

Aplikasi Komunikasi Data Antara Kunci Elektronik dan PC Menggunakan Port Serial

R.Budiarianto Suryo Kusumo
Bidang Komputer P2 Informatika LIPI
Komplek LIPI Gd. 20 Lt.3 Cisitu 154 Bandung
Budiarianto@informatika.lipi.go.id

Abstrak

Penelitian ini bertujuan untuk mengembangkan fungsi AT89C52 yang tidak hanya sebagai kontrol untuk menggerakkan perangkat keras secara fisik tetapi juga mampu sebagai media komunikasi data. Dengan cara, membangun sistem aplikasi yang dapat mengirimkan dan menerima komunikasi data string dari Mikrokontroler ke PC maupun sebaliknya, menggunakan port serial dan pemrograman Visual basic 6.0

Dalam penelitian ini Diperoleh kesimpulan bahwa Aplikasi komunikasi data antara AT89C52 – PC yang telah dibuat, memiliki kemampuan yang sama dengan perangkat lunak komersial yang beredar dipasaran, Pengujian yang dilakukan yaitu dengan mengirimkan string dengan ukuran maksimum 64kb dari PC pada satu sesi kirim data dan diterima pada mikrokontroler dengan data yang sama dan ukuran yang sama pula, dan begitu juga sebaliknya.

Kata kunci Mikrokontroler, AT89C52, PC, Komunikasi, Data, serial, port

1. Pendahuluan

Seiring dengan perkembangan teknologi komputer khususnya pada *embedded system*, saat ini peranan *software* tidak dapat dipisahkan dari pesatnya perkembangan *hardware*. *Software* dapat berperan sebagai nyawa yang berfungsi untuk mengontrol dan menggerakkan segala aktivitas dari sebuah *hardware*, Tanpa *software*, *hardware* tidak bisa berfungsi maksimal.

Agar penelitian ini lebih terfokus pada tujuan yang ingin dicapai, maka terdapat beberapa batasan masalah yang akan menjadi pokok pembahasan yaitu: Perangkat lunak dibangun untuk komunikasi data antara PC dan Mikrokontroler dengan *port* serial, mikrokontroler harus melalui RS232 / Max 232, komunikasi data hanya berupa karakter *string* yang dapat di konversikan pada bilangan *hexadecimal*, dan pengembangan serta pembangunan perangkat lunak menggunakan *Visual basic 6.0*.

Pembahasan komunikasi data antara PC dan mikrokontroler sedikit sekali yang mengupas tentang komunikasi pengiriman dan penerimaan nilai *string* dari suatu karakter. Pada saat ini banyak yang hanya membahas tentang memerintah piranti keras untuk dapat digerakan secara fisik dengan menggunakan perangkat lunak. Untuk itu penelitian ini ditujukan untuk membangun sistem aplikasi yang dapat mengirimkan dan menerima komunikasi data string dari Mikrokontroler ke PC maupun sebaliknya.

Hasil pembuatan perangkat lunak ini diharapkan dapat memberikan sumbangan pengetahuan yang berarti dalam perkembangan ilmu sains terapan, khususnya pada bidang komputer yang dapat menjembatani antara bidang elektronika digital dan informatika, juga sebagai solusi perangkat lunak murah dan terjangkau bagi user, dan memberikan alternatif perangkat lunak aplikasi komunikasi data selain perangkat lunak komersial yang telah ada dipasaran termasuk *software hyperterminal* yang terdapat pada *system* operasi produk dari *Microsoft*. Perangkat lunak ini dapat

juga menyimpan data Identitas pengguna yang dikemudian dipakai sebagai autentifikasi untuk kode akses pada suatu ruangan, sehingga dapat mengatasi kebutuhan dari bidang Komputer Puslit Informatika LIPI yang saat ini berupaya tidak tergantung dengan *software non freeware* yang beredar dipasaran.

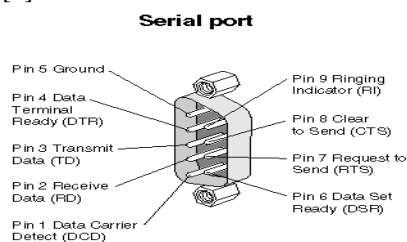
Penelitian sebelumnya yaitu *Prototype System Pengendali Pintu Untuk Keamanan Ruangan Berbasis Mikrokontroler AT89C51*[1]. Penelitian ini membahas tentang komunikasi menggunakan protokol *Multidrop*, karena memungkinkan setiap terminal berkomunikasi dengan terminal lain dalam jarak yang cukup jauh, prototype sistem bertujuan agar setiap ruangan terjaga keamanannya secara terpadu. Kemudian Penelitian berikutnya "Perancangan Alat Penyimpan Peran Berbasis AT89C52"[2]. Pada penelitian ini menunjukkan fungsi alat penyimpanan pesan, dirancang dengan menggunakan sistem minimum berbasis mikrokontroler AT89C52, yang berguna untuk menyimpan dan menampilkan pesan. Pesan diinputkan dan disimpan dalam memory RAM dan dibuka kembali oleh pengguna yang memiliki hak akses, User mempunyai hak akses. User memiliki hak akses jika mempunyai PIN.

Pada dasarnya metode penelitian yang dilakukan adalah studi pustaka yaitu mempelajari definisi-definisi dan metode-metode dan tatacara komunikasi rangkaian port serial serta RS232, serta fungsi-fungsi dari setiap pin yang ada pada mikrokontroler tersebut

2. Port Serial

Dibandingkan dengan menggunakan *port parallel*, penggunaan *port serial* terkesan lebih rumit. Berikut ini keuntungan-keuntungan penggunaan *port serial* dibanding dengan *port parallel* [3]. Pada komunikasi dengan kabel yang panjang, masalah kabel *loss* tidak akan menjadi masalah besar daripada menggunakan kabel *parallel*. *Port serial* mentransmisikan '1' pada level tegangan -3 volt sampai -25volt dan '0' pada level tegangan +3 volt sampai +25 volt.

Sedangkan port parallel mentransmisikan '0' pada level tegangan 0 volt serta '1' pada level tegangan 5 volt. Berikut tampilan Penampang dari *Serial port* pada computer PC [4]:



Gambar 1. Konektor serial DB-9 pada bagian belakang CPU

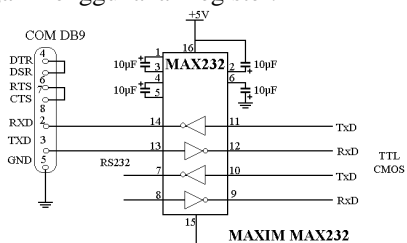
3. RS232/Max232

Nama lengkapnya adalah *EIA/TIA-232 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*. Meskipun nama resminya "menyeramkan", tapi standard ini hanya menyangkut komunikasi data antara komputer (*Data Terminal Equipment - DTE*) dengan alat-alat perlengkapan komputer (*Data Circuit-Terminating Equipment - DCE*) [3]. Keterangan mengenai fungsi saluran RS232/Max232 pada konektor DB-9 adalah sebagai berikut: received Line Signal Detect, dengan saluran ini DCE memberitahukan ke DTE bahwa pada terminal masukan ada data masuk, receive Data, Digunakan DTE menerima data dari DCE, transmit Data, digunakan DTE mengirimkan data ke DCE, data Terminal Ready, pada saluran ini DTE memberitahukan kesiapan terminalnya, signal Ground, saluran Ground. Ring Indicator, pada saluran ini DCE memberitahu ke DTE bahwa sebuah stasiun menghendaki hubungannya. Sedangkan Clear to send, dengan saluran ini DCE memberitahukan bahwa DTE boleh mulai mengirim data. Untuk request to Send. Dengan saluran ini DCE diminta mengirim data oleh DTE. Jika DCE Ready, Sinyal aktif pada saluran ini menunjukkan bahwa DCE sudah siap. Dari keterangan diatas

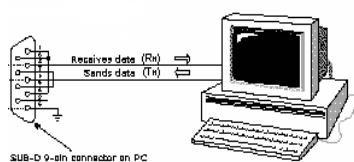
setiap saluran yang ada pada RS232/MAX 232 dapat dideskripsikan oleh gambar 2, berikut:

4. Karakteristik Sinyal RS232

Dalam standard RS232, tegangan antara +3 sampai +15 Volt pada *input Line Receiver* dianggap sebagai level tegangan '0', dan tegangan antara -3 sampai -15 Volt dianggap sebagai level tegangan '1'. Agar *output Line Driver* bisa dihubungkan dengan baik, tegangan *output Line Driver* berkisar antara +5 sampai +15 Volt untuk menyatakan level tegangan '0', dan berkisar antara -5 sampai -15 Volt untuk menyatakan level tegangan '1'[3]. Gambar 3, Merupakan ilustrasi pengiriman dan penerimaan data dari masing-masing piranti dengan menggunakan register.



Gambar 2. RS232 Diagram



Gambar3. koneksi dengan serial

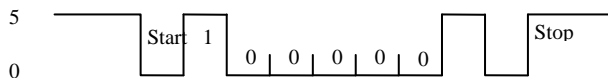
5. Atmel 89C52

AT89C52 adalah mikrokontroler keluaran Atmel dengan 8K byte Flash PEROM (*Programmable and Erasable Read Only Memory*), AT89C52 memiliki ruang alamat memory data dan program terpisah, pemisahan memory program dan data tersebut membolehkan memori data diakses

dengan alamat 8-bit, sehingga dapat dengan cepat dan mudah disimpan dan dimanipulasi oleh CPU 8-bit. Namun demikian, alamat memori data 16-bit bisa juga dihasilkan melalui register DPTR. Tetapi pada aplikasi ini hanya maksimal menggunakan memori data 8-bit [5].

6. Tata Cara Komunikasi Port Serial

Dalam komunikasi data serial dengan cara Asinkron, kecepatan pengiriman data (*baudrate*) dan *fase clock* pada sisi transmitter dan pada sisi receiver harus sinkron. Untuk itu diperlukan sinkronisasi antara transmitter dan receiver. Hal ini dilakukan oleh bit 'Start' dan bit 'Stop'. Ketika saluran transmisi dalam keadaan *idle*, *output* UART adalah dalam keadaan logika '1'. Ketika transmitter ingin mengirimkan data, *output* UART akan diset lebih dulu ke logika '0' untuk waktu satu bit. Sinyal ini pada receiver akan dikenali sebagai sinyal 'Start' yang digunakan untuk mensinkronkan *fase clocknya* sehingga sinkron dengan *fase clock transmitter* [6]. Berikut ini adalah contoh bagaimana pengiriman huruf dalam format ASCII (41 heksa/1000001) tanpa bit paritas yaitu huruf 'A'.

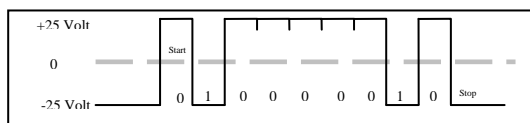


Gambar 4. Pengiriman huruf tanpa paritas

Kecepatan transmisi (*baud rate*) dapat dipilih bebas dalam rentang nilai yang umum digunakan adalah 110, 135, 150, 300, 600, 1200, 2400, dan 9600 (bit/detik). Dalam komunikasi data serial, *baud rate* dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama, selanjutnya harus ditentukan panjang data (maks 8 bit) paritas (genap, ganjil atau tanpa paritas) dan jumlah bit 'Stop'(1,1.5, atau 2 bit)[3].

7. Karakteristik Sinyal Port Serial

Standar sinyal serial RS232 memiliki ketentuan level tegangan sebagai berikut: Logika '1' disebut 'Mark' terletak antara -3volt hingga -25 Volt, Logika '0' disebut 'space' terletak antara +3Volt hingga +25Volt, daerah tegangan antara -3 Volt + 3 Volt adalah invalid level, yaitu level tegangan yang tidak memiliki level logika pasti sehingga harus dihindari. Demikian juga, lebih negatif dari -25 volt atau lebih positif dari + 25 volt juga harus dihindari karena tegangan tersebut dapat merusak *line Driver* pada saluran RS232 [3].



Gambar 5. Level tegangan RS232 pada pengiriman huruf 'A' tanpa bit paritas

Jika kecepatan transfer data dari DTE ke DCE (misal dari komputer ke Modem) lebih cepat dari pada transfer data dari DCE ke DCE (misal dari modem ke modem, cepat atau lambat kehilangan data akan terjadi karena *buffer* pada DCE akan mengalami *overflow*. Untuk itu diperlukan *flow control* untuk mengatasi masalah tersebut.

Dikenal dua macam *flow control* yaitu secara *software* dan *hardware*. *Flow control* secara *software* sering disebut Xon/Xoff *Flow Control* menggunakan karakter Xoff (tipikalnya karakter ASCII19) untuk melakukan *control* [6]. DCE akan mengirimkan Xoff untuk memberitahukan computer agar menghentikan pengiriman data jika *buffer* pada DCE telah penuh. Jika *buffer* telah kembali siap mengirimkan data, DCE akan mengirimkan karakter Xon ke *computer* dan *computer* akan mengirimkan data selanjutnya sampai data terkirim semua atau sampai *computer* menerima karakter Xoff kembali. Keuntungan *flow control* secara *software* ini adalah hanya membutuhkan kabel sedikit karena karakter *control* dikirimkan lewat saluran TX/RX. Akan tetapi, kecepatan pengiriman data menjadi lambat [3].

Untuk dapat mengaksesnya, kita dapat memasukan secara langsung ke dalam *source code* dengan hanya memasukan bilangan heksa yang tertera pada tabel.

8. Visual Basic

Visual Basic adalah perangkat lunak untuk menyusun program aplikasi yang bekerja dalam lingkungan system operasi *Windows*. Dengan *Visual Basic* kita bisa memanfaatkan kemampuan *Windows* secara optimal. Dengan kecanggihannya yang ditawarkan oleh *Visual Basic* kita akan merasakan begitu mudahnya menyusun program aplikasi dengan tampilan grafis yang menawan dalam waktu yang relatif singkat.

Tabel 1. Alamat dan lokasi bit pada register UART

Nama Pin	Nomor Pin pada BD-9	COM 1	COM 2	Bit	Arah
TXD	3	3FBh	2FBh	6	Output
DTR	4	3FCh	2FCh	0	Output
RTS	7	3FCh	2FCh	1	output
CTS	8	3FEh	2FEh	4	input
DSR	6	3FEh	2FEh	5	input
RI	9	3FEh	2FEh	6	input
DCD	1	3FEh	2FEh	7	input

9. Hasil penelitian dan pembahasan

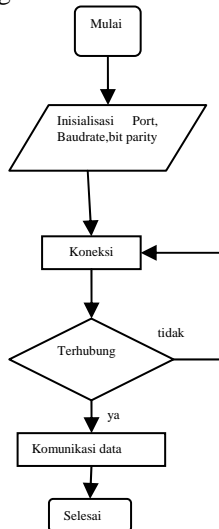
9.1. Pengaksesan dengan menggunakan kontrol ARComm

Kontrol ARComm menyediakan fasilitas komunikasi antara program aplikasi yang kita buat dengan *port* serial untuk mengirim atau menerima data melalui *port* serial. Setiap ingin menggunakan lebih dari satu *port* serial, kita juga harus menggunakan ARComm sebanyak *port serial* yang kita pakai. Untuk mengakses *port serial* kita dapat melakukannya secara langsung melalui *register* UART atau menggunakan kontrol MSComm dan ARComm yang telah disediakan pada *Visual Basic*.

Saluran yang digunakan UART untuk komunikasi baik untuk pengiriman maupun penerimaan data adalah saluran RxD dan saluran TxD serta saluran-saluran untuk control, yaitu saluran DCD, DSR, RTS, CTS, DTR, dan RI. Saluran-saluran ini ada yang sebagai *output* dan ada yang sebagai *input*. Kecuali saluran RxD, saluran-saluran ini dapat diakses secara langsung melalui *register* UART. Berikut adalah tabel alamat dan lokasi bit saluran tersebut pada *register* UART [7].

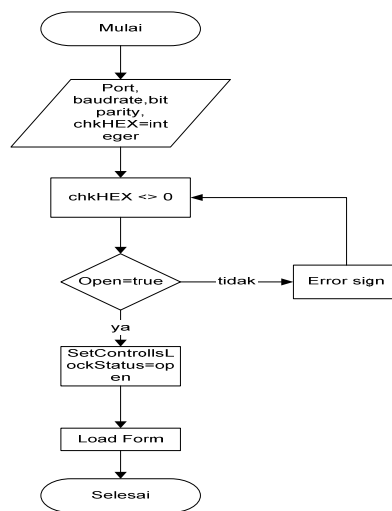
9.2. Perancangan Algoritma

Pada program aplikasi ini dibutuhkan jumlah kabel yang lebih sedikit, bisa hanya menggunakan tiga kabel, yaitu saluran *Transmit* data, saluran *Receive* data, dan saluran *Ground* (konfigurasi *Null Modem*). sebelum melakukan komunikasi antara PC dan Mikrokontroler, perlu adanya penyesuaian pada masing-masing perangkat, adapun yang harus disesuaikan yaitu kecepatan transfer (*baudrate*) serta pengaturan *port* yang akan dilewati untuk komunikasi antara dua perangkat. Gambar 6 adalah bagan alir system untuk menggambarkan keseluruhan langkah kerja dan system yang dibuat.



Gambar 6. Diagram alir system

Pada proses koneksi perangkat antara PC dan Mikrokontroler, perlu adanya inisialisasi dari ketersediaan *port* yang ada di PC, kemudian perlu juga penyesuaian seting dari transfer data. Informasi dari *port* pada PC dapat mudah diperoleh dengan ARComm yang tersedia pada VB, VB memudahkan dalam proses penyesuaiannya dan pemanggilan *port* yang terdapat pada PC. Gambar 7 merupakan Bagan alir yang menjelaskan langkah-langkah kerja dari proses koneksi perangkat keras dengan dijumpai piranti lunak.



Gambar 7. Diagram alir Koneksi system

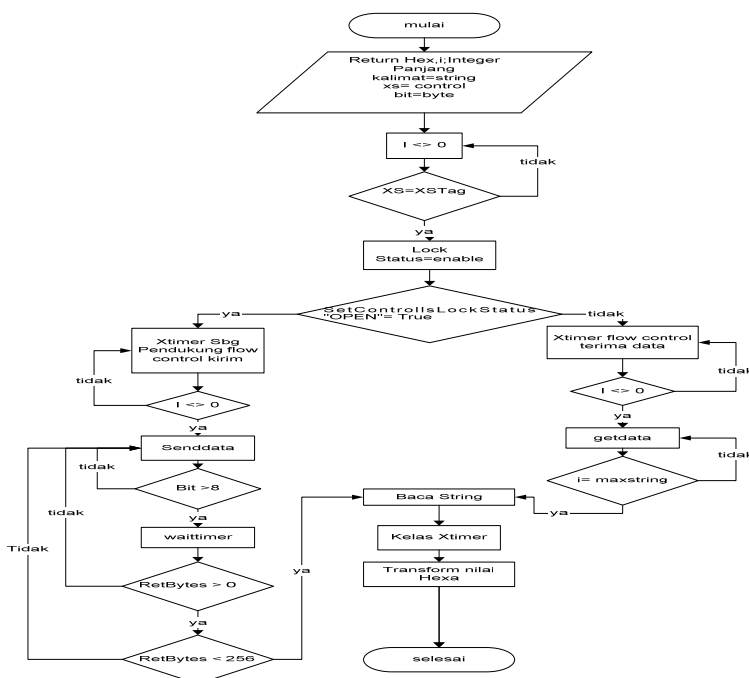
Terbuka dan tertutupnya sebuah *port* diatur oleh fungsi *control_lock_status* yang berperan sebagai pemberi informasi tentang keadaan *port* yang digunakan sebagai saluran komunikasi data.

Proses komunikasi data adalah proses utama dari program aplikasi ini, proses komunikasi dimulai dari inisialisasi masukan data yang akan dikomunikasikan antar perangkat, nilai Hexadesimal sangat berperan dalam melakukan komunikasi, hampir keseluruhan proses melibatkan nilai *hexa*. Proses komunikasi data dibagi menjadi dua bagian, bagian pertama yaitu proses pengiriman data.

Pada proses pengiriman data diatur dengan komponen xtimer berfungsi sebagai *buffer over flow control (Control buffer)* jika terjadi kelebihan pengiriman data. Untuk diketahui bahwa Mikrokontroler hanya mampu menerima 8 bit data untuk sekali proses pengiriman, untuk memulai pengiriman berikutnya, mikrokontroler harus menunggu jeda 20 μ s. untuk mengatur jeda tersebut diperlukan komponen xtimer yang tersedia pada VB.

Akan tetapi hal ini tidak berlaku untuk proses pengiriman dari mikrokontroler ke

PC, karena PC memungkinkan untuk menerima data lebih besar dari 8bit/detik. Pada proses penerimaan data, xtimer digunakan untuk mengatur agar data antara proses pengiriman dan proses pembacaan tidak tumpang tindih, dan tidak terjadi kesalahan penerimaan data pada setiap string yang diterima PC. Langkah kerja dapat terlihat dari bagan alir komunikasi system dibawah ini. Seperti yang ditampilkan pada Gambar 8 berikut ini:



Gambar 8. Diagram alir komunikasi system

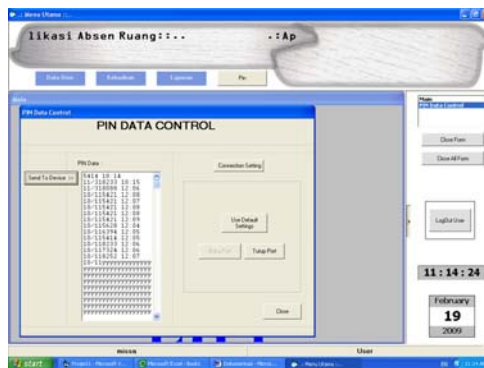
9.3. Hasil Penelitian

Pengujian perangkat lunak ini dengan menggunakan *downloader AT89C52* dan Kunci elektronik, data yang dikirim berupa *char 'T'* melalui Aplikasi komunikasi data antara AT89C52 dengan PC Menggunakan serial port. Data yang dikirimkan merupakan trigger untuk mengambil semua data yang ada didalam memory AT89C52, yang kemudian disimpan ke dalam memory PC. Begitu juga dengan mengirimkan data

maksimal yaitu dengan mengirimkan data sebesar 64kb pada satu sesi kirim, baik dari PC ke Mikrokontroler maupun sebaliknya dan diterima dengan data yang sama dan identik. Program aplikasi juga diujikan dengan perangkat kunci elektronik yang memiliki LCD *display* terintegrasi, pada layar display pun muncul *string* yang sama dengan *string* yang dikirimkan dari PC. Gambar 9 adalah contoh dari *downloader* dan *Prototype elektrik lock* yang akan

dipakai di ruang Bidang Komputer Puslit Informatika LIPI.

Gambar 10. menunjukkan tampilan antar muka dari program Aplikasi komunikasi data antara AT89C52 - PC Menggunakan serial *port*. Pengaturan dilakukan dengan sangat mudah hanya memilih *port* yang akan digunakan dan mengisikan jumlah kecepatan yang dibutuhkan dari mikrokontroler untuk memulai komunikasi. Kolom untuk pengaturan ditempatkan pada sisi kanan dari tampilan piranti lunak. Untuk sisi kiri ditujukan bagi proses input data dan proses komunikasi data.



Gambar 10. Tampilan Antarmuka Software

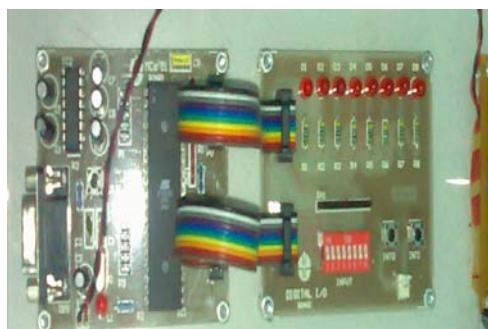
10. Diskusi dan Perbandingan dengan HyperTerminal

Pada *Hyperterminal* dapat diketahui proses pengaturan dalam koneksi perangkat melewati beberapa tahapan yang cukup panjang dan merepotkan bagi pengguna yang awam dengan masalah interkoneksi *hardware*. Tahapan yang harus dilewati yaitu konfirmasi dari lokasi berikut region perangkat lunak tersebut digunakan, kemudian masuk pada pengaturan modem yang akan dilewati sebagai *comm port*, lalu tahap selanjutnya koneksi harus dideskripsikan dengan sebuah nama yang diinputkan oleh *user* dan terakhir yaitu melakukan konfirmasi bahwa *hardware* dikenali sebagai modem yang akan dipanggil berdasarkan kode area.

Hal ini sangat jauh berbeda dengan Aplikasi komunikasi data antara AT89C52-PC Menggunakan serial *port*. Aplikasi ini hanya perlu pengaturan gerbang port yang akan dipakai, serta pengaturan standar yaitu pengaturan *baudrate* dan *parity*. Selain itu kemampuan piranti ini kurang lebih sama dengan kemampuan *hyperterminal* yaitu mengirimkan kode ASCII ke mikrokontroler dan sebaliknya.

11. KESIMPULAN

Berdasarkan hasil pengujian dari Aplikasi komunikasi data antara AT89C52 dengan PC Menggunakan serial *port*, maka dapat diambil kesimpulan, perangkat lunak yang telah dibuat, layak untuk dipergunakan sebagai sarana penghubung antar perangkat.



A



B

Gambar 9 .A. Downloader B.Elektrik lock

Software Aplikasi komunikasi data antara AT89C52-PC *serial port* menggunakan algoritma yang sangat sederhana, mudah dikembangkan kembali serta murah dalam sudut pandang harga, walaupun masih terdapat kekurangan dari segi tampilan dan instrument penunjang seperti jendela konfirmasi apabila koneksi telah berhasil.

Dari hasil pengujian yang dilakukan, masih banyak kelemahan yang ditemui. Antara lain adalah piranti ini tidak sepenuhnya *open source* dan tidak tergolong *freeware* karena dibangun dengan menggunakan VB yang masih dalam platform *Microsoft*, untuk perkembangan selanjutnya diharapkan *software* ini dikembangkan dalam *platform* yang benar-benar *open source* yaitu *JAVA*.

12. Ucapan terimakasih

Terimakasih disampaikan kepada pak Herlan. S.T. sebagai pembimbing materi pemrograman elektronika digital pada Puslit Informatika Bandung dan Kepada Prof.Riset. Sunartoto Gunadi. M.Eng. Sebagai pembimbing penulisan Karya Tulis Ilmiah.

13. Daftar Pustaka

- [1] Jaya, Edi Santoso; Yoannatan, Sugiyanto; Okvianes, Maulana; Andi., "Prototype Untuk Keamanan Ruangan Berbasis Mikrokontroler AT89C51", Laporan Teknis Berkala, 8(1) januari 2000,1-12.
- [2] Luka, Samuel; Hidayat, Chaerul; Suka, Martin; Budiharto., "Perancangan Alat Penyimpan Pesan Berbasis AT89C52", 1999., Laporan Teknis Berkala, 7(2) September, 1999., 23-41.
- [3] Praselia, Retna ; Widodo, Catur Edi., "Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0", Penerbit Andi, Yogyakarta, 2004.
- [4] Budioko, Totok., "Belajar Dengan Mudah dan Cepat Pemrograman Bahasa C dengan SDCC (Small Device C Compiler) Pada Mikrokontroler AT89X051/ AT89C51/52 Teori Simulasi

dan Aplikasi"., Gava Media, Yogyakarta, 2005.

- [5] Budiharto, Widodo., "Panduan Praktikum Mikrokontroler AVR ATMEGA16, Panduan Utama untuk Pelajar dan Penghobi", PT. Elex Media Komputindo, Kelompok Gramedia, Jakarta, 2008.
- [6] Predko, Myke., "Programming and Customizing the 8051 Microcontroller", McGraw-Hill Companies, United States, 1999.
- [7] Eko Putra, Agfianto., "Belajar Mikrokontroler AT89C51/52/55 Teori dan Aplikasi edisi 2", Penerbit Gava Media, Yogyakarta, 2004.