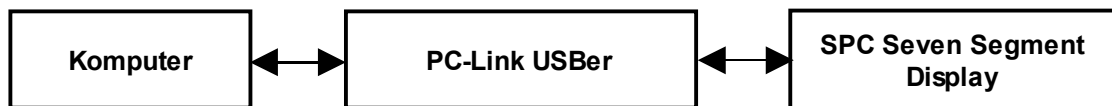


Sebuah contoh lagi mengenai penggunaan modul PC-Link USBer dengan menggunakan bahasa pemrograman Borland® Delphi™. Aplikasi AN126 ini akan memaparkan cara menggunakan PC-Link USBer sebagai jalur komunikasi SPI untuk mengendalikan SPC Seven Segment Display.

Modul yang diperlukan dalam aplikasi ini adalah:

- 1 PC-Link USBer,
- 1 SPC Seven Segment Display.

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1  
Blok Diagram AN126

Hubungan antara modul-modul tersebut adalah sebagai berikut:

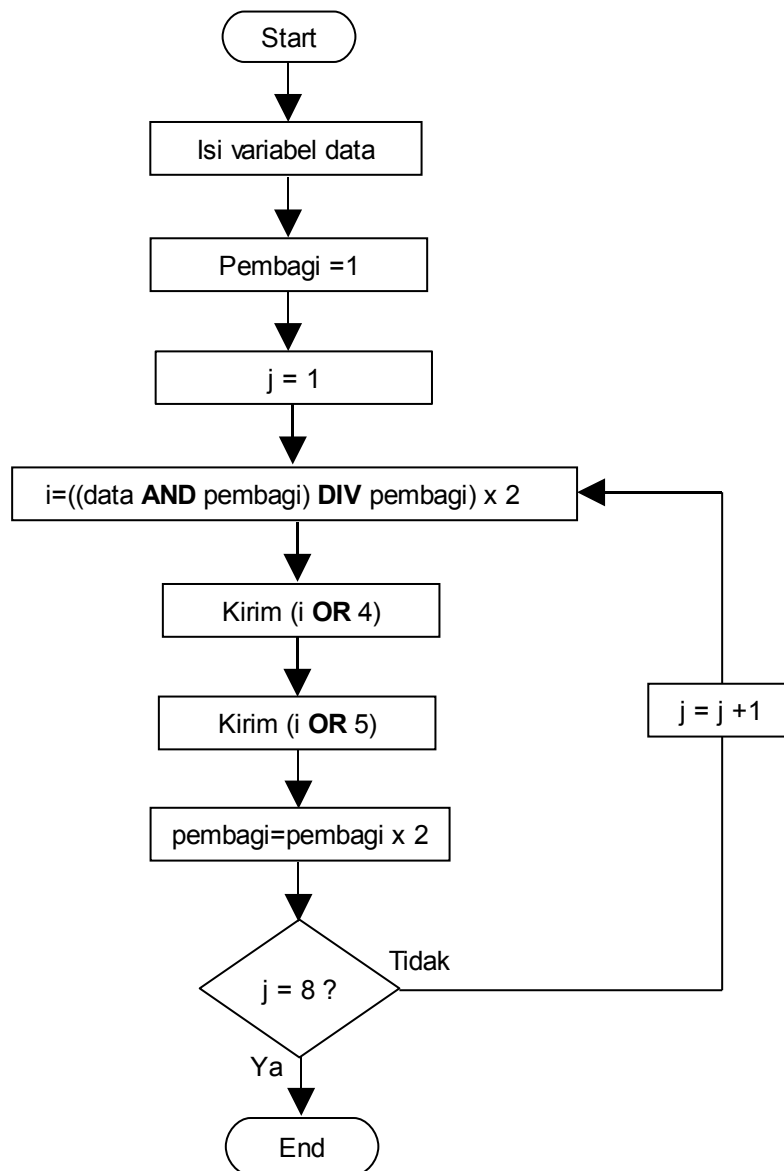
PC- Link USBer	SPC Seven Segment Display
TXD (J2 Pin 3)*	SCK (J12 Pin 7 )
RXD (J2 Pin 4)*	MOSI (J12 Pin 8)
RTS (J2 Pin 5)*	CLR (J12 Pin 9 )
CTS (J2 Pin 6)*	MISO (J12 Pin 10)
GND (J2 Pin 1)	GND (J13 Pin 1)

\* Pin ini tidak mutlak dan dapat diganti pin lain dengan cara mengubah program

Tabel 1  
Hubungan PC-Link USBer dengan SPC Seven Segment Display

Hubungkan PC-Link USBer ke port USB komputer menggunakan kabel USB yang tersedia pada paket penjualan PC-Link USBer. Hubungkan pin 1 dengan pin 2, pin 3 dengan pin 4, dan pin 5 dengan pin 6 pada jumper S2 yang terdapat pada SPC Seven Segment Display agar modul ini dapat berkomunikasi dengan antarmuka SPI. Pastikan rangkaian dan sumber tegangan telah terhubung dengan benar pada kedua modul, PC-Link USBer memerlukan input tegangan 5VDC pada J12 pin 2 (VCC). Setelah itu program **Seven.exe** dapat dijalankan.

**F**lowchart prosedur pengiriman data (TForm1.write\_byte) dengan antarmuka SPI menggunakan PC-Link USBer:

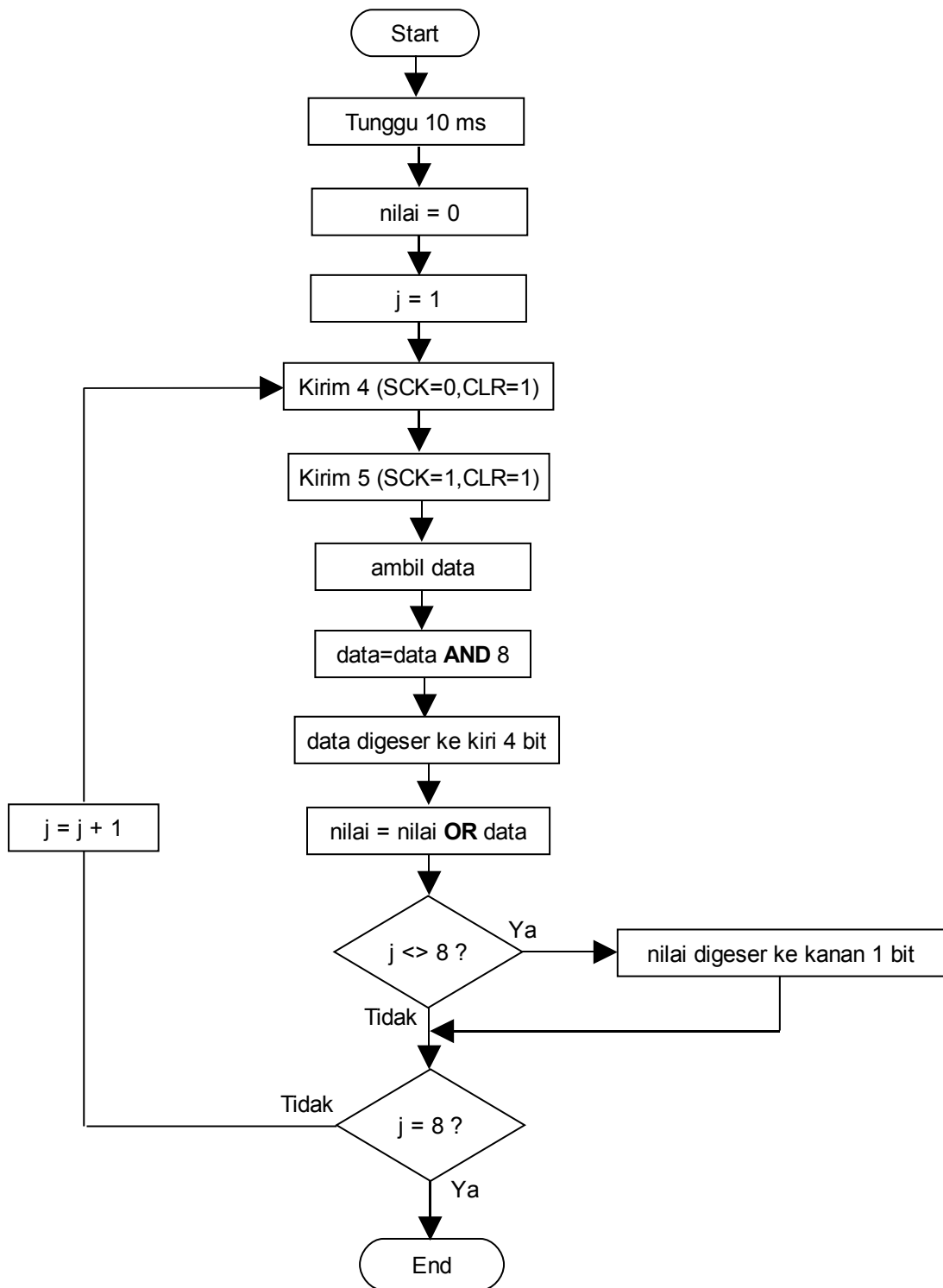


**Gambar 2**  
**Flowchart Prosedur Pengiriman Data dengan Antarmuka SPI menggunakan PC-Link USBer**

Cara kerja prosedur pengiriman data (Tform1.write\_byte) dengan antarmuka SPI menggunakan PC-Link USBer adalah sebagai berikut:

1. Pertama-tama nilai data yang akan dikirimkan dimasukkan pada variabel data.
2. Kemudian variabel pembagi diberi nilai 1.
3. Data yang akan dikirimkan adalah selebar 8 bit, oleh karena itu diperlukan proses perulangan sebanyak 8 kali yang dilakukan menggunakan instruksi **for-to-do** dengan indeks variabel j.
4. Kemudian ambil nilai bit dari variabel data dimulai dari LSB.
5. Bit data pada pin MOSI dikirimkan bersamaan dengan perubahan sinyal CLK dari logika *low* ke logika *high*. Sedangkan pin CLR selalu berada pada logika *high*.
6. Variabel pembagi dikalikan dengan nilai 2 untuk memproses bit selanjutnya.
7. Proses nomor 4 sampai 6 diulang sebanyak 8 kali.

**F**lowchart prosedur pembacaan data (TForm1.read\_byte) dengan antarmuka SPI menggunakan PC-Link USBer:



**Gambar 3**  
**Flowchart** Prosedur Pembacaan Data dengan Antarmuka SPI menggunakan PC-Link USBer

Cara kerja prosedur pembacaan data (TForm1.read\_byte) secara SPI menggunakan PC-Link USBer.

1. Pertama-tama variabel nilai diinisialisasi dengan nilai 0.
2. Kemudian variabel j juga diinisialisasi dengan nilai 1.
3. Pengambilan data diawali dengan memberikan sinyal *low* pada SCK dan sinyal *high* pada CLR.

4. Kemudian memberikan sinyal *high* pada SCK dan sinyal *high* pada CLR.
5. Setelah itu data siap diambil dan dibaca.
6. Data di-**AND**-kan dengan 8 untuk membaca logika pin MISO.
7. Data digeser 4 bit ke kiri.
8. Variabel nilai di-**OR**-kan dengan variabel data.
9. Jika j tidak sama dengan 8 maka variabel nilai digeser ke kanan 1 bit.
10. Prosedur nomor 3 sampai 9 diulang sebanyak 8 kali.

**C**iri khas dari sistem ini adalah penggunaan mode Bit Bang dengan *driver* D2XX yang dapat mengubah fungsi dari jalur I/O PC-Link USBer menjadi 8 bit bi-directional I/O. Di bawah ini akan dijelaskan cara mengakses jalur I/O pada mode Bit Bang dengan menggunakan Borland® Delphi™ yang sudah terpasang dengan komponen TaUSBDirect. Komponen TaUSBDirect ini tersedia dalam AN126.zip, tetapi tidak termasuk dalam CD produk PC-Link USBer. Pembacaan dan penulisan data dari/ke jalur 8-bit I/O ini tidak dapat dilakukan per bit melainkan harus dilakukan per byte.

Berikut contoh pembacaan dan penulisan jalur I/O PC-Link USBer pada mode Bit Bang:

- Mengubah fungsi pin 1 (TXD) menjadi *input* dan pin 2 (RXD) menjadi *output*.

Sintaksis:

```
usb.SetBitMode($FE,$03);
```

Penjelasan:

**\$FE** = 1111 1110

- satu berarti *output*
- nol berarti *input*
- bit 0 = TXD ; bit 1 = RXD ; bit 2 = RTS ; bit 3 = CTS; dst sesuai dengan urutan pin J2 pada PC-Link USBer

**\$03** = 0000 0011

- satu berarti aktif
- nol berarti tidak aktif

- Cara untuk menuliskan logika pada pin yang sudah difungsikan sebagai *output*, misalkan semua pin sudah difungsikan sebagai *output*.

Sintaksis:

```
FT_Out_Buffer[0]:=$AA;
```

```
usb.Write_USB_Device_Buffer(1);
```

Penjelasan:

**\$AA**=1010 1010

- satu berarti berlogika *high*
- nol berarti berlogika *low*
- **\$AA** artinya pin1=*low*, pin2=*high*, dst
- penentuan logika pin ini harus dilakukan langsung 1 byte

- Cara untuk membaca pin yang sudah difungsikan sebagai *input*.

Sintaksis:

```
usb.GetBitMode;
```

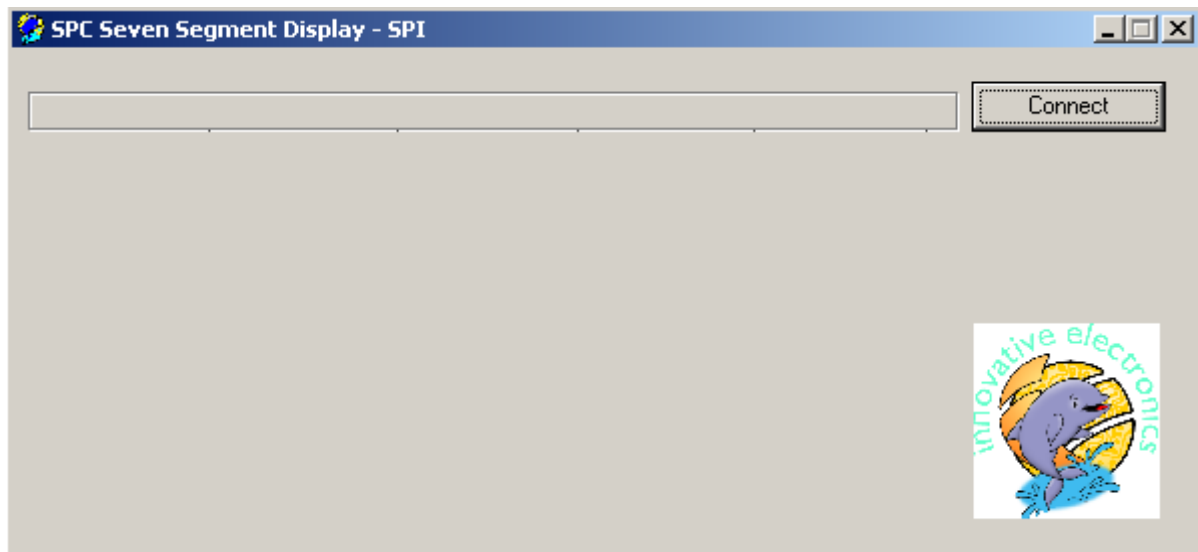
```
data:=get_bit_data;
```

Penjelasan:

- hasil pembacaan terdapat pada variabel data, bit 0 = logika pin 1, bit 1 = logika pin 2, dst.
- sebelumnya variabel data ditentukan sebagai tipe byte atau integer

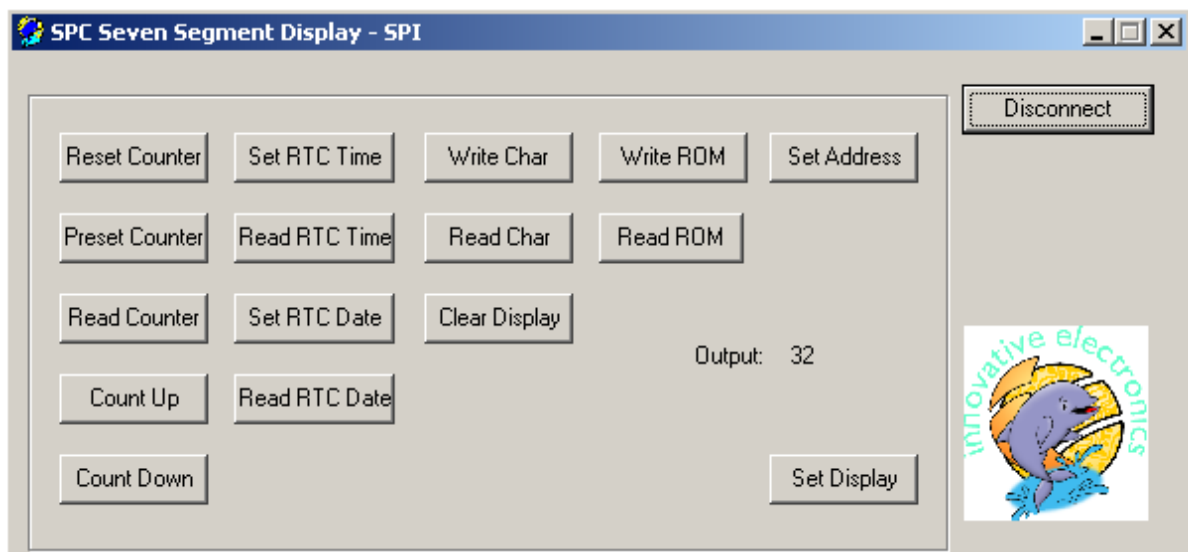
**Catatan:** Komponen TaUSBdirect memiliki nama *default* "aUSBDirect1" , nama ini dapat diganti dengan mengubahnya lewat *properties* pada jendela Delphi. Pada contoh di atas nama komponen TaUSBdirect diganti dengan nama "usb".

Tampilan program komputer sebelum terkoneksi dengan *hardware* adalah sebagai berikut:



**Gambar 4**  
**Tampilan Program Komputer Sebelum Ada Koneksi Dengan Sistem *Hardware***

Tampilan program komputer setelah terkoneksi dengan *hardware* adalah sebagai berikut:



**Gambar 5**  
**Tampilan Program Komputer Setelah Ada Koneksi dengan Sistem *Hardware***

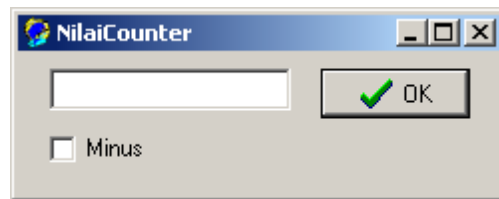
Fungsi dari masing-masing tombol pada program di atas adalah sebagai berikut:

- ◆ Tombol "Connect" / "Disconnect" berfungsi untuk menyambung atau memutus koneksi USB dengan *hardware*.
- ◆ Tombol "Reset Counter" berfungsi untuk mengirim perintah pada SPC Seven Segment Display untuk mengubah nilai Counter di dalamnya menjadi nol.
- ◆ Tombol "Preset Counter" berfungsi untuk mengubah nilai Counter pada SPC Seven Segment Display sesuai dengan keinginan pengguna.

- ◆ Tombol “Read Counter” berfungsi untuk mengambil nilai Counter yang ada pada SPC Seven Segment Display dan menampilkannya pada layar komputer.
- ◆ Tombol “Count Up” berfungsi untuk menambahkan nilai Counter dengan angka satu.
- ◆ Tombol “Count Down” berfungsi untuk mengurangi nilai Counter dengan angka satu.
- ◆ Tombol “Set RTC Time” berfungsi untuk mengubah nilai Jam Menit Detik pada SPC Seven Segment Display sesuai dengan keinginan pengguna.
- ◆ Tombol “Read RTC Time” berfungsi untuk membaca nilai Jam Menit Detik pada SPC Seven Segment Display dan menampilkannya pada layar komputer.
- ◆ Tombol “Set RTC Date” berfungsi untuk mengubah nilai Tanggal Bulan Tahun pada SPC Seven Segment Display sesuai dengan keinginan pengguna.
- ◆ Tombol “Read RTC Date” berfungsi untuk membaca nilai Tanggal Bulan Tahun pada SPC Seven Segment Display dan menampilkannya pada layar komputer.
- ◆ Tombol “Write Char” berfungsi untuk menuliskan karakter pada kolom yang diinginkan.
- ◆ Tombol “Read Char” berfungsi untuk membaca karakter pada kolom yang diinginkan dan menampilkannya pada layar komputer.
- ◆ Tombol “Clear Display” berfungsi untuk menghapus seluruh karakter pada SPC Seven Segment Display.
- ◆ Tombol “Write ROM” berfungsi untuk mengisi data ke dalam EEPROM pada alamat yang diinginkan.
- ◆ Tombol “Read ROM” berfungsi untuk membaca data pada EEPROM pada alamat yang diinginkan.
- ◆ Tombol “Set Address” berfungsi untuk mengubah nilai alamat SPC Seven Segment Display.
- ◆ Tombol “Set Display” berfungsi untuk mengatur mode tampilan dan tingkat kecerahan tampilan SPC Seven Segment Display.

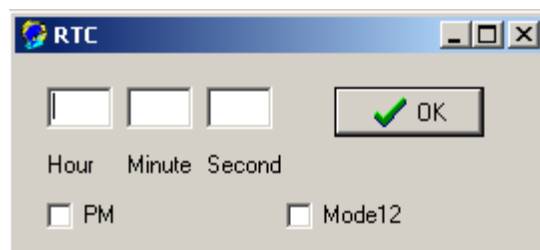
**Catatan:** Pastikan alamat pada program sudah sesuai dengan alamat yang terprogram pada SPC Seven Segment Display. Cara mengubahnya dengan memasukkan angka alamat pada kolom “Address” di jendela “Setting Display” yang akan muncul saat tombol “Set Display” ditekan.

Jendela “NilaiCounter” yang muncul setelah tombol “Preset Counter” ditekan



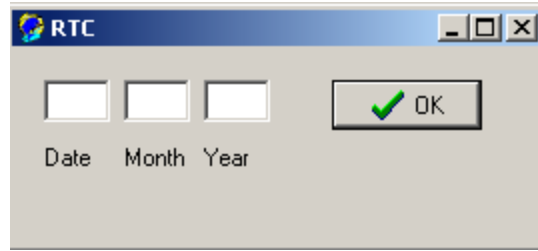
**Gambar 6**  
**Tampilan Program Komputer Setelah Tombol “Preset Counter” Ditekan**

Jendela “RTC” yang muncul setelah tombol “Set RTC Time” ditekan



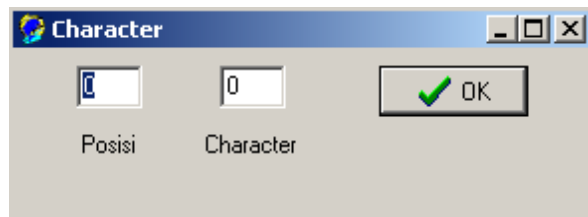
**Gambar 7**  
**Tampilan Program Komputer Setelah Tombol “Set RTC Time” Ditekan**

Jendela “RTC” yang muncul setelah tombol “Set RTC Date” ditekan



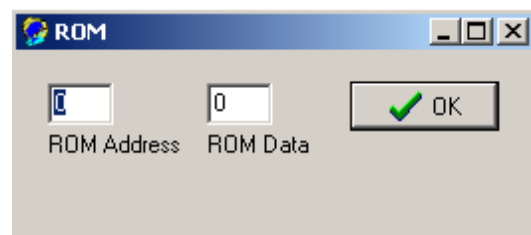
**Gambar 8**  
**Tampilan Program Komputer Setelah Tombol “Set RTC Date” Ditekan**

Jendela “Character” yang muncul setelah tombol “Write Char” ditekan



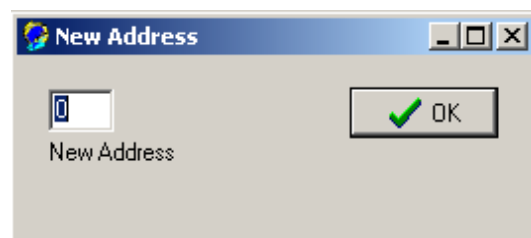
**Gambar 9**  
**Tampilan Program Komputer Setelah Tombol “Write Char” Ditekan**

Jendela “ROM” yang muncul setelah tombol “Write ROM” ditekan



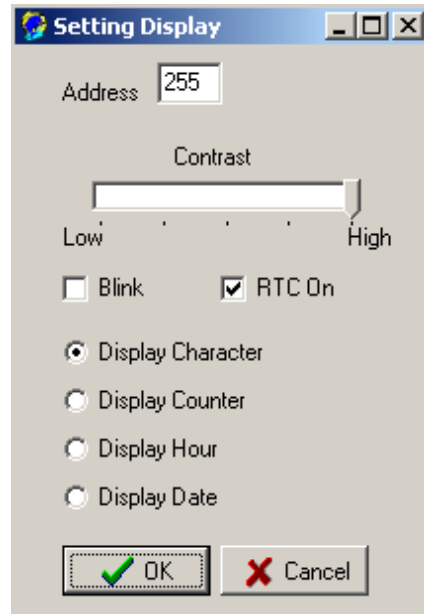
**Gambar 10**  
**Tampilan Program Komputer Setelah Tombol “Write ROM” Ditekan**

Jendela “New Address” yang muncul setelah tombol “Set Address” ditekan



**Gambar 11**  
**Tampilan Program Komputer Setelah Tombol “Set Address” Ditekan**

Jendela “Setting Display” yang muncul setelah tombol “Set Display” ditekan



**Gambar 12**  
**Tampilan Program Komputer Setelah Tombol “Set Display” Ditekan**

**L**isting program terdapat pada **AN126.ZIP**.

**S**elamat berinovasi!

All trademarks, trade names, company names, and product names are the property of their respective owners.  
All softwares are copyright by their respective software publishers and/or creators.