

# STANDALONE AVR PROGRAMMER

Wiedjaja<sup>1</sup>; Hendra Tejo Saputra<sup>2</sup>; Indra Buana Utama<sup>3</sup>; Rinaldo Yotanto Setiawan<sup>4</sup>

<sup>1, 2, 3, 4</sup>Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Bina Nusantara,  
Jl. K.H. Syahdan No. 9, Kemanggis/Palmerah, Jakarta Barat 11480,  
steff@binus.edu

## ABSTRACT

Article clarifies a research having goals to make AVR programming system that is portable, by making system modul based on AVR AT8535 as AVR Programmer/Master and AT8515 as AVR Socket. The module instruction at first is hex file was sending from PC to AVR Programmer/Master that later will be saved to external memory. After the hex file saved, programming process will not need help from PC anymore. The saved hex file that saved to external memory can be choosen that later will be written to AVR Socket. So, the AVR program can be done repeatedly as long as data on external memory still saved. The standalone that was made can only programed AVR ATmega 8515L. From the research ita can be concluded that on case of repeated program, Stand Alone AVR able to save time 20% more faster that programming using AVR codevision.

**Keywords:** standalone AVR program, external memory, module

## ABSTRAK

Artikel menjelaskan suatu penelitian yang bertujuan membuat suatu sistem pemrograman AVR yang bersifat portable, yaitu dengan membuat sebuah modul sistem berbasis AVR AT8535 sebagai AVR Programmer/Master dan AT8515 sebagai AVR Socket. Cara kerja modul adalah mula-mula data file hex dikirim dari PC ke AVR Programmer/Master kemudian akan menyimpannya ke Memory Eksternal. Setelah file hex tersimpan, proses pemrograman tidak memerlukan bantuan PC lagi. Data file hex yang sudah tersimpan pada Memory Eksternal dapat dipilih untuk nantinya ditulis ke AVR Socket. Dengan demikian, pemrograman AVR dapat dilakukan berulang-ulang selama data pada Memory Eksternal masih tersimpan. Standalone yang dibuat hanya dapat memprogram AVR ATmega 8515L. Dari hasil percobaan didapat simpulan bahwa pada kasus pemrograman berkali-kali, Stand Alone AVR dapat menghemat waktu 20% lebih cepat dari pemrograman menggunakan AVR codevision.

**Kata kunci:** standalone AVR program, memori eksternal, modul

---

## PENDAHULUAN

Peranan elektronika di segala bidang menjadi semakin besar pada abad kedua puluh satu ini. Bermula dari penerapan rangkaian elektronika analog, kemudian digital, dan kini hampir semua peralatan sudah menggunakan sistem mikrokontroler, seperti *handphone*, televisi, *radiocassete*, dan mesin cuci. Mikrokontroler yang dibuat juga telah menjadi semakin canggih, jutaan transistor dijejalkan ke dalamnya, miniaturisasi dimensi semakin ditingkatkan dengan kemampuan mengolah program yang lebih kompleks sehingga memungkinkan untuk penerapan di segala bidang.

Sekarang banyak jenis mikrokontroler yang beredar di pasaran, seperti MCS, AVR, ARM. Penelitian ini membahas bagaimana pemrograman

mikrokontroler AVR dapat diprogram untuk keperluan industri atau keperluan lainnya.

Tujuan penelitian adalah merancang dan membuat sebuah sistem pemrograman AVR yang dapat berdiri sendiri. Manfaat yang diharapkan adalah meningkatkan mobilitas dan kecepatan dalam proses pemrograman AVR karena PC hanya diperlukan untuk penyimpanan file hex ke Standalone AVR Programmer.

## PEMBAHASAN

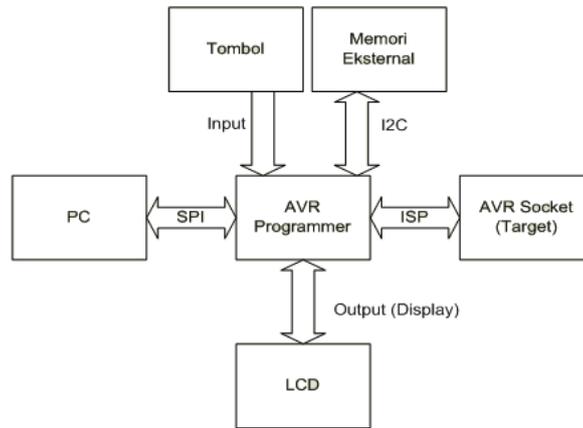
### Rancangan Perangkat Keras

Blok diagram rancangan perangkat keras dapat dilihat pada Gambar 1. Modul AVR Programmer terhubung dengan memori eksternal sebagai tempat penyimpanan file yang akan diprogram, socket untuk

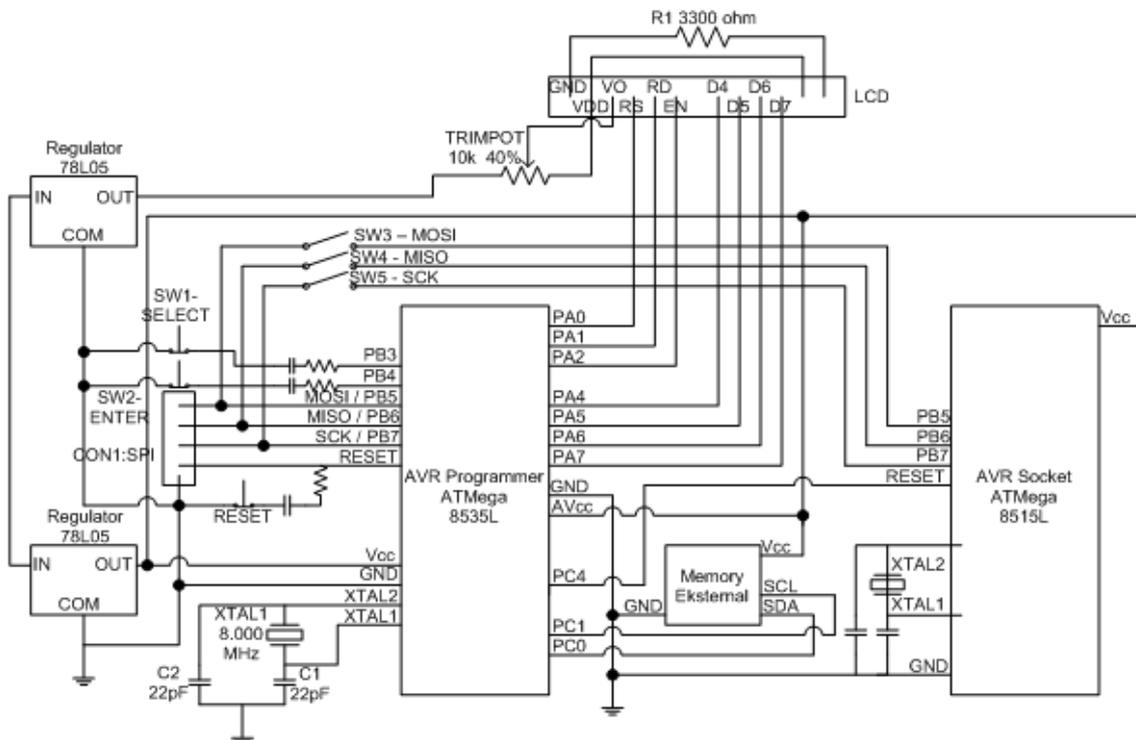
AVR, tombol untuk pemilihan modus operasi, LCD untuk tampilan status, dan *Personal Computer* (PC) yang berfungsi untuk men-download program dalam format file .hex kedalam AVR Programmer. Adapun cara kerjanya adalah pertama memilih mode PC Mode dengan Tombol Select kemudian jalur SPI dipindahkan ke PC dengan, pada kondisi ini AVR programmer siap menerima program yang ditransfer oleh PC. Setelah program pada PC mentransfer file .hex ke modul AVR programmer, file ini ke dalam

eksternal memory, kemudian Mode AVR programmer dipindahkan ke Mode Programming dengan memilih tombol Select dan disertai tombol OK untuk melakukan programming ke AVR yang terpasang pada AVR Socket.

Berikut adalah skematik lengkap rangkaian Standalone AVR Programmer menggunakan ATmega8535L sebagai komponen utama dan AT24C512 sebagai memori eksternal yang diakses melalui protokol I2C.

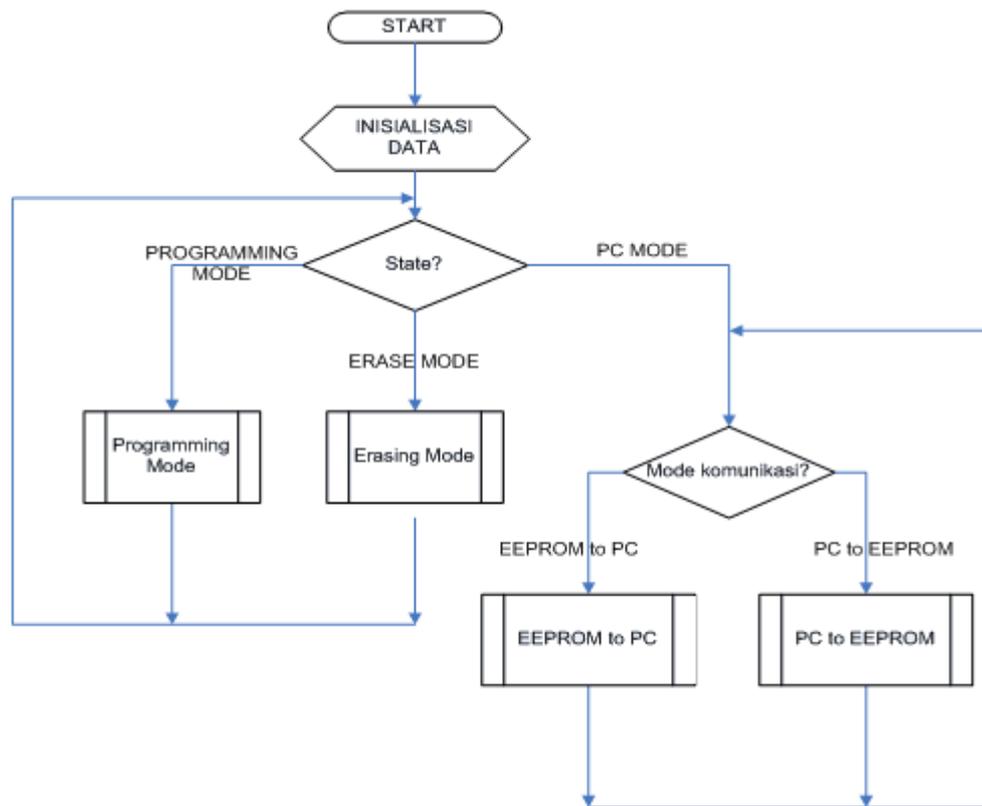


Gambar 1 Blok Diagram Sistem



Gambar 2 Skematik Rangkaian AVR Programmer

## Rancangan Perangkat Lunak



Gambar 3 Diagram Alir Program Utama

Program utama pada AVR Programmer dibuat dengan *state switch case*. Program itu akan melakukan proses *looping* yang secara terus-menerus akan memeriksa state saat ini. *User* dapat memilih *state* mana yang diinginkan dengan dua buah tombol yang tersedia pada modul Standalone AVR Programmer. Ada tiga mode yang dapat dipilih, yaitu *Programming Mode*, *PC Mode*, dan *Erase Mode*.

### Diagram Alir *Programming Mode*

Jika *user* memilih mode ini, mula-mula AVR Programmer akan memeriksa apakah ada file yang tersimpan di Memory Eksternal. Jika ada file yang tersimpan maka selanjutnya *user* dapat memilih file hex menggunakan tombol Select. Setelah *user* menekan tombol OK, AVR Programmer akan memeriksa keberadaan AVR Socket dengan mengirimkan perintah Programming Enable. Jika AVR Socket terhubung, AVR Socket akan membalas dengan mengirimkan nilai 53 hexa. Jika AVR Programmer mendapat nilai balasan ini, berarti pemrograman dapat dilanjutkan. AVR Programmer

akan mengambil data hex yang tersimpan di Memory Eksternal per byte, dan dituliskan ke memory Flash pada AVR Socket. Proses ini dilakukan terus menerus sampai seluruh data hex telah dituliskan. Setelah proses penulisan selesai, dilakukan proses verifikasi data dengan membandingkan data yang tersimpan di Memory Eksternal dengan data yang baru saja dituliskan ke AVR Socket. Proses verifikasi ini dilakukan dari byte awal sampai akhir.

### Diagram Alir *PC Mode*

Dalam PC Mode ini, ada dua proses yang dapat dilakukan oleh AVR Programmer, yaitu proses untuk menyimpan file hex ke memory eksternal dan program untuk mengirim file hex dari memory eksternal ke PC. AVR Programmer akan memilih proses mana yang harus dijalankan berdasarkan perintah yang dikirimkan dari PC (Lihat Gambar 4).

Gambar 4 Diagram Alir *Programming Mode*

Modus kerja Standalone AVR Programmer pada PC Mode sebagai berikut. Pertama, penyimpanan file hex dari PC ke Memory Eksternal. Pada proses ini, AVR Programmer mula-mula akan memeriksa jumlah file hex yang tersimpan dalam Memory Eksternal. Jika file hex-nya belum penuh maka AVR Programmer akan memberitahu ke PC untuk mulai mengirimkan data yang akan dituliskan ke Memory Eksternal. Mula-mula yang dikirimkan adalah nama file, yaitu sebesar 16 byte. Setelah itu, baru dikirimkan panjang data. AVR Programmer akan melakukan *looping* untuk mengambil data dari PC dan menuliskannya ke Memory Eksternal, dan *looping* dilakukan sebanyak panjang data yang dikirimkan oleh PC tadi.

Kedua, pengiriman file hex dari Memory Eksternal ke PC. Pada proses ini, mula-mula AVR Programmer akan mengambil nomor index file yang dikirimkan oleh PC. Jika filenya ditemukan, proses akan dilanjutkan oleh AVR Programmer dengan membaca nama file dari Memory Eksternal dan mengirimkan ke PC. Setelah itu, AVR Programmer akan membaca panjang data Memory Eksternal dan dikirimkan ke PC. AVR Programmer selanjutnya akan melakukan perulangan untuk membaca data dari Memory Eksternal dan mengirimkannya ke PC, sebanyak panjang data yang didapat.

### **Diagram Alir Erase Mode**

Pada mode ini, AVR Programmer akan mengambil jumlah file hex yang tersimpan di Memory Eksternal. Jika ada file yang tersimpan maka selanjutnya user dapat memilih file hex mana yang akan dihapus dengan tombol Select. Setelah user menekan tombol OK, selanjutnya AVR Programmer akan menghapus nomor index file dengan mengubahnya menjadi '0'. Setelah itu, AVR Programmer akan menghapus nama file yang telah tersimpan di Memory Eksternal. Lalu AVR Programmer akan membaca panjang data yang tersimpan di Memory Eksternal dan melakukan perulangan sebanyak panjang data itu untuk menghapus data yang tersimpan di Memory Eksternal.

### **Diagram Alir Program VB untuk Download File Hex dari PC ke AVR Programmer**

Pada proses ini, mula-mula user harus mengetikkan nama file yang diinginkan. Nama file itu nantinya digunakan untuk menamai file hex yang tersimpan di Memory Eksternal. Panjang nama file dibatasi maksimal 16 karakter. Jika user menekan

tombol Open File, akan muncul sebuah Common Dialog Box yang berfungsi untuk membuka file. Program kemudian akan membuka isi file yang telah dipilih tadi dan mengolah datanya. Isi file hex ini kemudian ditampilkan di sebuah listbox, yaitu List1.

Jika tombol Send ditekan maka program akan memeriksa jumlah file yang tersimpan di Memory Eksternal yang dikirim oleh AVR Programmer. Jika file hex belum penuh maka PC akan mengirimkan nama file yang telah diisikan user. Setelah itu, program akan menghitung panjang data file dan dikirimkan ke AVR Programmer. Baru diikuti dengan pengiriman data perbyte, sebanyak panjang data sebelumnya.

### **Diagram Alir Program VB untuk Upload dari AVR Programmer ke PC**

Untuk proses ini, mula-mula user harus memilih index file yang ingin dibaca dengan sebuah Combo Box. Setelah user menekan tombol Baca, program akan mengirimkan index file yang baru saja dipilih user ke AVR Programmer. Program kemudian akan memeriksa balasan dari AVR Programmer. Jika balasan dari AVR Programmer adalah 0, artinya di Memory Eksternal tidak ada file dengan index seperti yang telah dipilih user, maka program akan memunculkan sebuah Message Box dengan pesan bahwa data tidak ada. Jika balasan dari AVR Programmer bukan 0, berarti file yang diinginkan tersimpan di Memory Eksternal. Dan program bersiap-siap menerima data yang dikirimkan dari AVR Programmer. Urutan pengiriman data ini adalah nama file, panjang data, dan isi data hex-nya. Semua itu kemudian ditampilkan di sebuah listbox, yaitu List2 (Lihat Gambar 5).

Gambar 5 Diagram Alir untuk *Download* dari PC ke AVR Programmer dan *Upload* dari AVR Programmer ke PC

Gambar 6 Diagram Alir *Erasing Mode*

Gambar 7 Diagram Alir Program pada Mikrokontroler untuk Write dan Read dari/ke AVR Target

## Implementasi Perangkat Keras

Berikut ini adalah langkah implementasi modul Standalone AVR Programmer sebelum sistem bekerja. Pertama, masukkan jack adaptor dengan memberi supply 9 volt. Kedua, aktifkan daya pada modul dengan cara menggeser Switch daya ke kanan. Jika langkah 1 dan 2 dilakukan dengan benar maka layar LCD akan menyala dengan menampilkan tulisan “Standalone AVR Programmer”. Ketiga, untuk memilih menu, gunakan tombol SELECT (SW1). Ada tiga mode yang bisa dipilih, yaitu Programming Mode, PC Mode, dan Erase Mode. Keempat, jika sudah menemukan mode yang diinginkan, tekan tombol ENTER (SW2).

Pada Programming Mode: Gunakan tombol pilih untuk memilih file yang akan diprogramkan ke AVR Socket; Pastikan bahwa ketiga switch SW3-SW5 untuk jalur MOSI, MISO, dan SCK dalam keadaan tersambung, yaitu dengan menggeser switch ke kanan; Setelah menemukan file yang ingin diprogramkan ke AVR Socket, tekan tombol enter; Proses pemrograman ke AVR Socket memerlukan waktu beberapa saat. Jika pemrograman telah selesai, maka pada LCD akan muncul pemberitahuan bahwa “Pemrograman Selesai”.

Pada PC Mode: Untuk mencegah terjadinya kesalahan/gangguan dalam komunikasi dengan PC, pastikan ketiga switch SW3-SW5 MOSI, MISO, dan SCK dalam keadaan terbuka (geser switch SW3-SW5 ke kiri atau dalam keadaan Open); Pasangkan kabel STK 200 ke pin yang ada pada modul Standalone AVR Programmer. Sedangkan ujung yang satunya lagi dipasangkan ke port parallel pada PC.

Pada Erase Mode: Pastikan bahwa Memory Eksternal terpasang pada tempatnya; Gunakan

tombol pilih untuk memilih file yang akan diprogramkan ke AVR Socket; Setelah menemukan file yang ingin dihapus, tekan tombol enter; Proses penghapusan file hex memerlukan waktu beberapa saat. Jika penghapusan telah selesai maka pada LCD akan muncul pemberitahuan bahwa “Penghapusan Selesai”.

## Implementasi Perangkat Lunak

Program utama pada AVR Programmer dibuat dengan Codevision AVR, yang berbasis bahasa pemrograman C. Untuk komunikasi dengan PC, diperlukan bantuan program pendukung yang harus dijalankan pada komputer yang bersangkutan. Program pendukung itu dibuat dengan bahasa Visual Basic 6.0. Supaya dapat mengakses port parallel pada PC, harus digunakan library khusus, yaitu inport32.dll. File dll itu harus disimpan ke folder system32 pada direktori dari Sistem Operasi yang digunakan, misalnya di C:\WINDOWS\system32. Setelah file ini sudah disimpan, barulah program VB dapat dijalankan.

## Evaluasi Hasil Percobaan 1 Penyimpanan File Hex dari PC ke Memory Eksternal

Percobaan ini bertujuan untuk mengetahui waktu yang diperlukan untuk menyimpan file hex dari PC ke Memory Eksternal. Percobaan dilakukan dengan menghitung waktu yang diperlukan pada saat pengiriman file hex dari PC ke Memory Eksternal sampai selesai. Pada percobaan ini, digunakan file dengan ukuran yang berbeda-beda untuk melihat bagaimana kecepatan pengiriman file hex ke Memory Eksternal. Waktu tunda yang digunakan dalam percobaan ini adalah sebesar 4 ms (Lihat Tabel 1).

Tabel 1 Waktu Penyimpanan file Hex dari PC ke Memory Eksternal

Penyimpanan Ke -	Panjang Data (Byte)	Waktu Penyimpanan (s)	Status	Kecepatan Penyimpanan (Byte/s)
1	238 Bytes	1,01	Berhasil	235,6
2	1076 Bytes	4,83	Berhasil	222,77
3	1620 Bytes	6,25	Berhasil	259,2
4	2642 Bytes	11,06	Berhasil	238,87
5	5124 Bytes	20,47	Berhasil	250,31

**Evaluasi Hasil Percobaan 2 Perbandingan Pemrograman antara Standalone AVR Programmer dengan Codevision AVR:**

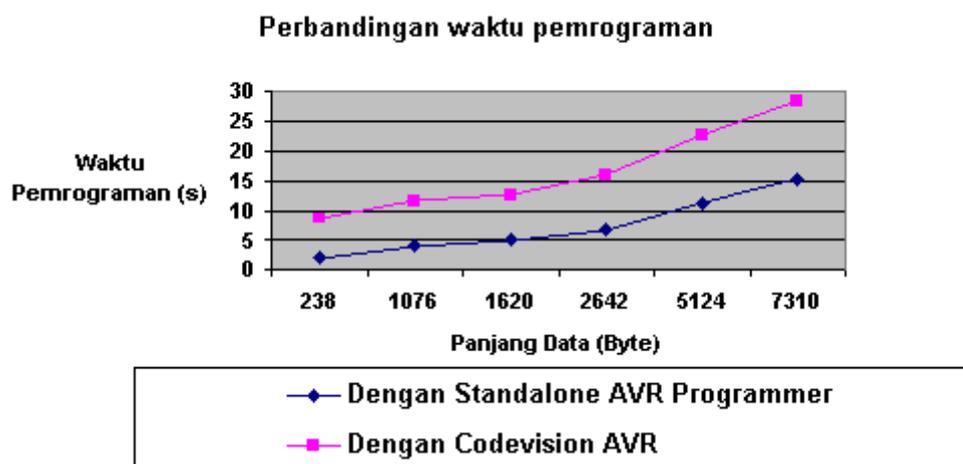
Percobaan ini dilakukan untuk membandingkan kecepatan pemrograman antara Standalone AVR Programmer dengan Codevision AVR. Pengujian pertama dilakukan Standalone AVR Programmer. File hex yang digunakan adalah file hex dengan ukuran yang berbeda-beda yang telah disimpan ke memory eksternal pada percobaan sebelumnya. Untuk menghitung lamanya pemrograman, digunakan stopwatch secara manual sedangkan kecepatan pemrograman diperoleh dengan membagi ukuran file dengan waktu pemrograman yang diperlukan.

Setelah seluruh data untuk Standalone AVR Programmer didapat, berikutnya adalah pengujian

pemrograman dengan menggunakan Codevision AVR. Codevision AVR adalah software khusus yang dikembangkan untuk pemrograman AVR. Selain dapat digunakan untuk membuat program untuk AVR berbasis bahasa C (semua program AVR yang digunakan dalam penelitian ini dibuat dengan Codevision AVR), Codevision AVR juga dapat digunakan untuk memprogram langsung AVR. Pemrograman dapat dilakukan dengan satu modul khusus untuk pemrograman AVR dan dengan kabel STK 200 untuk menghubungkannya ke PC. File hex yang digunakan dalam percobaan pemrograman ini adalah file hex yang sama dengan yang digunakan dalam percobaan pemrograman menggunakan Standalone AVR Programmer. Perbandingan pemrograman antara Standalone AVR Programmer dan Codevision AVR menggunakan file hex yang sama sehingga dapat dilakukan perbandingan antara keduanya (Lihat Tabel 2).

Tabel 2 Perbandingan Kecepatan Pemrograman antara Standalone AVR Programmer dengan Codevision AVR

Uji ke -	Ukuran File	Standalone AVR Programmer			Codevision AVR			Speed Standalone AVR Prog vs Codevision AVR	
		Time(s)	Speed (Bps)	Status	Time(s)	Speed (Bps)	Status	Speed(Bps)	%
1	238 Bytes	1,9	125,26	OK	8,57	27,77	OK	97,49	351%
2	1076 Bytes	4,08	263,75	OK	11,54	93,24	OK	170,51	183%
3	1620 Bytes	5,00	324	OK	12,52	129,40	OK	194,6	150%
4	2642 Bytes	6,55	403,36	OK	16,02	164,92	OK	238,44	145%
5	5124 Bytes	11,5	445,56	OK	22,51	227,63	OK	217,93	96%
6	7310 Bytes	15,47	472,53	OK	28,19	259,32	OK	213,21	82%

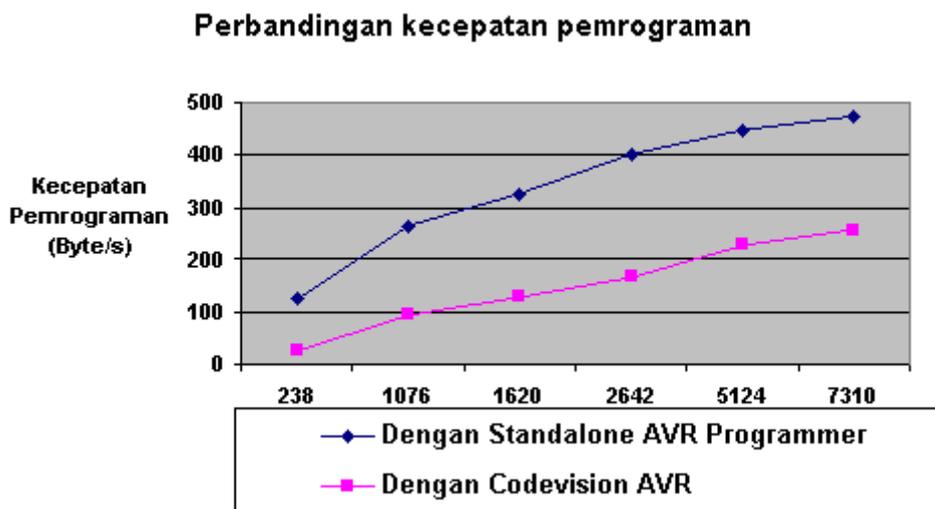


Gambar 8 Grafik Perbandingan Waktu Pemrograman antara Standalone AVR Programmer dengan Codevision AVR

Dari grafik pada Gambar 8 terlihat bahwa pemrograman menggunakan Standalone AVR Programmer memiliki waktu yang jauh lebih baik. Dengan ukuran data sebesar 238 byte, perbedaan waktu antara keduanya mencapai 6 detik lebih sedangkan dengan ukuran data sebesar 7310 byte, perbedaan waktu mencapai 12 detik lebih.

Salah satu faktor yang menyebabkan Codevision AVR lebih lambat adalah perbedaan proses pemrograman. Pada Standalone AVR Programmer ini, proses pemrograman dilakukan dengan mengirimkan perintah Erase Chip, diikuti dengan penulisan file hex yang diinginkan, baru dilakukan verifikasi data sedangkan pada Codevision AVR, setelah dilakukan Erase Chip, Codevision AVR akan melakukan Blank Checking, yaitu pemeriksaan apakah memory Flash pada AVR memang sudah kosong. Proses *Blank Checking* yang dilakukan oleh Codevision AVR ini memerlukan waktu yang sama untuk semua ukuran file hex, mulai dari 238 byte sampai 7310 byte, yaitu sekitar 6 detik. *Blank Checking* inilah yang menjadi salah satu faktor yang memperlambat Codevision AVR jika dibandingkan dengan Standalone AVR Programmer.

Grafik pada Gambar 9 menunjukkan perbandingan kecepatan pemrograman antara Standalone AVR Programmer dengan Codevision AVR. Pemrograman dengan ukuran data terkecil sampai data terbesar, tetap menunjukkan keunggulan Standalone AVR Programmer. Dengan ukuran data 238 byte, kecepatan penulisan Standalone AVR Programmer hampir lima kali lipat dari Codevision AVR sedangkan dengan ukuran data 7310 byte, kecepatan Standalone AVR Programmer sekitar 1,8 kali dari Codevision AVR. Total waktu yang diperlukan untuk melakukan pemrograman dengan Standalone AVR Programmer besarnya berbeda-beda, tergantung berapa kali proses pemrograman dilakukan. Untuk melakukan pemrograman, diperlukan dua proses, yaitu penyimpanan file hex ke memory eksternal dan pemrograman AVR socket menggunakan file hex yang telah disimpan ke memory eksternal. Waktu yang diperlukan untuk melakukan penyimpanan file hex ke memory eksternal relatif lebih lambat daripada waktu yang diperlukan untuk melakukan pemrograman AVR Socket. Akan tetapi, penyimpanan file hex ke memory eksternal hanya dilakukan sekali saja. Jadi, efisiensi kecepatan pada pemrograman dengan Standalone AVR Programmer ditentukan dari berapa kali pemrograman AVR Socket dilakukan.



Gambar 9 Grafik Perbandingan Kecepatan Pemrograman Antara Standalone AVR Programmer dengan Codevision AVR

Untuk perhitungan berikut ini, nilai yang digunakan adalah nilai yang didapat dari percobaan yang telah dilakukan sebelumnya. Untuk ukuran file hex yang terkecil, misalnya 238 byte yang digunakan dalam percobaan ini, jika pemrograman AVR Socket hanya dilakukan satu kali maka total waktu yang diperlukan adalah sebagai berikut.

Waktu total = waktu penyimpanan + waktu pemrograman  
 Waktu total = 1,01 + 1,9  
 Waktu total = 2,91 detik

Jadi, dengan ukuran data 238 byte, cukup dengan 1 kali pemrograman saja efisiensi waktunya sudah lebih besar daripada Codevision AVR yang memerlukan waktu 8,57 detik.

Untuk ukuran file hex sebesar 7310 byte:  
 Dengan Standalone AVR Programmer:

Waktu penyimpanan file hex ke memory eksternal: 28,98 detik

Waktu pemrograman AVR Socket : 15,47 detik

Dengan Codevision AVR: Waktu pemrograman: 28,19 detik

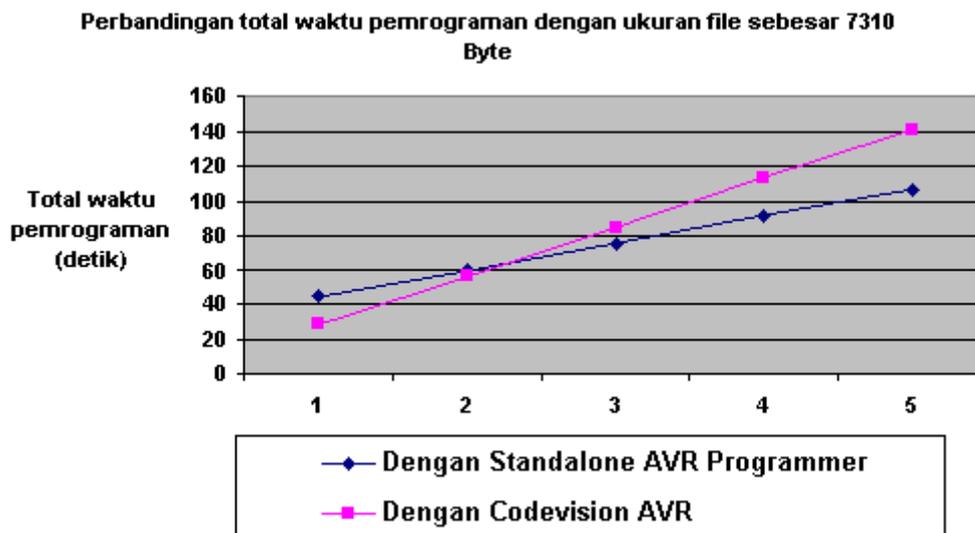
Dengan data ini, dibuat tabel perhitungan waktu total pemrograman yang diperlukan, mulai dari 1 kali pemrograman sampai 5 kali pemrograman (Lihat Tabel 3).

Tabel 3 Perbandingan Total waktu Pemrograman antara Standalone AVR Programmer dan Codevision AVR

# pemrograman	Standalone AVR Programmer (s)	Codevision AVR (s)	Selisih	
			Time (s)	%
1	44,45	28,19	16,26	-58%
2	59,92	56,38	3,54	-6%
3	75,39	84,57	-9,18	11%
4	90,86	112,76	-21,9	19%
5	106,33	140,95	-34,62	25%

Dari tabel hasil perhitungan tersebut, setelah dilakukan pemrograman sebanyak tiga kali atau lebih, total waktu pemrograman dengan Standalone AVR Programmer lebih cepat daripada dengan Codevision AVR. Hasil perhitungan itu menunjukkan efisiensi waktu yang didapat dalam pemrograman dengan

Standalone AVR Programmer akan terus meningkat seiring dengan semakin banyaknya proses pemrograman yang dilakukan (Lihat Gambar 10).



Gambar 10 Grafik Perbandingan Total Waktu Pemrograman

## PENUTUP

Dari penelitian ini, dapat diambil beberapa simpulan, antara lain proses penyimpanan file hex dari PC ke Memory Eksternal dapat dilakukan dengan baik. Dengan waktu tunda sebesar 4 ms tiap byte, proses penulisan ke Memory Eksternal dapat dilakukan dengan baik. Proses penulisan file hex ke AVR Socket dapat dilakukan dengan baik dan terakhir dengan Standalone AVR programmer, dapat dilakukan penghematan waktu, bila pemrograman dilakukan minimal 3 kali dibandingkan dengan Codevision AVR.

## DAFTAR PUSTAKA

- Anonymous. 2007. "AT24C512 Preliminary Complete," Atmel Corporation, Diakses dari [www.atmel.com/datasheet/AT24C512](http://www.atmel.com/datasheet/AT24C512).
- \_\_\_\_\_. 2007. "ATmega8515(L) Preliminary Complete," Atmel Corporation, Diakses dari [www.atmel.com/datasheet/ATMega8515](http://www.atmel.com/datasheet/ATMega8515).
- \_\_\_\_\_. 2007. "ATmega8535(L) Preliminary Complete," Atmel Corporation, Diakses dari [www.atmel.com/datasheet/ATMega8535](http://www.atmel.com/datasheet/ATMega8535).
- \_\_\_\_\_. 2007. "STK200 Preliminary Complete," Atmel Corporation, Diakses dari [www.atmel.com/product/STK200](http://www.atmel.com/product/STK200).
- \_\_\_\_\_. "Atmel AVR, en.wikipedia.org/avr," Diakses tahun 2007 dari wikipedia.com
- \_\_\_\_\_. "I2C, en.wikipedia.org/i2c," Diakses tahun 2007 dari wikipedia.com
- \_\_\_\_\_. "SPI, en.wikipedia.org/serial\_peripheral\_interface," Diakses tahun 2007 dari wikipedia.com
- Barnett, Cox and O'Cull. 2003. *Embedded C Programming and the Atmel AVR*. United States: Thomson Delmar Learning.