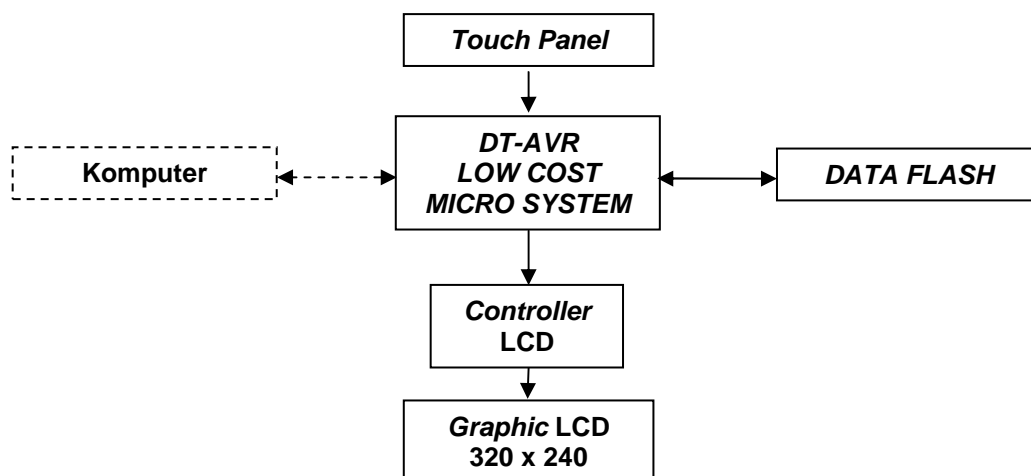


Pada jaman sekarang ini, *touch screen panel* semakin umum dipakai sebagai media *input*. PDA, *handphone*, mesin ATM, dan *panel control* pabrik adalah beberapa contoh peralatan yang menggunakan teknologi *touch screen*. Fenomena ini berkembang pesat dikarenakan selain praktis *touch screen panel* cukup mudah dikonfigurasi dengan hanya menggunakan mikrokontroler. Aplikasi penggunaan Graphic LCD 320 x 240 sudah dijelaskan secara lengkap pada AN155. Pada kesempatan kali ini, AN155 akan dikembangkan untuk membuat sistem GUI sederhana dengan bantuan *touch screen panel* pada layar LCD. *Touch screen panel* yang digunakan pada artikel ini memiliki jenis *resistive* yang berarti *touch screen panel* akan menghasilkan hambatan yang berbeda jika disentuh. Untuk membaca perubahan nilai dari *touch screen panel* ini, digunakan bantuan mikrokontroler untuk membaca input *touch screen panel* dan mengatur tampilan dari Graphic LCD 320 x 240. Modul mikrokontroler yang digunakan adalah DT-AVR Low Cost Micro System dengan mengganti *default crystal* 4MHz menjadi 16MHz. Modul DT-AVR Low Cost Micro System merupakan modul mikrokontroler yang berbasis mikrokontroler AVR ATmega8535. Pemilihan mikrokontroler ini disebabkan karena telah disediakan ADC internal di dalam mikrokontroler ATmega8535 tersebut. Mikrokontroler ATmega8535 akan langsung terhubung ke LCD kontroler untuk menampilkan gambar di LCD Graphic 320 x 240. Pertama, mikrokontroler akan menampilkan 4 buah gambar kecil pada layar LCD. Kemudian mikrokontroler akan membaca input tegangan yang dihasilkan oleh *touch screen panel*. Perubahan tegangan pada *touch screen panel* dikarenakan adanya perubahan hambatan yang disebabkan oleh penekanan *touch screen panel*. Hasil penekanan *touch screen panel* tersebut dianalisa, apabila penekanan tersebut berada di daerah gambar 1, maka gambar tersebut akan secara otomatis diperbesar (*full screen*). Demikian juga halnya untuk penekanan pada gambar 2, 3 dan 4. Program untuk DT-AVR Low Cost Micro System ini dikembangkan dalam bahasa C menggunakan program AVR Studio 4 yang terintegrasi dengan WinAVR. Untuk memasukkan gambar ke dataflash digunakan program BMP Flash Programmer yang dikembangkan dengan Borland Delphi 7. Aplikasi ini menggunakan modul berikut:

- o 1 DT-AVR Low Cost Micro System (penggantian *crystal* 4MHz menjadi 16MHz)
- o *Touch Screen Panel* untuk Graphic LCD WC320240A-FCI-NUTS
- o Graphic LCD WC320240A-FCI-NUTS
- o *Dataflash* AT45DB041B
- o *Embedded Memory LCD Controller* SED1375

Adapun blok diagram secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram Touch Screen Panel Graphic LCD

Hubungan antara modul-modul tersebut adalah sebagai berikut:

<i>Touch Panel</i>	DT-AVR Low Cost Micro System
X+	PA.0
Y+	PA.1
X-	PA.2
Y-	PA.3

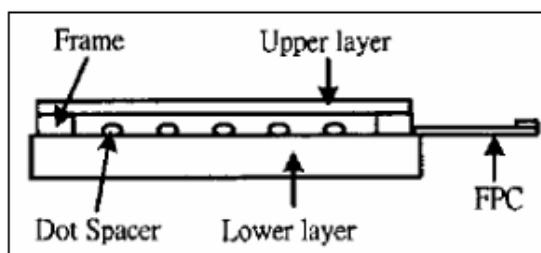
Tabel 1
Hubungan antara *Touch Panel* dengan DT-AVR Low Cost Micro System

3.3.1 Upper layer

- substrate : ITO Film ◦
- type : Haze 8±4% ◦
- resistance : 200Ω~900Ω ◦
- thickness : 188um ◦

3.3.2 Lower layer

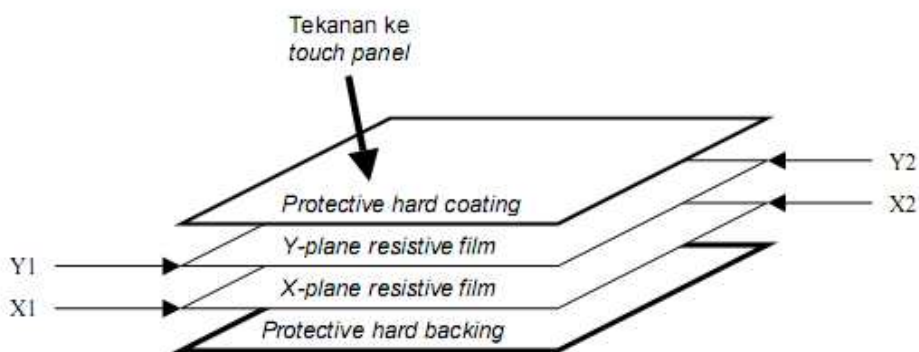
- substrate : ITO Glass ◦
- resistance : 200Ω~900Ω ◦
- thickness : 1.1mm ◦



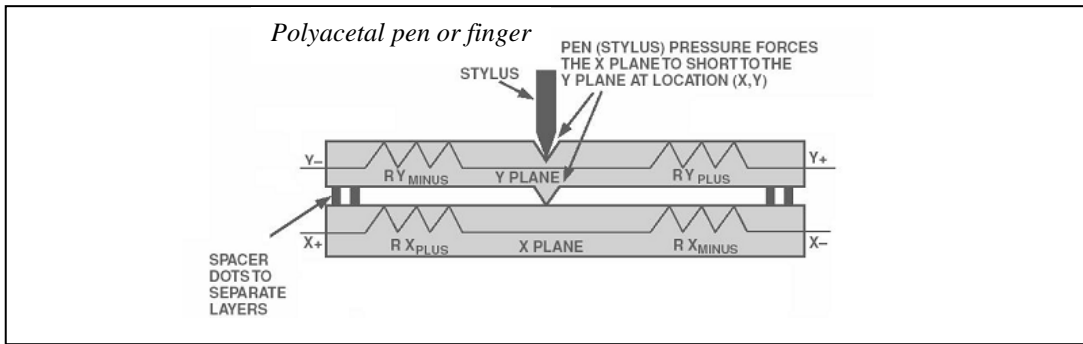
3.3.3 FPC.

- substrate : Polyimide Film ◦
- length : 76.7mm(From panel edge) ◦
- circuit material : copper foil ◦

Gambar 2
Ilustrasi touch panel resistif

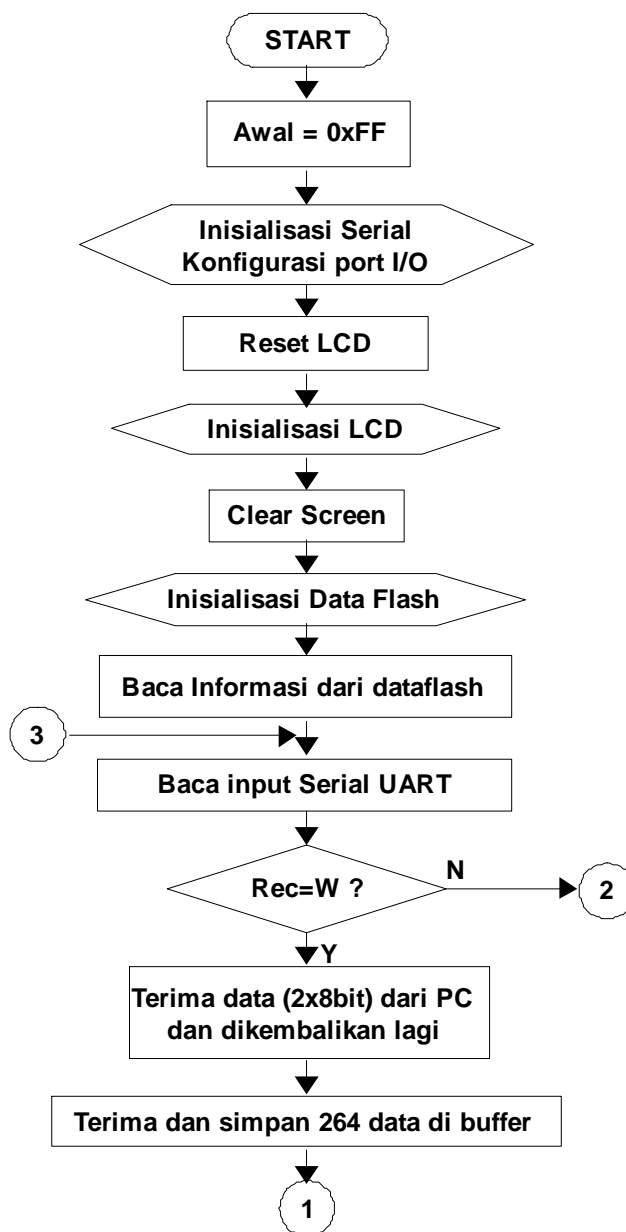


Gambar 3
Model mekanis touch panel resistif

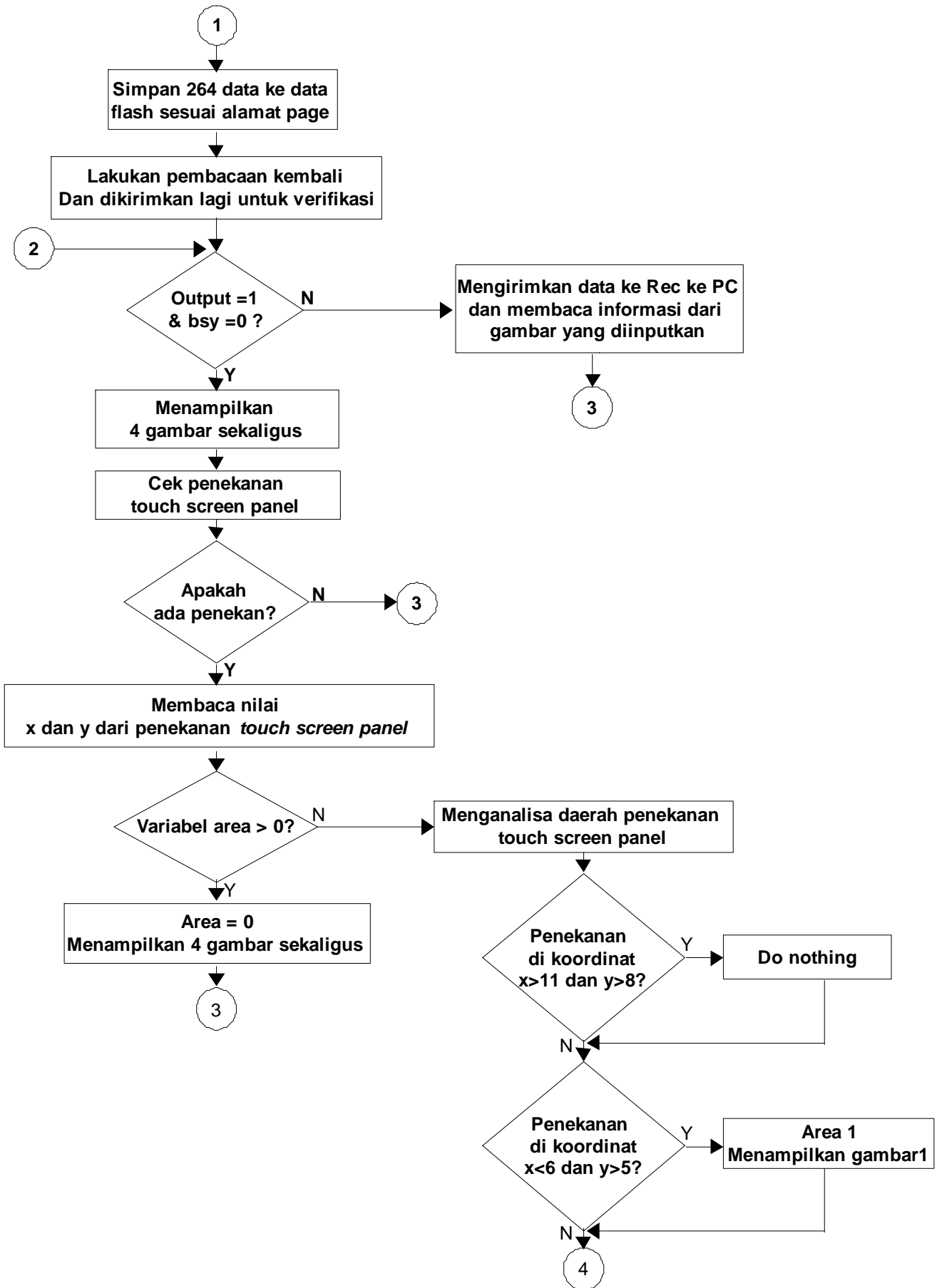


Gambar 4
Ilustrasi penekanan pada touch panel resistif

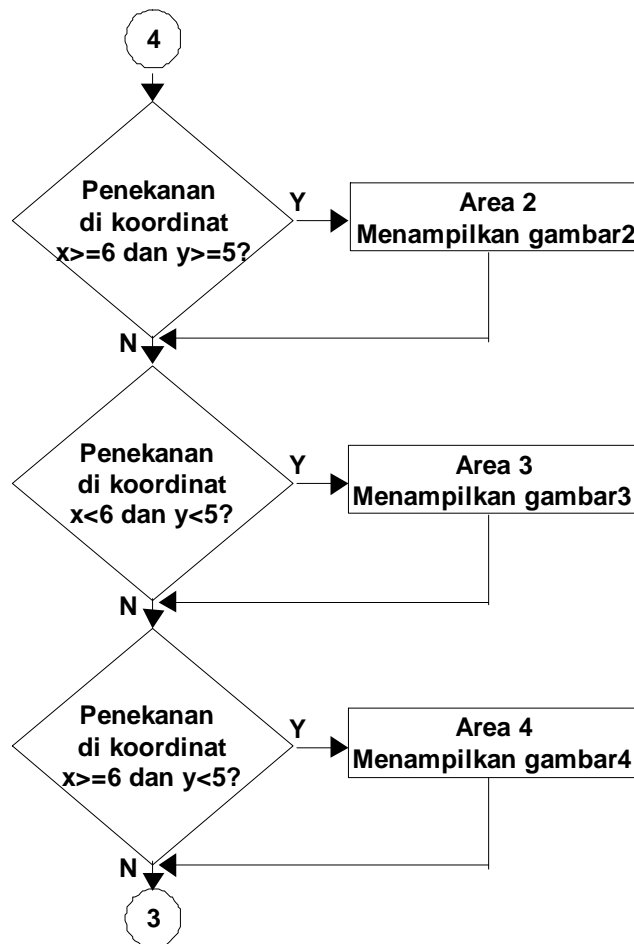
Flowchart dari program AVRTouchScreenLCD.c adalah sebagai berikut:



Gambar 5
Flowchart program AVRTouchScreenLCD



Gambar 6
Flowchart program AVRTouchScreenLCD (lanjutan)



Gambar 7
Flowchart program AVRTouchScreenLCD (lanjutan)

Program AVRTouchScreenLCD.c akan diproses sebagai berikut:

1. Pada awal program akan mengisikan variabel *awal* dengan nilai hexa **0xFF**. Variabel ini berfungsi sebagai tanda agar program dapat menampilkan 4 gambar secara bersama pada saat awal program berjalan.
2. Inisialisasi serial UART 8 bit, 1 stop bit, none parity dan baud rate 38400 bps dan konfigurasi port I/O.
3. Kemudian reset LCD dan inisialisasi LCD. Setelah terinisialisasi, maka layar LCD akan dibersihkan dengan cara memberi 1 jenis warna.
4. Pengaturan jalur komunikasi SPI untuk dataflash.
5. Setelah jalur komunikasi selesai diatur, kemudian membaca informasi file gambar yang ada di dalam dataflash, seperti jumlah gambar, jumlah *palette*, alamat *page* dan *byte* dari data pertama *palette*, halaman dan *byte* dari setiap gambar serta ukuran dari setiap gambar. Jika telah selesai membaca informasi, maka akan menghasilkan nilai "1" yang diisi pada variabel "output" yang menandakan bahwa informasi telah didapat.
6. Kemudian variabel *bsy*=0. Variabel ini berfungsi untuk menandakan sistem komunikasi serial UART. Jika *bsy* bernilai "1" maka antara mikrokontroler dan PC sedang melakukan komunikasi serial, Jika "0" maka tidak ada komunikasi antara PC dengan mikrokontroler melalui jalur komunikasi serial UART.
7. Inisialisasi ADC pada internal ATmega8535 dan mengisikan variabel **Area** dengan 5. ADC berfungsi untuk membaca nilai tegangan yang dihasilkan pada *touch screen panel*.
8. Cek data input dari komunikasi serial UART yang disimpan pada variabel "Rec". Jika Rec berisi karakter "W" berarti PC (program **BMPFlashProgrammer.exe**) akan melakukan pengisian data ke dataflash. Apabila bukan karakter "W", maka program akan lompat ke langkah no 15.
9. Setelah masuk ke bagian pembacaan data serial UART, maka variabel Rec diisi dengan nilai "0" dan *bsy* diisi dengan nilai 1. Pengisian variabel Rec = 0 berfungsi agar tidak terjadi pengisian data ke dalam dataflash setelah mengisikan data ke dalam dataflash.

10. Program menerima pengiriman data *page* yang merupakan *page*/halaman dari dataflash dan disimpan di dalam variabel array "address[0]" dan "address[1]". Dikarenakan jumlah halaman dari dataflash adalah 2048 (memerlukan 11 bit data) dan sistem komunikasi serial adalah 8 bit, maka data yang diterima sebanyak 2 x 8 bit. Setelah didapat kedua data tersebut, maka digabungkan menjadi satu kesatuan di dalam variabel *page*. Kemudian data di variabel *page* dipecah kembali menjadi 8 bit dan dikirimkan ke serial UART untuk proses verifikasi oleh program **BMPFlashProgrammer.exe**.
11. Setelah proses verifikasi *page* selesai, maka **BMPFlashProgrammer.exe** akan mengirimkan data sebanyak 264 data dari **page pertama** dataflash ke dalam mikrokontroler melalui komunikasi serial UART. Jumlah 264 data merupakan jumlah data yang disimpan pada setiap halaman dataflash. Hasil pembacaan 264 data oleh mikrokontroler disimpan dalam variabel array "*buffer[]*"
12. Setelah selesai, maka mikrokontroler akan mengisikan data ke dalam dataflash dengan mengacu pada variabel *page* dan *buffer[]* yang didapat dari **BMPFlashProgrammer.exe** (sesuai dengan langkah no 10 dan 11).
13. Data di dalam dataflash dibaca kembali oleh mikrokontroler mengacu pada isi dari variabel *page* sebelumnya. Setelah didapat hasil pembacaan, kemudian data tersebut dikirimkan lagi ke PC melalui komunikasi serial untuk proses verifikasi data.
14. Jika data verifikasi benar, maka program di dalam **BMPFlashProgrammer.exe** akan secara otomatis menambah nilai dari variabel *page*. Kemudian program diulangi seperti pada langkah no 8. Hal ini terdapat perbedaan pada langkah no 11, **BMPFlashProgrammer.exe** tidak lagi mengirimkan data sebanyak 264 data dari **page pertama**, melainkan mengirimkan data sebanyak 264 data dari **page berikutnya**. Proses pengulangan ini berakhir ketika semua data gambar yang akan dimasukkan ke dalam dataflash sudah selesai.
15. Jika tidak menerima karakter "W" dari input serial UART, maka dicek apakah variabel *output=1* dan *bsy=0*. Jika tidak, maka program akan langsung menuju ke langkah no 22.
16. Kemudian program akan memeriksa nilai dari variabel **awal**. Jika variabel **awal** bernilai lebih dari 0x00 hexa, maka program akan menampilkan 4 gambar sekaligus dalam 1 layar, mengisikan variabel **awal** menjadi 0x00 hexa dan mengisikan variabel **Area** dengan 0. Pada awal program berjalan (no 1) program mengisikan variabel **awal** dengan 0xFF hexa. Sehingga pada awal program berjalan pasti masuk ke dalam rutin ini. Jika telah terjadi pengulangan program (tidak reset), maka tidak akan masuk ke dalam rutin ini melainkan langsung lompat dari no 15 ke no 16. Hal ini terjadi karena variabel **awal** telah diisi dengan 0x00 Hexa.
17. Program akan memeriksa apakah ada penekanan pada *touch screen panel*. Jika terdapat penekanan, maka dilakukan pembacaan nilai tegangan pada sumbu X dan sumbu Y lalu disimpan pada variabel **X** dan **Y**.
18. Setelah itu program akan memeriksa isi dari variabel **Area**, jika berisi nilai lebih besar dari 0, maka program akan mengisikan nilai dari variabel **Area** dengan 0 serta akan menampilkan 4 gambar sekaligus dalam 1 layar. Jika tidak, maka program akan langsung lompat ke no 19.
19. Program akan menghitung dan menganalisa hasil pembacaan sumbu x dan y dengan rumus:

$$x = 11 - (x - 34) / 18$$

$$y = 8 - (y - 52) / 21$$
 Fungsi ini berfungsi untuk membagi daerah penekanan *touch screen panel* menjadi 4 buah bagian. Setelah didapatkan hasil, kemudian dianalisa di mana letak penekanan tersebut. Hal ini berguna untuk menentukan tugas yang harus dilakukan oleh mikrokontroler ketika terjadi penekanan.
20. Jika didapat $x > 11$ dan $y > 6$ maka program tidak akan melakukan tugas apapun. Kemudian program akan memeriksa kembali
 - a. Jika $x < 6$ dan $y \geq 5$ maka variabel **Area=1** dan menampilkan gambar1 dengan tampilan *full screen*.
 - b. Jika $x \geq 6$ dan $y \geq 5$ maka variabel **Area=2** dan menampilkan gambar2 dengan tampilan *full screen*.
 - c. Jika $x < 6$ dan $y < 5$ maka variabel **Area=3** dan menampilkan gambar3 dengan tampilan *full screen*.
 - d. Jika $x \geq 6$ dan $y < 5$ maka variabel **Area=4** dan menampilkan gambar4 dengan tampilan *full screen*.
 Kemudian program menuju ke langkah no 22.
21. Periksa isi dari variabel Rec. Jika berisi karakter "C" maka program akan mengirimkan kembali karakter tersebut ke PC yang berarti semua pengiriman data dari PC ke mikrokontroler untuk mengisikan data ke dataflash telah selesai. Kemudian membaca kembali informasi data gambar di dalam dataflash. Jika tidak langsung menuju ke langkah no 8.
22. Setelah selesai, program akan terus berulang ke langkah 8.

Langkah-langkah perihal pengisian program ke dalam mikrokontroler maupun pengisian gambar ke dataflash dapat dilihat pada AN155. Hubungan detil antar modul dapat dilihat pada masing-masing datasheet modul.

Listing program AVRTouchScreenLCD.c dan BMPFlashProgrammer.exe terdapat pada **AN156.ZIP**

Selamat berinovasi!

AVR Studio is Copyright by ATMEL Corp.

All trademarks, trade names, company names, and product names are the property of their respective owners.