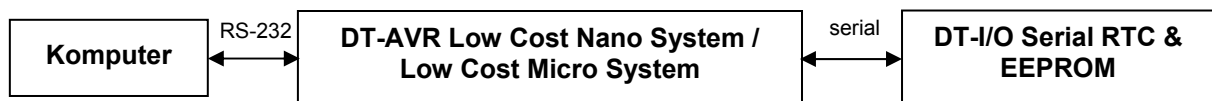


Aplikasi kali ini akan memberikan contoh rutin-rutin untuk DT-I/O Serial RTC & EEPROM dalam bahasa pemrograman C dengan menggunakan CodeVisionAVR© yang dewasa ini telah banyak digunakan. DT-AVR Low Cost Series digunakan agar aplikasi ini menjadi lebih handal. Rutin-rutin dalam AN ini dapat digunakan untuk aplikasi DT-AVR Low Cost Series (atau mikrokontroler AVR pada umumnya) yang dihubungkan dengan DT-I/O Serial RTC & EEPROM atau RTC DS1302 atau serial EEPROM AT93C66, misalnya: data logger, sistem pengaturan otomatis terjadwal, dll.

Komponen yang diperlukan:

- 1 DT-AVR Low Cost Nano System / Low Cost Micro System
- 1 DT-I/O Serial RTC & EEPROM

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN88

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-AVR Low Cost Nano System / Low Cost Micro System	DT-I/O Serial RTC & EEPROM
GND	GND (J3)
+5VDC	+5V (J3)
PB.0*	CLK (pin9 PortC & Port1)
PB.1*	D (pin10 PortC & Port1)
PB.2*	CS (pin12 PortC & Port1)

* pin ini tidak mutlak dan dapat diganti pin lain tetapi harus mengubah konfigurasi pin pada program

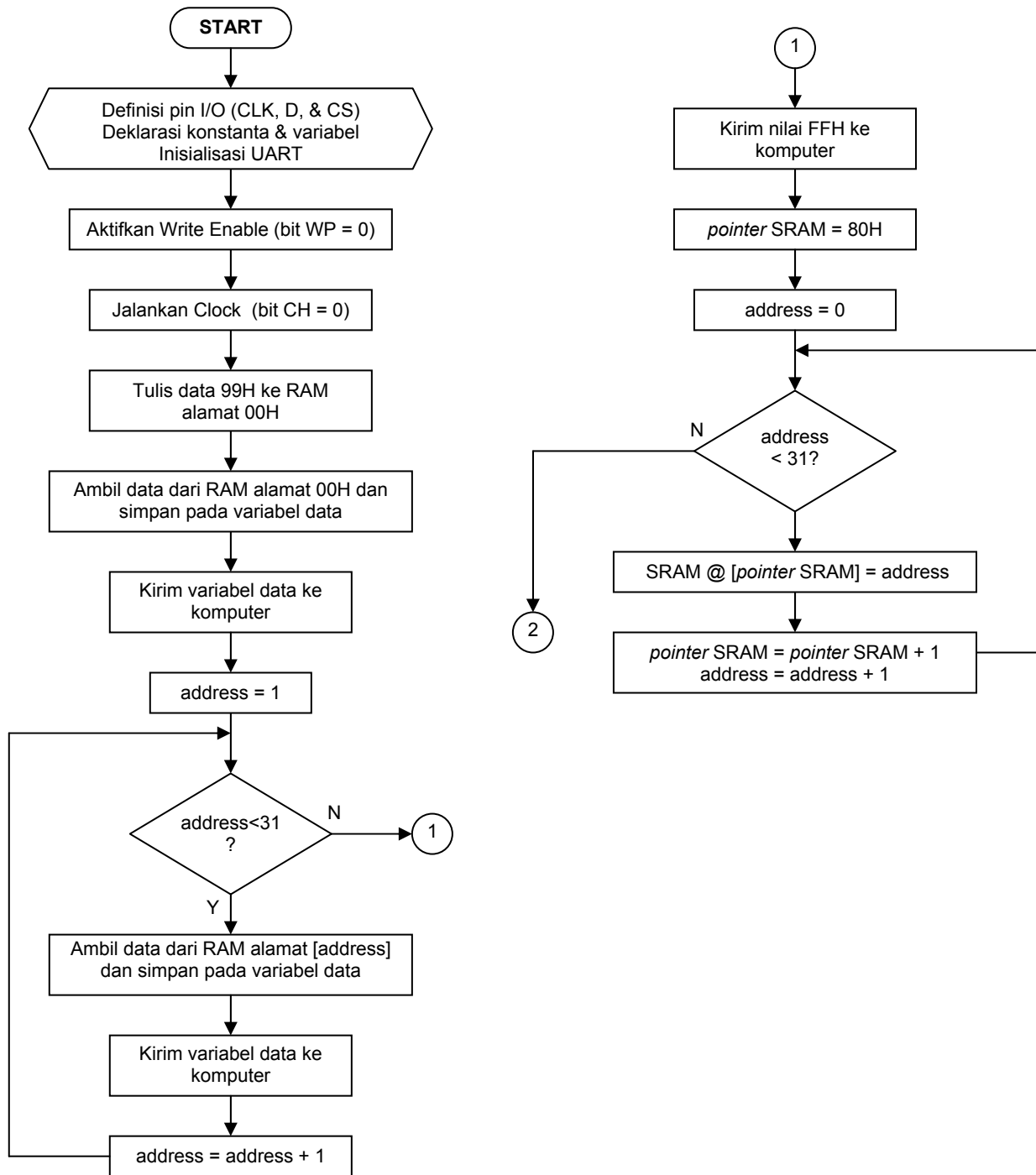
Tabel 1

Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan DT-I/O Serial RTC & EEPROM

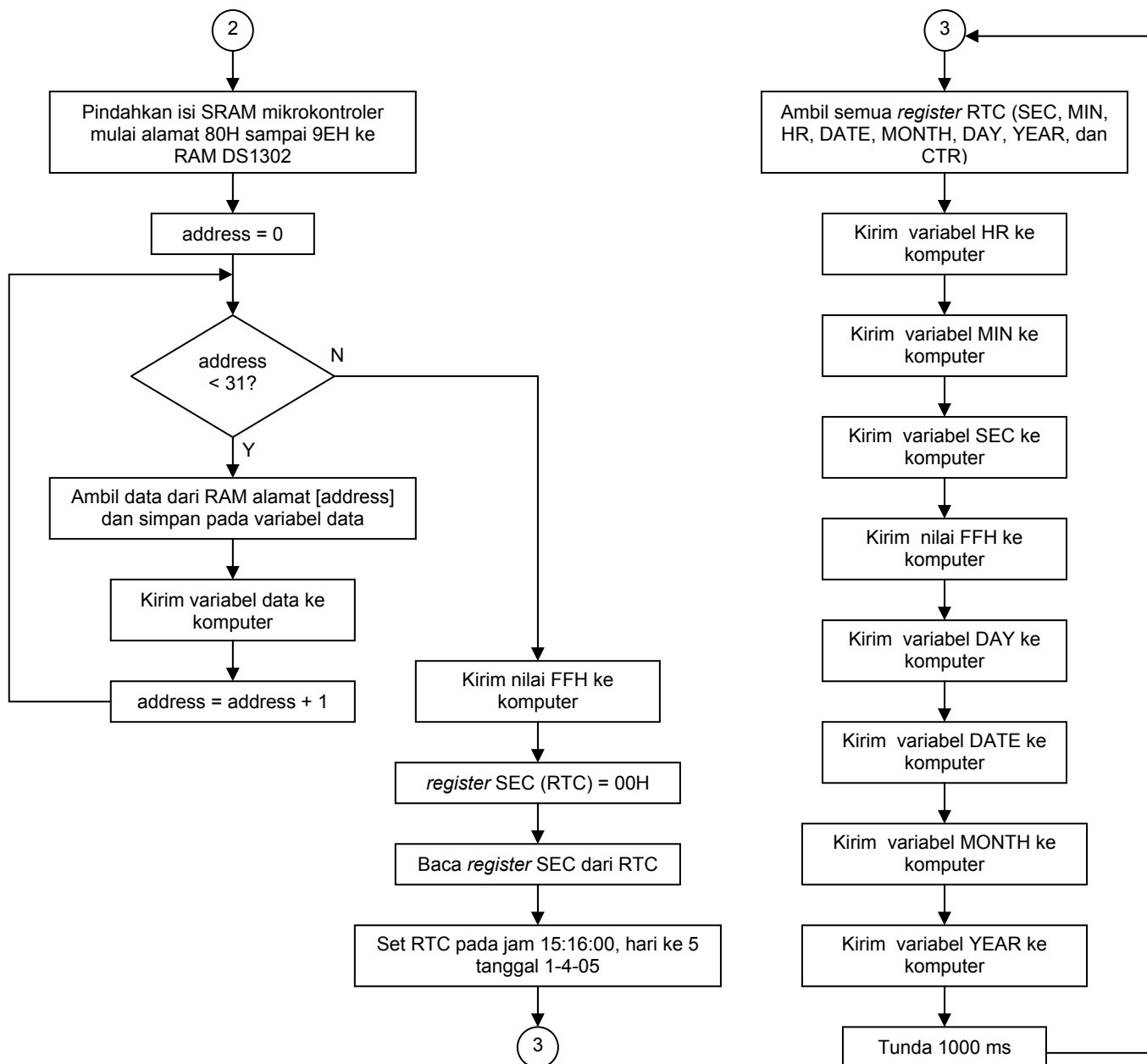
Gunakan kabel serial DT-AVR Low Cost Nano System / Low Cost Micro System untuk menghubungkan modul ke komputer. Kabel ini digunakan untuk mengirimkan data ke komputer dan ditampilkan pada layar monitor dengan bantuan program Terminal© atau *tool* Terminal pada CodeVisionAVR©. Konfigurasi UART pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit stop, dan tanpa *flow control*.

Setelah semua rangkaian terhubung dengan tepat, berilah catu daya pada rangkaian dan bukalah proyek SerialRTCEEPROM.prj menggunakan CodeVisionAVR© (versi 1.24.5 atau lebih tinggi). Lalu kopikan *listing* "Program RTC.c" atau "Program EEPROM.c" ke Program.c pada proyek tersebut. Gunakan menu Project → Make atau tekan Shift+F9 untuk membuat *image file*. Kemudian programlah hasilnya (program.rom atau program.hex) ke dalam DT-AVR Low Cost Nano System / Low Cost Micro System dengan DT-HiQ AVR In System Programmer atau AVR ISP programmer lain dengan konektor 10 pin standar Atmel.

Flowchart program "Program RTC.c" adalah sebagai berikut:



Gambar 2
Flowchart Program "Program RTC.c" (bagian 1)



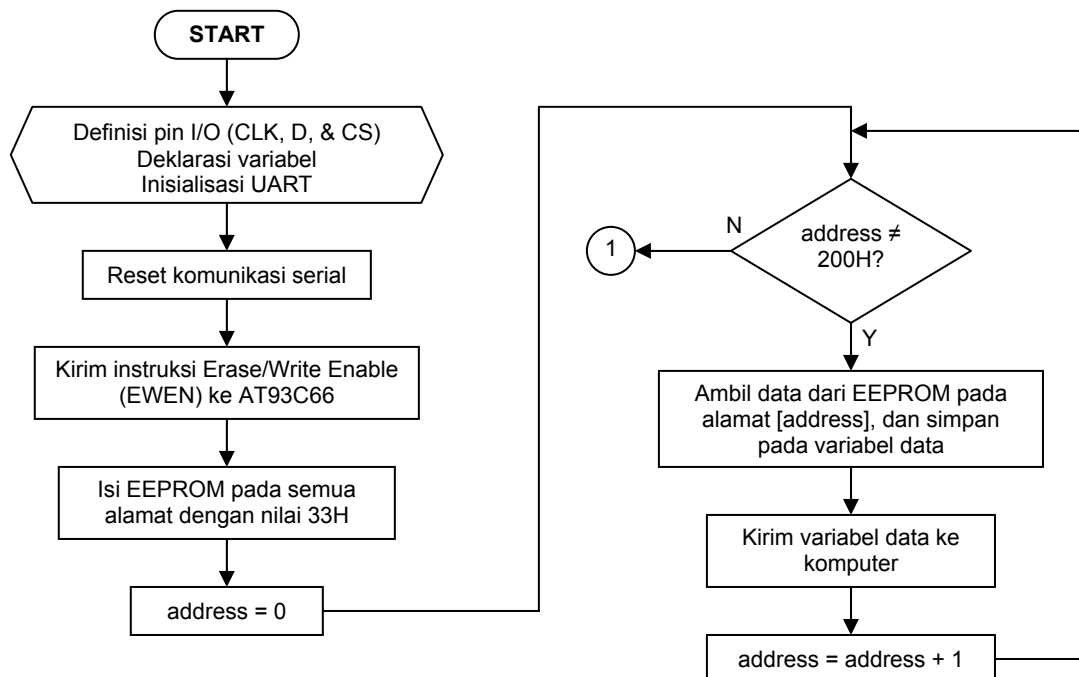
Gambar 3
Flowchart program “Program RTC.c” (bagian 2)

Program utama (Program RTC.C) akan diproses sebagai berikut:

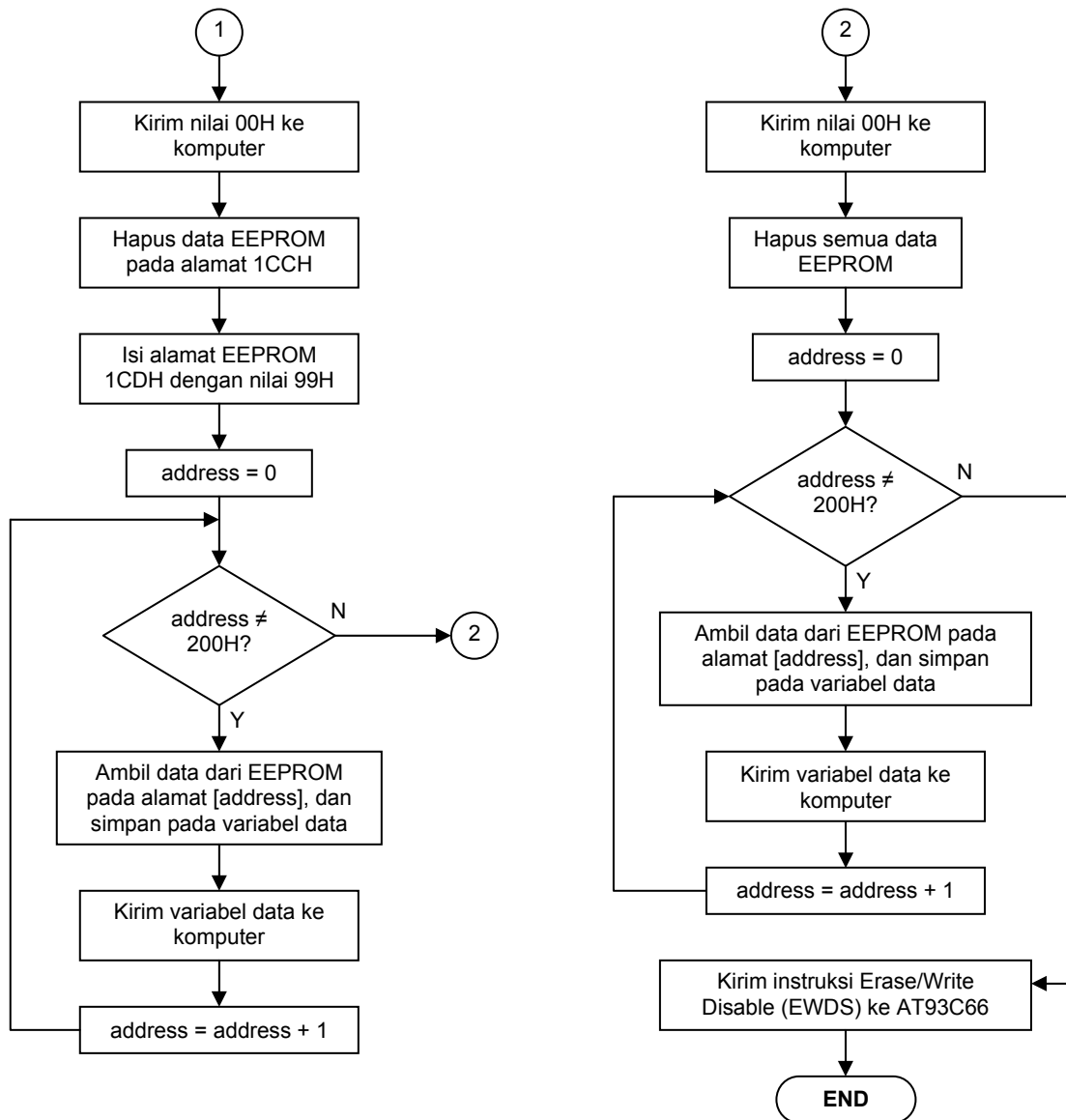
1. Proses yang pertama kali dilakukan adalah menentukan definisi pin I/O CLK, D dan CS untuk komunikasi serial dengan RTC DS1302. Dan deklarasi variabel:
 - SEC, MIN, HR, DATE, MONTH, DAY, YEAR, dan CTR sebagai *buffer* pembacaan / penulisan *register* DS1302,
 - address untuk kegunaan umum,
 - data sebagai *buffer* pembacaan / penulisan RAM DS1302.
2. Proses berikutnya yaitu inisialisasi komunikasi serial UART yaitu pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, dan 1 bit stop.
3. Setelah proses inisialisasi selesai, berikutnya program akan memberi logika '0' pada bit WP dalam *register* CONTROL DS1302 dengan menggunakan prosedur WREN. Jika bit WP (*Write Protect*) bernilai '0' maka menulis (*write*) data ke DS1302 diperbolehkan. Kemudian program akan memberi logika '0' pada bit CH dalam *register* SEC DS1302 dengan menggunakan prosedur RUNCLK. Jika bit CH (*Clock Halt*) bernilai '0' maka osilator *clock* pada DS1302 akan aktif.

4. Setelah itu program akan menulis data sebesar 99H ke RAM DS1302 pada alamat 00H menggunakan prosedur BYTEWRRAM. Kemudian membacanya lagi (dari RAM DS1302 alamat 00H) dengan menggunakan fungsi BYTERDRAM dan mengirimkan hasilnya ke komputer.
5. Proses berikutnya yaitu membaca data (dengan menggunakan fungsi BYTERDRAM) satu-persatu dari RAM DS1302 mulai alamat 1 sampai 30 dan mengirimkan hasil pembacaan satu-persatu ke komputer. Setelah proses pembacaan data RAM DS1302 selesai, program mengirimkan nilai FFH ke komputer.
6. Program akan mengisi memori SRAM mikrokontroler mulai alamat 80H sampai 9EH dengan nilai (data) mulai 00H sampai 1EH. Setelah proses pengisian SRAM mikrokontroler selesai, program akan memindahkan isi SRAM mikrokontroler mulai alamat 80H sampai 9EH ke RAM DS1302 dengan menggunakan prosedur BURSTWRRAM. Sehingga data pada RAM DS1302 mulai alamat 0 sampai 30 berisi nilai mulai 00H sampai 1EH.
7. Kemudian program akan membaca data (menggunakan fungsi BYTERDRAM) satu-persatu dari RAM DS1302 mulai alamat 0 sampai 30 dan mengirimkan hasil pembacaan satu-persatu ke komputer. Setelah proses ini selesai, program mengirimkan nilai FFH ke komputer.
8. Selanjutnya program akan mengisi *register* SEC DS1302 dengan nilai 00H dengan menggunakan prosedur BYTEWRCLKREG. Kemudian membacanya lagi dengan menggunakan fungsi BYTERDCLKREG.
9. Kemudian mengatur *register* RTC pada jam 15:16:00 dan pada hari ke 5 tanggal 1-04-05 dengan menggunakan prosedur BURSTWRCLKREG. Sebelum memanggil prosedur BURSTWRCLKREG, variabel SEC, MIN, HR, DAY, DATE, MONTH, dan YEAR pada RTC diisi dengan nilai yaitu SEC = 00H, MIN = 16H, HR = 15H, DAY = 05H, DATE = 01H, MONTH = 04H dan YEAR = 05H.
10. Proses yang terakhir adalah membaca semua *register* RTC DS1302 dengan menggunakan prosedur BURSTRDCLKREG dan mengirimkan hasil pembacaan satu-persatu ke komputer mulai variabel HR, MIN, SEC, nilai FFH, DAY, DATE, MONTH dan terakhir variabel YEAR. Setelah menunda selama 1000 ms, program mengulangi langkah 10 ini secara terus menerus.

Flowchart program “Program EEPROM.C” adalah sebagai berikut:



Gambar 4
Flowchart Program “Program EEPROM.c” (bagian 1)



Gambar 5
Flowchart Program “Program EEPROM.c” (bagian 2)

Program utama (Program EEPROM.C) akan diproses sebagai berikut:

1. Proses yang pertama kali dilakukan adalah menentukan definisi pin I/O CLK, D dan CS untuk komunikasi serial dengan AT93C66 (EEPROM). Dan deklarasi variabel:
 - o data sebagai buffer penulisan / pembacaan EEPROM.
 - o address sebagai indeks alamat EEPROM.
2. Proses berikutnya adalah inisialisasi komunikasi serial UART pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, dan 1 bit stop. Kemudian program akan me-reset komunikasi serial dengan memberikan logika ‘1’ pada pin CS selama 1 us menggunakan prosedur reset.
3. Selanjutnya program akan mengirimkan instruksi Erase/Write Enable (menggunakan prosedur EWEN) agar AT93C66 dapat ditulis, sebab AT93C66 selalu dalam kondisi *erase/write disable* saat pertama kali diberi tegangan catu.
4. Setelah itu program akan menulis data 33H ke semua alamat EEPROM dengan menggunakan prosedur WRAL.
5. Proses berikutnya yaitu membaca data (dengan menggunakan fungsi READEE) satu-persatu dari EEPROM mulai alamat 000H sampai 1FFH dan mengirimkan hasil pembacaan satu-persatu ke komputer. Sebagai batas akhir pembacaan data EEPROM, dikirimkan nilai 00H ke komputer.
6. Program menghapus data EEPROM pada alamat 1CCH dengan menggunakan prosedur ERASEEE. Selanjutnya program akan menulis data 99H pada alamat 1CDH ke EEPROM dengan menggunakan prosedur WRITEEE.

7. Kemudian program akan membaca data (dengan menggunakan fungsi READEE) satu-persatu dari EEPROM mulai alamat 000H sampai 1FFH dan mengirimkan hasil pembacaan satu-persatu ke komputer. Sebagai tanda batas akhir pembacaan data EEPROM, dikirimkan nilai 00H ke komputer.
8. Selanjutnya program akan menghapus semua isi data EEPROM dengan menggunakan prosedur ERAL.
9. Proses yang terakhir adalah membaca data (dengan menggunakan fungsi READEE) satu-persatu dari EEPROM mulai alamat 000H sampai 1FFH dan mengirimkan hasil pembacaan satu-persatu ke komputer.
10. Setelah itu program mengirimkan instruksi Erase/Write Disable ke AT93C66 (menggunakan prosedur EWDS) sehingga AT93C66 tidak dapat diisi atau dihapus.

Listing program "Program RTC.C" dan "Program EEPROM.C" terdapat pada **AN88.ZIP**.

Selamat berinovasi!

CodeVisionAVR is copyright by Pavel Haiduc, HP InfoTech s.r.l.
Terminal is copyright by Bray++.