

**T**ouch panel dewasa ini telah banyak digunakan pada peralatan elektronika, misalnya: *handphone* dan PDA. Touch panel digunakan untuk menggantikan fungsi *keypad*, *keyboard*, dan *mouse* sebagai divais *input*, dan tentunya akan lebih mempermudah antarmuka dengan manusia.

Aplikasi kali ini merupakan contoh sederhana penggunaan *resistive touch panel* dengan menggunakan DT-AVR Low Cost Micro System untuk melakukan pembacaan koordinat tekanan. *Internal ADC* pada DT-AVR Low Cost Micro System dimanfaatkan untuk mengukur resistansi *touch panel*. Resistansi ini menunjukkan koordinat tekanan, lalu koordinat yang diperoleh akan dikorespondensikan sebagai sebuah titik pada jendela program aplikasi komputer (juga disertakan dalam AN ini). Program untuk DT-AVR Low Cost Micro System dikembangkan dalam bahasa C yaitu menggunakan *compiler* CodeVisionAVR<sup>®</sup>, sedangkan program aplikasi untuk komputer dikembangkan menggunakan Borland Delphi<sup>®</sup>.

Aplikasi ini membutuhkan modul berikut:

- 1 DT-AVR Low Cost Micro System,
- 1 Touch Panel Graphic LCD 128x64,
- 1 Resistor 4K7 ohm ( $\frac{1}{4}W$  - 5%).

**A**dapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1  
Blok Diagram AN105

**H**ubungan antara modul-modul tersebut adalah sebagai berikut:

DT-AVR Low Cost Micro System	Touch Panel Graphic LCD 128x64
PB.0	Y1
PB.1	Y2
PB.2	X2
PB.4	X1
PA.0	X1
PA.1	Y2

Tabel 1  
Hubungan DT-AVR Low Cost Micro System dengan Touch Panel Graphic LCD 128x64

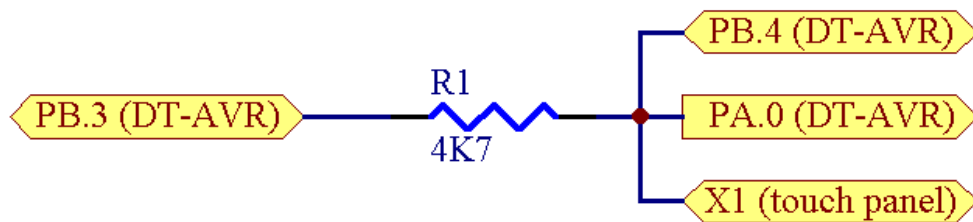
Penggunaan pin mikrokontroler yang tersebut di atas dapat diganti dengan pin lain tetapi tentunya juga harus mengubah alokasi pin pada program mikrokontrolernya. Cara kerja rangkaian dapat dijelaskan sebagai berikut:

- PB.0 dan PB.1 pada DT-AVR Low Cost Micro System digunakan untuk memberikan tegangan pada resistansi layer Y dalam *touch panel*.
- PB.2 dan PB.4 pada DT-AVR Low Cost Micro System digunakan untuk memberikan tegangan pada resistansi layer X dalam *touch panel*.
- PB.3 pada DT-AVR Low Cost Micro System digunakan untuk memberikan *pull-up* pada pin X1 saat tidak ada tekanan pada permukaan *touch panel*. Hal ini dilakukan untuk mendeteksi adanya sentuhan / tekanan.

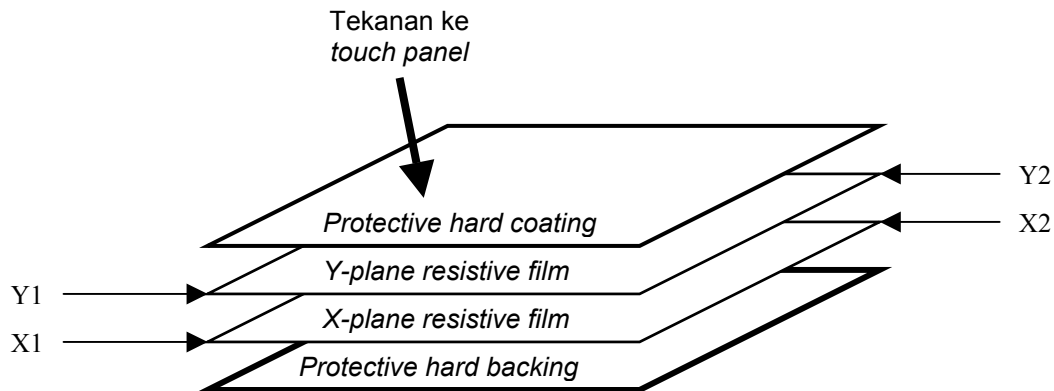
Pada saat tidak ada penekanan, jika PB.3 diberi logic '1' maka pada PA.0 akan terbaca tegangan sebesar  $\pm 5$  Volt.

- PA.0 dan PA.1 pada DT-AVR Low Cost Micro System berfungsi sebagai *input* ADC dan digunakan untuk membaca tegangan pada resistansi X dan Y secara bergantian.

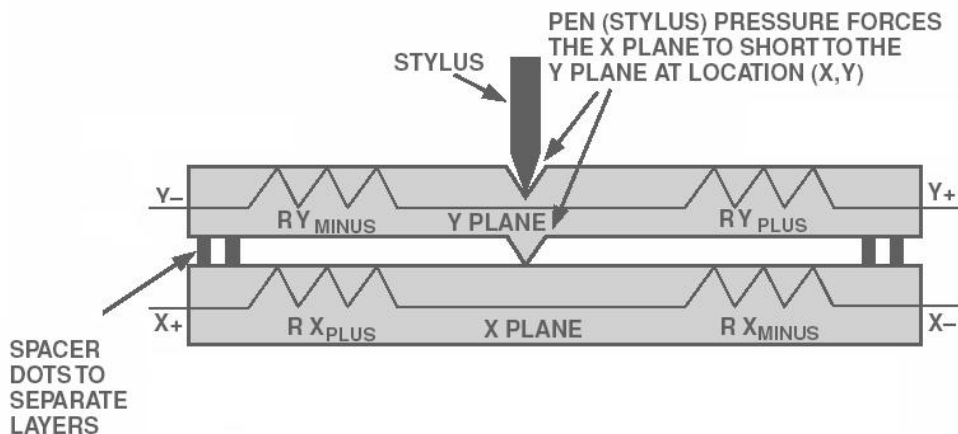
DT-AVR Low Cost Micro System yang digunakan dalam aplikasi ini menggunakan mikrokontroler ATmega8535 yang memiliki 8 kanal 8-bit *internal ADC*. Aturlah *jumper* J6, J7, dan J8 DT-AVR Low Cost Micro System pada posisi 1-2 agar tegangan referensi *internal ADC* ATmega8535 mengambil dari AVCC. Selain itu aturlah juga *jumper* J4 dan J5 DT-AVR Low Cost Micro System posisi 1-2 agar komunikasi serial UART RS-232 dapat digunakan. Gunakan kabel serial DT-AVR Low Cost Micro System untuk menghubungkan modul dengan komputer. Setelah semua rangkaian dan sumber tegangan terhubung dengan tepat, programlah **Coba.HEX** ke DT-AVR Low Cost Micro System dengan menggunakan **DT-HiQ AVR In System Programmer** atau divais *ISP programmer* lainnya yang memiliki konektor 10 pin yang sesuai dengan standar Atmel.



**Gambar 2**  
Hubungan DT-AVR Low Cost Micro System dengan Touch Panel Graphic LCD 128x64

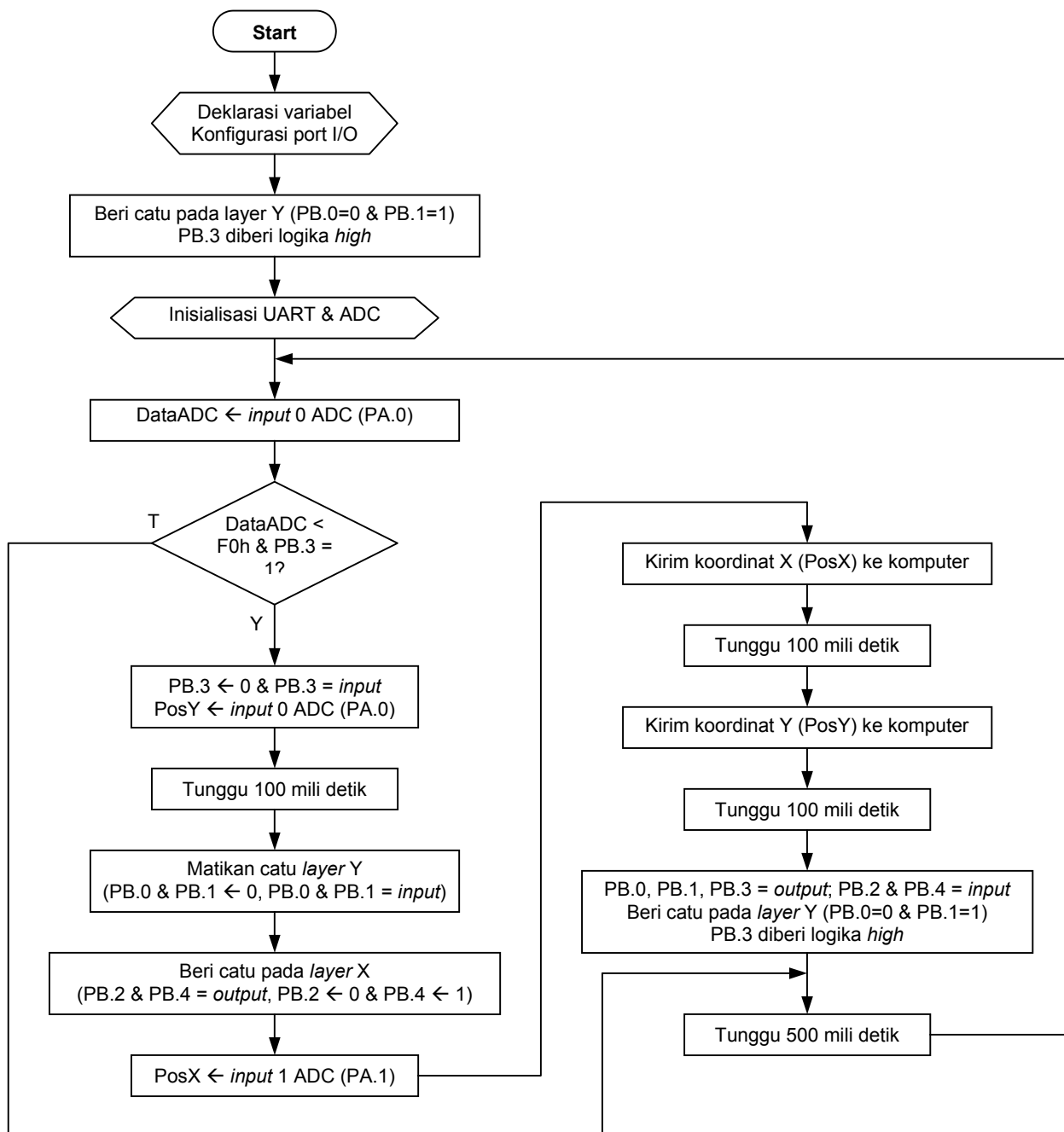


**Gambar 3**  
Model Mekanis dari Resistive Touch Panel



**Gambar 4**  
Ilustrasi Penekanan pada Resistive Touch Panel

**F**lowchart dari program **Coba.c** adalah sebagai berikut:



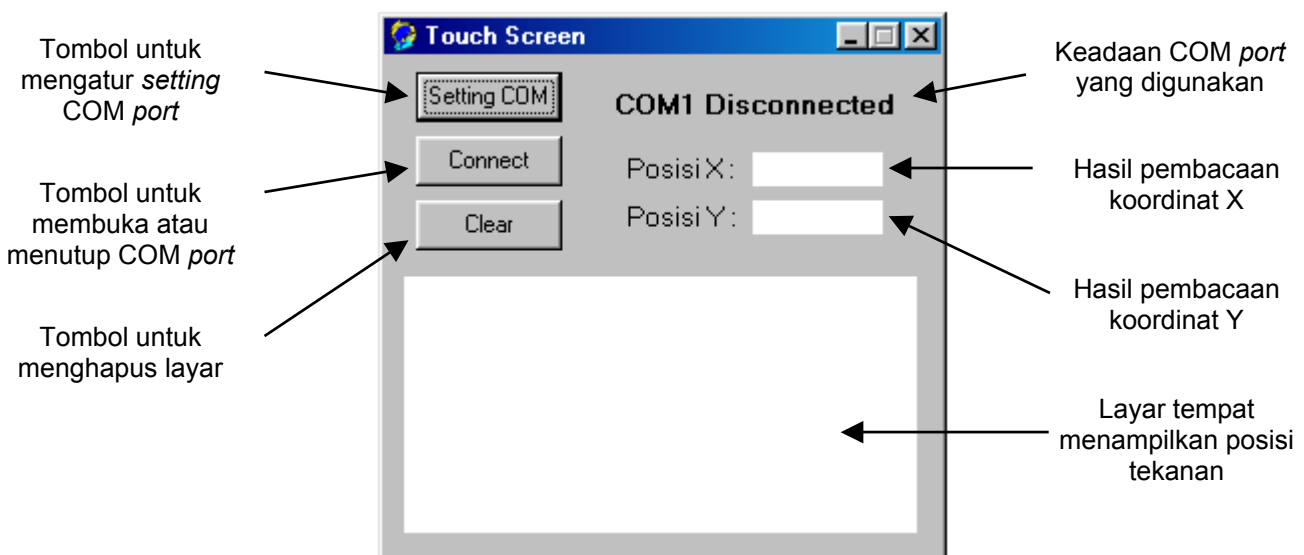
**Gambar 5**  
**Flowchart Program Utama Coba.c**

**P**rogram utama **Coba.c** akan diproses sebagai berikut:

1. Program melakukan deklarasi variabel yang diperlukan yaitu: PosX, PosY, dan DataADC. PosX digunakan untuk menyimpan hasil pembacaan koordinat X. PosY digunakan untuk menyimpan hasil pembacaan koordinat Y. DataADC digunakan untuk menyimpan hasil pembacaan ADC pada saat program melakukan polling keadaan resistansi *layer Y* dari *touch panel*.
2. Lalu program melakukan konfigurasi terhadap *port I/O* ATMEGA8535, yaitu mengatur PB.0, PB.1, dan PB.3 sebagai *output* sedangkan bit lainnya dalam port B sebagai *input*. Kemudian program membuat PB.0 = 0 dan PB.1 = 1 sehingga *layer Y* mendapatkan tegangan catu 5 VDC.
3. Program melakukan konfigurasi port UART sehingga dapat berkomunikasi dengan komputer secara serial (*baudrate* 9600 bps, 8 bit data, 1 bit stop, tanpa bit *parity*). Lalu program juga melakukan inisialisasi *internal ADC* ATMEGA8535 agar dapat berfungsi sebagai ADC 8-bit.

4. Program akan menunggu adanya tekanan pada permukaan *touch panel*. Dalam proses ini, program melakukan pembacaan *input* ADC setiap 500 mili detik (secara poling). Program akan terus melakukan poling hingga nilai pembacaan ADC lebih kecil dari F0h (4,6875 Volt) yang menandakan adanya penekanan dan setelah itu program akan mulai membaca koordinat penekanan.
5. Setelah terdeteksi adanya penekanan pada *touch panel*, maka program akan membuat PB.3 = 0 dan mengkonfigurasi PB.3 sebagai *input*. Lalu program membaca *input* 0 ADC (PA.0) dan disimpan ke dalam variabel PosY, nilai yang diperoleh ini merupakan koordinat Y dari penekanan tersebut.
6. Program menunggu selama 100 mili detik.
7. Program mematikan tegangan catu untuk *layer* Y dengan memberi PB.0 & PB.1 logika 0 serta mengkonfigurasi PB.0 & PB.1 sebagai *input*. Lalu program memberi tegangan catu pada *layer* X dengan mengkonfigurasi PB.2 & PB.4 sebagai *output* serta memberi PB.2 logika 0 dan PB.4 logika 1. Setelah itu program membaca *input* 1 ADC (PA.1) dan disimpan ke dalam variabel PosX, nilai yang diperoleh ini merupakan koordinat X dari penekanan tersebut.
8. Program mengirimkan nilai koordinat X (dari variabel PosX) ke komputer, menunggu selama 100 mili detik, kemudian mengirimkan nilai koordinat Y (dari variabel PosY) ke komputer dan menunggu lagi selama 100 mili detik.
9. Setelah itu program mematikan tegangan catu untuk *layer* X dan mengaktifkan tegangan catu untuk *layer* Y. Untuk itu program mengkonfigurasi PB.0, PB.1, dan PB.3 sebagai *output* sedangkan bit lainnya dalam port B sebagai *input* serta memberi PB.0 logika 0 dan PB.1 & PB.3 logika 1.
10. Kemudian program menunggu selama 500 mili detik dan kembali ke langkah 4.

Data yang dikirimkan oleh DT-AVR Low Cost Micro System dapat dilihat dengan bantuan program **Ending.EXE** yang dapat menampilkan nilai koordinat X dan Y, serta gambaran posisi penekanan. Pengaturan komunikasi serial adalah *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit stop, dan tanpa *flow control*.



**Gambar 6**  
Tampilan Program Ending.EXE

**L**isting program **Coba.c** dan program **Ending.EXE** terdapat pada **AN105.ZIP**.

**S**elamat berinovasi!

CodeVisionAVR is copyright by Pavel Haiduc, HP InfoTech s.r.l.  
Borland Delphi 5.0 is copyright by Inprise Corporation.