

DT-AVR

DT-AVR *Application Note*

AN154 – Menggunakan Mouse PS/2 untuk Graphic LCD dengan bantuan DT-AVR

Low Cost Series

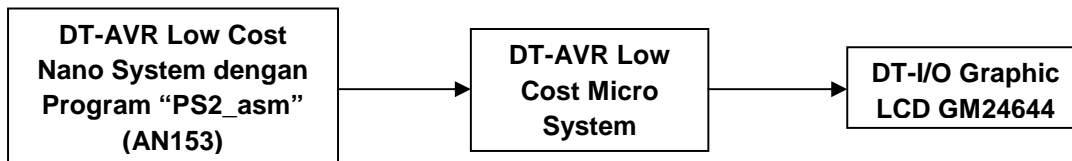
Oleh: Tim IE dan Nathanael R.A (UK Petra)

Graphic LCD GM24644 adalah sebuah LCD grafik yang berukuran 240x64 *pixel*, dan dapat difungsikan dalam dua mode yaitu mode teks dan grafik secara bersamaan. Artikel ini menggunakan sebuah mouse sebagai *input device* dari *user* dengan kemampuan untuk mendeteksi adanya pergeseran *mouse*, penekanan tombol kiri, tombol kanan ataupun *event scroll* yang diimplementasikan pada *graphic LCD* untuk membuat sebuah *User Interface Menu*. Perihal permasalahan untuk mendeteksi dan pengolahan data dari mouse telah dijelaskan pada AN153, sehingga artikel kali ini akan lebih menekankan pada proses pembuatan *User Interface* pada layar *graphic LCD* 240x64. Modul mikrokontroler yang digunakan pada artikel ini adalah DT-AVR Low Cost Micro System dengan menggunakan mikrokontroler ATmega8535. Aplikasi ini menggunakan beberapa modul sebagai berikut :

- Aplikasi pada AN153 (DT-AVR Low Cost Nano System + konektor DIN 6 pin + *Mouse PS/2*)
- DT-AVR Low Cost Micro System
- DT-I/O Graphic LCD GM24644 *

* DT-I/O Graphic LCD GM24644 saat ini telah discontinued dan diganti dengan DT-I/O Graphic LCD 240X64. Penyesuaian pin-pin tersebut akan dijelaskan pada artikel ini.

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN154

Hubungan antara DT-AVR Low Cost Micro System dengan DT-AVR Low Cost Nano System adalah sebagai berikut:

DT-AVR Low Cost Micro System	DT-AVR Low Cost Nano System
RXD (J13 Pin 3)	TXD (J8 Pin 4)
GND (J10/J11/J12/J13 Pin 1)	GND (J7/J8 Pin 1)

Tabel 1
Hubungan antara DT-AVR Low Cost Micro System dengan DT-AVR Low Cost Nano System

Komunikasi serial antara DT-AVR Low Cost Micro System dengan DT-AVR Low Cost Nano System hanya bersifat satu arah saja, sehingga hanya diperlukan satu pin saja (pin TXD pada sisi pengirim, dan pin RXD pada sisi penerima).

Aturlah jumper konfigurasi UART pada DT-AVR Low Cost Micro System dan DT-AVR Low Cost Nano System seperti pada Tabel 2.

DT-AVR Low Cost Micro System	DT-AVR Low Cost Nano System
J4 dan J5 pada posisi 2-3	J4 dan J5 pada posisi 2-3

Tabel 2
Konfigurasi UART TTL pada DT-AVR Low Cost Series

Hubungan antara DT-I/O Graphic LCD GM24644 dengan DT-AVR Low Cost Micro System adalah sebagai berikut:

DT-AVR Low Cost Micro System	DT-I/O Graphic LCD GM24644	DT-I/O Graphic LCD 240x64
VCC (J12 Pin 2)	VCC (J2 Pin 1)	VCC (J4 Pin 2)
GND (J12 Pin 1)	GND (J2 Pin 2)	GND (J4 Pin 1)
PA.0 – PA.7 (J10)*	DB0 – DB7 (J1 Pin 1 – Pin 8)	DB0 – DB7 (J4 Pin 3 – Pin 10)
PB.2 (J11 Pin 5)*	T0 (J2 Pin 5)	T0 (J5 Pin 7)
PB.0 (J11 Pin 3)*	WR (J2 Pin 7)	WR (J5 Pin 9)
PB.1 (J11 Pin 4)*	RD (J2 Pin 8)	RD (J5 Pin 10)
PB.4 (J11 Pin 7)*	RST (J2 Pin 9)	RST (J5 Pin 5)

* pin ini tidak mutlak dan dapat diganti pin lain tapi juga harus mengubah program

Tabel 3
Hubungan antara DT-AVR Low Cost Micro System dengan DT-I/O Graphic LCD GM24644

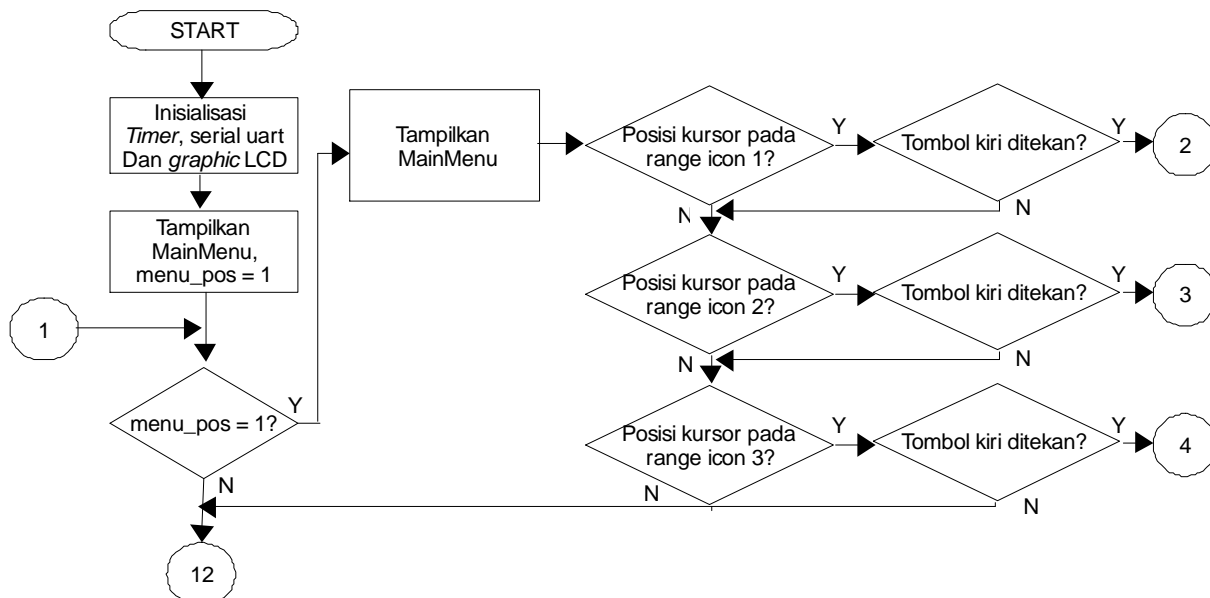
Pada DT-I/O Graphic LCD GM24644 aturlah jumper J3 pada posisi 1-2 untuk ukuran *font* 8x8 (default). Hal ini penting karena gambar tidak akan tampil dengan baik apabila *jumper* diatur pada posisi 2-3 (ukuran *font* 8x6). Sedangkan untuk DT-I/O Graphic LCD 240x64 memerlukan beberapa pengaturan, antara lain :

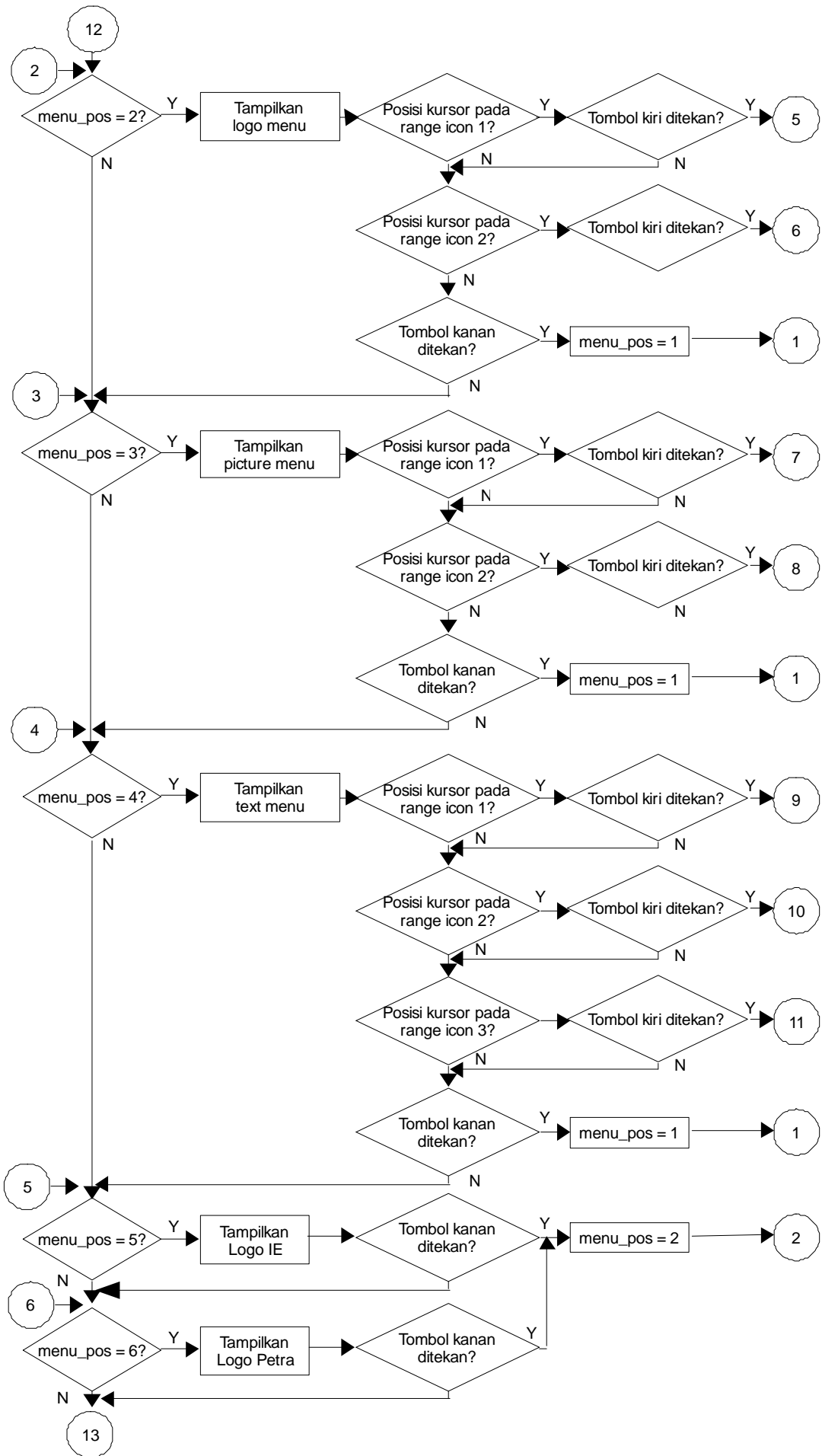
- Atur jumper J6 pada posisi 2-3 sebagai pengaturan ukuran font 8x8.
- Atur jumper J7 pada posisi 1-2 yang menyebabkan pengaturan Reset dari mikrokontroler

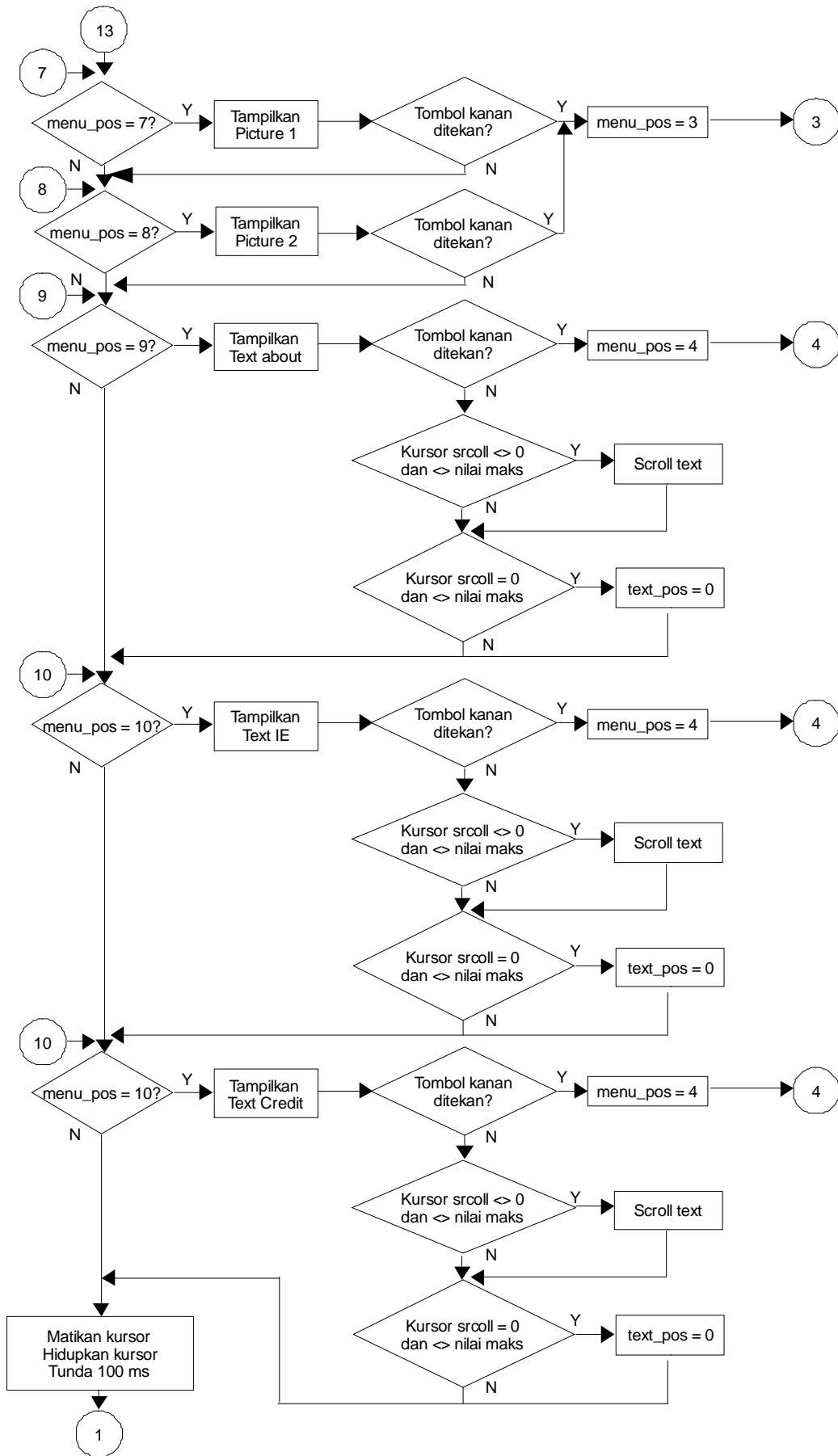
Setelah semua rangkaian dan catu daya terhubung dengan benar, programlah **Graphic_LCD.hex** ke dalam DT-AVR Low Cost Micro System menggunakan **DT-HiQ AVR In System Programmer / DT-HiQ AVR USB ISP** atau divais *in-system programmer* lain yang memiliki konektor 10 pin dan sesuai dengan standar ATMEL.

Jika tampilan LCD tidak jelas atau kabur, aturlah *contrast* LCD dengan memutar VR1 pada board DT-I/O Graphic LCD GM24644 / DT-I/O Graphic LCD 240x64.

Flowchart dari program utama Graphic_LCD.c adalah sebagai berikut:

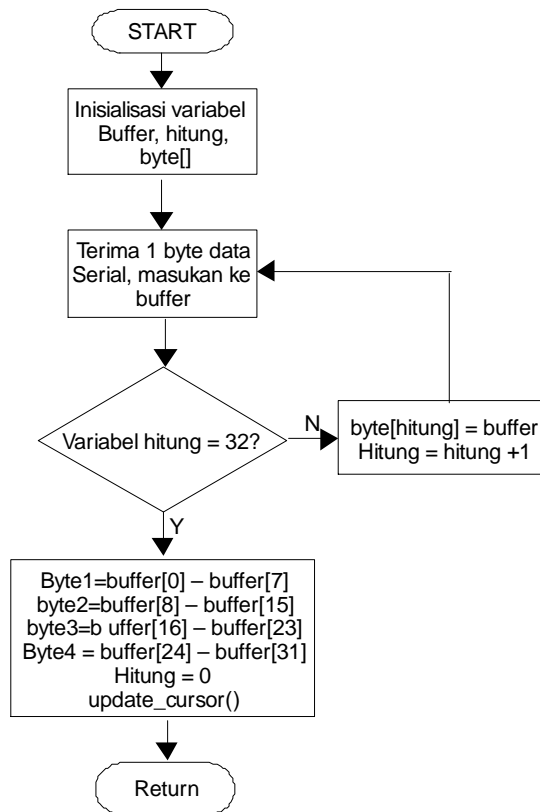






Gambar 2
Flowchart Program utama Graphic_LCD.c

Flowchart dari *interrupt* serial adalah sebagai berikut:



Gambar 3
Flowchart subrutin interup serial **Graphic_LCD.c**

Cara kerja program secara garis besar adalah sebagai berikut:

Program utama:

1. Pertama-tama, program akan melakukan inisialisasi *timer* yang digunakan untuk *delay*, inisialisasi komunikasi serial beserta *interrupt*-nya dan inisialisasi LCD grafik.
2. Kemudian program akan menampilkan MainMenu yang merupakan menu utama, dan mengisi variabel `menu_pos = 1`.
3. Program akan mengecek variabel `menu_pos`. Menu yang ditampilkan pada layar sangat tergantung dari isi variabel `menu_pos`.
4. Setelah menampilkan menu utama, program akan menampilkan menu yang bersangkutan dan mengecek status *mouse*. Apabila tombol *mouse* kiri ditekan dan kursor *mouse* berada pada salah satu *icon*, maka variabel `menu_pos` akan diisi sesuai dengan kode menu yang berkaitan dengan *icon* tersebut, dan pada *looping* berikutnya, menu tersebut akan ditampilkan. Apabila tombol *mouse* kanan ditekan, maka tampilan akan kembali ke menu sebelumnya.
5. Khusus pada menu tampilan gambar penuh, tombol yang berlaku adalah tombol *mouse* kanan saja. Sedangkan pada menu tampilan teks, melakukan *scroll* pada *mouse* akan menggeser tulisan ke bawah/atas. Tombol yang berlaku juga hanya tombol *mouse* kanan saja.
6. Program akan memperbarui posisi kursor *mouse* dengan perintah `Cursor_Off()` dan `Cursor_On()` kemudian melakukan *delay* selama 100 ms agar layar tidak berkedip terlalu cepat.
7. Program kembali ke no. 3

Rutin interupsi serial:

1. Apabila ada data yang masuk melalui *port* serial, maka interupsi serial akan aktif dan program pada vektor interupsi serial akan dijalankan.
2. Pertama-tama, program akan melakukan deklarasi variabel-variabel yang dipakai, dan me-*reset* semua nilai variabel menjadi nol. Setelah itu, program akan menerima satu byte data serial yang didapat dari *register* UDR pada mikrokontroler dan memasukkan data tersebut ke variabel `buffer`.

3. Program akan memeriksa, apakah variabel hitung sudah bernilai 32. Apabila belum, maka isi variabel *buffer* akan disalin ke variabel *array* byte pertama, kedua, dan seterusnya. Setelah itu nilai hitung ditambah dengan satu, dan program menunggu data selanjutnya.
4. Apabila nilai variabel hitung sudah mencapai 32, maka proses perulangan akan berhenti. Setelah itu, nilai yang disimpan pada variabel *array* byte tadi disusun menjadi empat byte data. Masing-masing byte terdiri dari 8 bit data, dengan ketentuan:
 - byte1 = *buffer*[0] – *buffer*[7] (data pertama sampai kedelapan)
 - byte2 = *buffer*[8] – *buffer*[15] (data kesembilan sampai keenam belas)
 - byte3 = *buffer*[16] – *buffer*[23] (data ketujuh belas sampai data keduapuluh empat)
 - byte4 = *buffer*[24] – *buffer*[31] (data keduapuluh lima sampai ketigapuluh dua)Byte pertama dari pembacaan data *mouse* berisi *overflow flag* dari *counter* pada *mouse*, *sign bit*, dan status penekanan masing-masing tombol pada *mouse*. Byte kedua berisi hitungan gerakan *mouse* pada sumbu x. Byte ketiga berisi hitungan gerakan *mouse* pada sumbu y. Byte keempat berisi hitungan gerakan *scroll* pada *mouse*.

Listing program *Graphic_LCD.c* terdapat pada **AN154.ZIP**

Selamat 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.