCS51. Dengan jenis memori flash memudahkan memori program untuk diprogram ulang sistem.

Selain itu AT90S8515 mempunyai banyak fitur. Fitur tersebut antara lain adalah Flash memory sebesar 4 Kwords mempunyai EEPROM internal sebesar 512 bytes, RAM sebesar 512 bytes, kecepatan clock sampai dengan 8 MHz dengan frekuensi kerja sama dengan frekuensi kristal osilator, 11 interupsi, analog komparator, 32 register yang langsung terhubung dengan ALU sehingga menyerupai akumulator, serta dirancang dengan mempertimbangkan sifat-sifat pengkodean bahasa C.

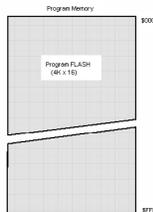
2.3 Organisasi Memori

AVR menggunakan arsitektur Harvard sehingga memisahkan memori serta bus data dengan program. Program ditempatkan *Flash Memory* sedangkan memori data terdiri dari 32 buah register serbaguna, 64 register serbaguna, 512 bytes internal SRAM dan 64 Kbytes SRAM eksternal yang dapat ditambahkan. Berdasarkan fungsinya terdapat 4 macam memori pada AT90S8515 yaitu:

1. Memori Program
2. Memori Data
3. 32 Register Serbaguna.
4. Ruang Memori I/O

Berikut ini penjelasan dari masing-masing memori pada AT90S8515:

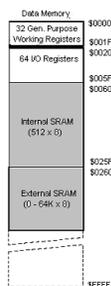
1. Memori Program



Gambar 2.1 Memori Program.

AT90S8515 mempunyai kapasitas memori program sebesar 8 Kbytes. Format memori program yang digunakan adalah 16 bit sehingga format memori program yang digunakan adalah 4Kx16bit. Memori Flash ini dirancang untuk dapat di hapus dan tulis sebanyak seribu kali

2. Memori Data



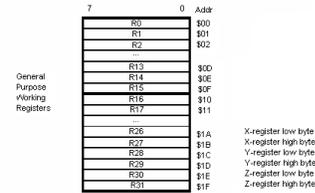
Gambar 2.2 Memori Data.

Memori data pada AT90S8515 tersebut terdiri atas memori data internal dan eksternal. SRAM internal ini menempati ruang alamat setelah 32 lokasi register serbaguna, dan setelah 64 register I/O. Jika SRAM eksternal digunakan, ia

akan mengikuti besar SRAM external sampai dengan maksimum 64K tergantung ukuran SRAM eksternal.

Ketika alamat yang digunakan untuk mengakses melebihi alamat dari ruang memori internal SRAM, maka SRAM eksternal akan terakses dengan instruksi yang sama dengan instruksi pada SRAM internal. Jalur pengontrolan SRAM untuk baca dan tulis masing-masing menggunakan kaki \overline{RD} dan \overline{WR} .

3. 32 Register Serbaguna



Gambar 2.3 32 Register Serbaguna.

Pada 32 bytes *general purpose working register* atau register umum serbaguna mendukung adanya konsep register akses cepat. Ini berarti bahwa waktu akses dari register adalah satu detak dan berarti pula satu operasi ALU (Arithmetics Logic Unit) dilaksanakan.

4. Ruang memori I/O

Ruang memori I/O pada AT90S8515 berisi 64 alamat untuk fungsi periferal CPU seperti register kontrol, Timer/Counter, dan fungsi I/O yang lain. Semua I/O dan periferal AT90S8515 ditempatkan dalam ruang I/O.

2.4 PORTA/B/C/D, DDRA/B/C/D Dan PINA/B/C/D

PORTA/B/C/D dan DDRA/B/C/D merupakan *register-register* yang digunakan untuk mengatur PORTA/B/C/D. Sedangkan PIN PORTA/B/C/D digunakan untuk mengakses pin pada *port* A,B,C,D secara individu. Hubungan antara PORT PORTA/B/C/D dan DDR PORTA/B/C/D diperlihatkan pada Tabel 2.2.

Tabel 2.2 Kombinasi bit DDRA/B/C/Dn dan PORTA/B/C/Dn

DDRBn/Dn	PORTBn/Dn	I/O	Keterangan
0	0	Input	Tri-state (High-Z)
0	1	Input	PORTA/B/C/Dn akan menghasilkan arus jika eksternal pull-low
1	0	Output	Push-pull zero output
1	1	Output	Push-pull one output

2.5 Bahasa C

Bahasa ini digunakan untuk mengatur kerja dari Mikrokontroler AT90S8515. Bahasa C yang digunakan ini hampir semuanya sesuai dengan standar dari ANSI (*American National Standards Institute*) dengan penambahan beberapa fitur untuk menyesuaikan dengan arsitektur AVR dan sistem pada mikrokontroler. Program *compiler* C yang digunakan pada Tugas Akhir ini adalah CodeVisionAVR versi 1.23.8c yang dapat diperoleh di website www.hpinfotech.ro. Program hasil kompilasi yaitu *file object* COFF dapat didebug

menggunakan AVR studio versi 3.55 yang juga dapat diperoleh secara cuma-cuma di website www.atmel.com.

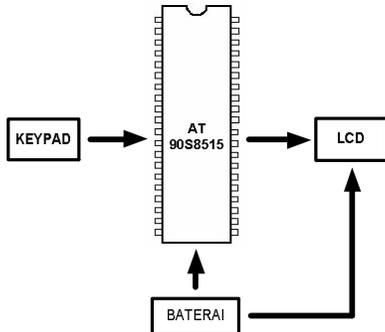
2.6 Bahasa Rakit (Assembly)

Bahasa rakit lain yang digunakan adalah bahasa ASSEMBLER. Bahasa ini digunakan untuk menginisialisasikan port bagi LCD.

3 Perancangan Perangkat Keras Dan Perangkat Lunak

3.1 Perancangan Perangkat Keras (Hardware)

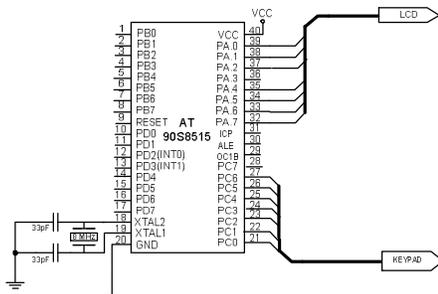
Blok diagram sistem secara keseluruhan pada perancangan alat ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Blok diagram sistem secara keseluruhan.

3.2 Rangkaian Sistem Minimum Mikrokontroler AT90S8515

Mikrokontroler ini dirancang sebagai suatu rangkaian *single chip* seperti tampak pada Gambar 3.2.

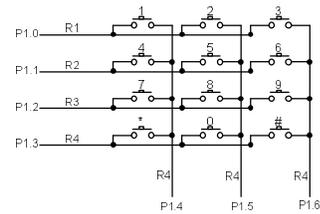


Gambar 3.2 Rangkaian Pengontrol Mikrokontroler AT90S8515

Rangkaian mikrokontroler AT90S8515 ini menggunakan osilator kristal eksternal sebagai pembangkit frekuensi internal (*on chip osilator*) sebesar 8 MHz yang dapat digunakan sebagai sumber detak bagi CPU. Untuk menggunakan osilator internal diperlukan sebuah kristal atau resonator keramik antara kaki Xtal1 dan kaki Xtal2 dan sebuah kapasitor ke *ground*. Pada port A pin0-2, dan pin 4-7 diperuntukkan untuk tampilan LCD. Sedangkan pada port C pin0-6 diperuntukkan untuk *input* dari *keypad*.

3.3 Keypad

Keypad digunakan sebagai masukan input yang langsung dihubungkan ke mikrokontroler. *Keypad* yang digunakan adalah tipe matrik 3x4 seperti tampak pada Gambar 3.3 yang dihubungkan ke Port C. Cara kerja di dalam pemrograman yaitu salah satu jalur dipakai sebagai input dan lainnya sebagai output atau sebaliknya.



Gambar 3.3 Konstruksi keypad 4x3.

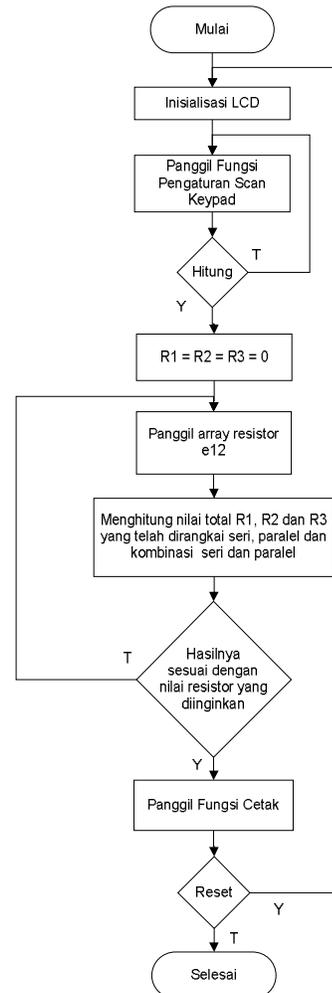
Pada *keypad* jenis ini, metode yang digunakan adalah metode *scanning*. Pada saat *scan* baris Pin 1, Pin 6, Pin 5 dan Pin 3 berfungsi sebagai input sedangkan Pin 2, Pin 0 dan Pin 4 berfungsi sebagai output. Pada saat *scan* kolom Pin 1, Pin 6, Pin 5 dan Pin 3 berfungsi output sedangkan Pin 2, Pin 0 dan Pin 4 berfungsi sebagai input.

3.4 Perancangan Perangkat Lunak (Software)

Pada tugas akhir ini bahasa pemrograman yang digunakan adalah bahasa C. Hal ini dikarenakan pemrograman mikrokontroler dengan menggunakan bahasa C akan lebih terstruktur dan mudah dipahami selain itu juga karena bahasa C merupakan gabungan dari bahasa tingkat tinggi dan juga tingkat rendah yang menyediakan kemampuan operasi-operasi bit, *byte*, pengaksesan alamat-alamat *memori*, dan register.

3.5 Program Utama

Dalam Gambar 3.4 diperlihatkan diagram alir program utama dalam perancangan Tugas Akhir ini.

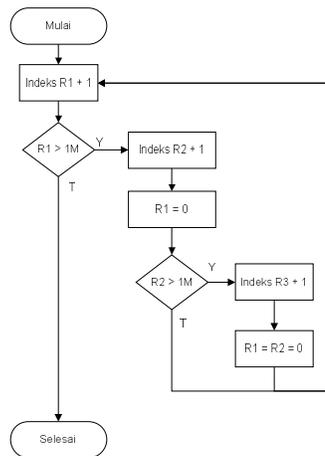


Gambar 3.4 Diagram alir program utama.

Kerja dari perangkat lunak ini secara garis besar adalah setelah eksekusi program maka mikrokontroler akan menginisialisasi terlebih dahulu sistem. Kemudian mikrokontroler melakukan *scan keypad* kemudian menghitung hasil *scan keypad*. Setelah itu mikrokontroler menghitung beberapa resistor e12 yang dirangkai seri, paralel maupun kombinasi seri dan paralel. Kemudian membandingkan nilai total resistansi resistor e12 yang telah dirangkai tersebut dengan hasil perhitungan *scan keypad*. Jika telah didapatkan hasil yang sesuai mikrokontroler menampilkan nilai-nilai resistor e12 yang telah dirangkai tersebut di LCD.

3.6 Fungsi Array Resistor e12

Nilai-nilai resistor e12 dikelompokkan dalam *array*. *Array* resistor e12 ini akan terus dipanggil jika hasil perhitungan beberapa resistor e12 ini belum sama atau mendekati nilai yang diinginkan.

Gambar 3.5. Diagram alir *array* resistor e12

Fungsi *array* resistor e12 berisi program untuk memanggil indeks *array* R1, R2 dan R3 secara berurutan dari yang terkecil. Setiap nilai indeks *array* menunjukkan nilai-nilai resistor e12.

4 Pengujian Dan Analisa

Pengujian tugas akhir ini dilakukan dengan melakukan pengujian pada sistem secara keseluruhan yaitu dengan membandingkan hasil perhitungan kalkulator resistor yang berupa nilai resistor e12 yang dirangkai dengan seri, paralel ataupun kombinasi seri dan paralel dengan nilai resistor yang diinginkan.

4.1 Uji Coba 1

Pada pengujian pertama ini dicoba dimasukkan nilai-nilai resistor yang digunakan dibebberapa rangkaian elektronika. Sebagai contoh resistor yang digunakan pada rangkaian pengikut emitter dengan V_{out} 173 mV, rangkaian pengurang arus untuk membatasi arus yang melewati LED dan sebagainya.

Jika resistor yang diinginkan lebih kecil dari 10 K maka mikrokontroler akan melakukan beberapa operasi aritmatika pada tiga buah nilai variabel yang terdapat pada R1, R2 dan R3. Hasil dari operasi aritmatika ini dibandingkan dengan nilai resistor yang diinginkan. Operasi aritmatika yang dilakukan adalah penghitungan nilai total R1, R2 dan R3 yang dimisalkan sebagai resistor seri e12 yang dirangkai seri, paralel dan kombinasi antara seri dan paralel. Nilai variabel pada R2 dan R3 mula-mula nol. Sedangkan nilai R1 mula-mula bernilai 10 (nilai terkecil resistor e12). Nilai variabel R1, R2 dan R3 ini

akan terus berubah berurutan dimulai dari nilai resistor e12 yang terkecil sampai hasil dari operasi aritmatika pada R1, R2 dan R3 sama atau mendekati nilai resistor yang diinginkan. Dengan kata lain, mikrokontroler melakukan 2 proses *looping* yaitu *looping* nilai-nilai variabel R1, R2 dan R3 dan *looping* operasi penghitungan nilai total variabel R1, R2 dan R3. Jika sudah didapatkan nilai yang sesuai dengan nilai resistansi yang diinginkan mikrokontroler akan berhenti melakukan *looping* dan kemudian menampilkan hasilnya pada LCD.

Untuk nilai resistor yang nilainya antara 10 K sampai 2 M proses penghitungannya sama dengan yang dilakukan pada resistor yang nilainya kurang dari 10 K tetapi mikrokontroler melakukan *looping* nilai variabel R1, R2 dan R3 yang berbeda dengan yang dilakukan jika nilai resistor yang diinginkan kurang dari 10 K. Untuk nilai variabel R1 dimulai dari nilai sepuluh (nilai terkecil resistor seri e12) sedangkan nilai variabel R2 dimulai dari nilai nol. Sedangkan untuk nilai variabel R3 tidak dilakukan proses *looping*. Variabel R3 tetap bernilai nol. Jika nilai variabel R1 dan R2 sudah sampai 1 M (nilai terbesar resistor seri e12) tetapi belum didapatkan nilai yang sesuai dengan nilai resistor yang diinginkan maka mikrokontroler akan melakukan *looping* proses penghitungan pada nilai variabel R1, R2 dan R3 yang dimulai dari nilai satu mega (nilai terbesar resistor seri e12) sampai ditemukan nilai resistor yang sesuai dengan nilai resistor yang diinginkan.

Pengujian pada uji coba 1 ini dilakukan dengan dua puluh lima macam nilai resistor yang digunakan dibebberapa rangkaian elektronika. Hasil pengujian pada uji coba ini diperlihatkan pada Tabel 4.1 berikut ini:

Tabel 4.1 Hasil pengujian pada uji coba 1

Resistor Yang Diinginkan (Ω)	Hasil Pengujian	Error (%)
240	120+120	0
600	1.2K 1.2K	0
300	150R+150R	0
100	100R	0
60	120R 120R	0
30	15R+15R	0
5	10R 10R	0
10	10R	0
2 K	1.0K+1.0K	0
4 K	3.9K+100R	0
470	470R	0
3 K	1.5K+1.5K	0
500	1.0K 1.0K	0
75	150R 150R	0
2.5 K	1.5K+1.0K	0
900	1.8K 1.8K	0
5.1 K	3.9K+1.2K	0
6.2 K	4.7K+1.5K	0
950	680R+270R	0
2.22 K	2.2K+10R+10R	0
3.5 K	[33K 3.9K]+10R	0.062
8	[22R+18R] 10R	0
95	68R+27R	0
150	150R	0
33K	33K	0

Pada Tabel 4.1 tampak bahwa untuk nilai resistor 3.5 K Ω terdapat perbedaan antara resistor yang diinginkan dengan hasil pengujian. Nilai resistor diberi nilai toleransi untuk mempercepat waktu *processing* dan untuk menghindari tidak dapat ditemukannya kombinasi resistor yang sesuai dengan nilai resistor yang diinginkan. Karena pada kalkulator resistor ini maksimal hanya mengkombinasikan 3 buah resistor e12. Sehingga tidak semua nilai resistansi dapat dicari kombinasinya dengan hasil yang seratus persen sama. Dengan adanya toleransi ini dapat dicari kombinasi resistor yang hasilnya mendekati nilai resistor yang diinginkan.

Berdasarkan data pada Tabel 4.1 di atas juga tampak bahwa untuk nilai 100 Ω , 10 Ω , 470 Ω , 150 Ω dan 33 K Ω hanya ditampilkan satu resistor yaitu nilai resistor itu sendiri. Hal ini terjadi karena resistor tersebut tersedia di seri e12. Hal ini sesuai dengan algoritma program yang hanya menampilkan satu buah nilai resistor jika resistor yang dimasukkan terdapat pada seri e12.

4.2 Uji Coba 2

Pada pengujian kedua ini dicoba dimasukkan nilai resistor dengan nilai resistansi yang tidak tersedia dalam resistor seri e12. Nilai-nilai resistor tersebut di antaranya adalah: 9 Ω , 29 Ω , 33.67 Ω , 57 Ω , 115 Ω , 213.45 Ω , 235, 555 Ω dan seterusnya.

Pengujian pada uji coba 2 ini dilakukan dengan dua puluh lima macam nilai resistor. Hasil pengujian pada uji coba ini diperlihatkan pada Tabel 4.2 berikut ini:

Tabel 4.2 Hasil pengujian pada uji coba 2

Resistor Yang Diinginkan (Ω)	Hasil Pengujian	Error (%)
9	18R 18R	0
10.35	33R 15R	0.36
29	[270R 18R]+12R	0.43
33.67	120R 47R	0.3
57	47R+10R	0
88	330R 120R	0
115	100R+15R	0
234	[680R 330R]+12R	0.076
555	68K 560R	0.077
875.12	680R+180R+15R	0.013
1.4 K	1.0K+390R+10R	0
5.678 K	5.6K+82R	0.07
10.4 K	8.2K+2.2K	0
67.89 K	[1M 12K]+56K	0.048
90 K	180K 180K	0
125.7 K	12K+560R	0.8
489.45 K	[1M 820K]+39K	0.02
589 K	560K+27K+2.2K	0.033
621 K	[560K 68K]+560K	0.058
897 K	560K+330K+6.8K	0.022
1.125 M	1M+120K+5.6K	0.053
1.22 M	1M+220K	0
1.890 M	1M+560K+330K	0
2.5 M	1M+820K+680K	0
2.99 M	1M+1M+1M	0.033

Pada uji coba 2 ini nilai resistor yang digunakan ada yang bernilai lebih dari 2 M. Untuk nilai resistor yang nilainya lebih dari 2 M proses penghitungannya sama dengan yang dilakukan pada resistor yang nilainya kurang dari 10 K dan resistor yang nilainya antara 10 K dan 2 M tetapi mikrokontroler melakukan *looping* nilai variabel R1, R2 dan R3 yang berbeda. Pertama-tama mikrokontroler melakukan *looping* proses perhitungan pada variabel R1, R2 dan R3. Untuk nilai variabel R3 dimulai dari nilai sepuluh (nilai terkecil resistor seri e12) sedangkan untuk nilai variabel R1 dan R2 dimulai dari 390 K. Nilai variabel R1, R2 dan R3 akan terus berubah dengan berurutan sampai ditemukan nilai resistor yang sesuai dengan nilai resistor yang diinginkan.

Pada Tabel 4.2 tampak bahwa untuk nilai resistor 10.35 Ω , 29 Ω , 33.67 Ω , 234 Ω , 555 Ω , 875 Ω , 5.678 K Ω , 67.89 K Ω , 125.7 K Ω , 489.45 K Ω , 589 K Ω , 621 K Ω , 897 K Ω , 1.125 K Ω dan 2.99 K Ω terdapat perbedaan antara resistor yang diinginkan dengan hasil pengujian. Seperti pada uji coba 1, nilai resistor diberi nilai toleransi untuk mempercepat waktu *processing* dan untuk menghindari tidak ditemukannya kombinasi resistor yang resistansinya sama dengan nilai yang diinginkan.

Pada Tabel 4.2 juga tampak bahwa untuk nilai resistor yang kecil (di bawah 100 Ω) nilai errornya lebih besar dari pada error pada resistor yang resistansinya besar. Hal ini disebabkan resistor e12 paling kecil yang digunakan pada kalkulator resistor ini bernilai 10 Ω . Sedangkan untuk menghitung resistor dengan resistansi yang kecil dibutuhkan beberapa resistor yang kecil pula. Sehingga untuk menghindari tidak dapat ditemukannya kombinasi resistor e12 yang sesuai dengan resistansi yang diinginkan digunakan nilai toleransi yang lebih besar untuk menghitung resistor yang resistansinya kecil.

5 Kesimpulan

Dari hasil pengujian pada alat Kalkulator Resistor E12 dengan menggunakan mikrokontroler AT90S8515 dapat ditarik kesimpulan sebagai berikut :

1. Kalkulator resistor dengan menggunakan mikrokontroler AT90S8515 yang telah dibuat dapat diselesaikan dan bekerja dengan baik.
2. Untuk nilai-nilai resistor tertentu terdapat perbedaan antara hasil perhitungan kalkulator resistor dan resistor yang diinginkan. Hal ini disebabkan karena adanya program untuk memberikan error pada resistor yang dimasukkan. Program ini untuk mempercepat waktu *processing* dan untuk menghindari tidak dapat ditemukannya kombinasi resistor untuk resistor yang dimasukkan.
3. Untuk nilai resistor yang kecil (di bawah 100 Ω) nilai errornya lebih besar dari pada error pada resistor yang resistansinya besar. Hal ini disebabkan resistor e12 paling kecil yang digunakan pada kalkulator resistor ini bernilai 10 Ω .
4. Untuk resistor yang tersedia di seri e12 hanya ditampilkan satu nilai resistor yaitu nilai resistor itu sendiri.

Daftar Pustaka

- [1]. Abdul Kadir, *Pemrograman Dasar Turbo C untuk IBM PC*, ANDI, 1997.
- [2]. Dr. Ir. Widodo Sri Thomas, DEA, Dipl.Ing, *Elektronika Dasar*, 2000.

- [3]. Fajar Mahadmadi., *Embeded C Pada Mikrokontroler AVR AT90S8515*, Tugas Akhir , Teknik Elektro Undip, 2003.
- [4]. Jogyanto Hartono, MBA, Ph.D., *Konsep Dasar Pemrograman Bahasa C*, Edisi Kedua, ANDI, 2000.
- [5]. Malvino P. Albert, Ph.D., *Prinsip-prinsip Elektronika*, Edisi Kedua, Penerbit Erlangga, 1984.
- [6]. Utis Sutisna., *Aplikasi Mikrokontroler AT89C51 Untuk Keamanan Ruangan Pada Rumah Cerdas*, Tugas Akhir , Teknik Elektro Undip, 2003.
- [7]. _____, *8-bit AVR with 8K Bytes In-System Programable Flash*, <http://www.Atmel.com>, 2003.
- [8]. _____, *CodeVisionAVR Version1.0.1.7 User Manual*, <http://www.hpinfofotech.ru>, 2003.
- [9]. _____, *Hippy's Happy Resistor Kalkulator*, <http://www.hippy.freeseerve.co.uk>, 2004.
- [10]. _____, *E Series Resistor Value*, <http://www.hippy.freeseerve.co.uk>, 2004.
- [11]. _____, *Resistore*, <http://www.electroniclab.com>, 2004.
- [12]. _____, *Resistor values*, <http://www.logwell.com>, 2004.
- [13]. _____, *Liquid Crystal Display Module M1632 User Manual*. Seiko Instrumen Inc., Japan, 1987.
- [14]. _____, *AVR Studio User Manuat*, <http://www.Atmel.com>, 2003.



Penulis adalah mahasiswa S-1 Angkatan '99 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro, dengan konsentrasi di bidang Elektronika Telekomunikasi.
E-mail: argoo@telkom.net

Semarang, Desember 2004
Mengetahui,

Pembimbing I

Pembimbing II

Sumardi, ST, MT
NIP. 132 125 670

Trias Andromeda, ST, MT
NIP. 132 283 185