

# DESAIN SOFTWARE DAN HARDWARE PENGIDENTIFIKASI DINAMIKA MOTOR DC

Lasman Parulian Purba, Ardian Fitrianto

Program Studi Sistem Komputer, Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya  
(STIKOM), email: [lasman@stikom.edu](mailto:lasman@stikom.edu)

**Abstract:** Some dynamic systems like: mechanics, electrical, and hydrolic can be specified using diferencial equation. Diferencial equation shows behaviour of dynamic system a long the time. For control design requirements like PID, that diferencial equation information needed, special for assure performance achievement dynamic required. This research discuss about software design process needed for identifying DC motor dynamic include its hardwares. Interface used PPI8255A. Result shows that integration between software and hardware found model for DC motor dynamic. There are two DC motors for trying integrity validity design for software and its hardware.

**Keywords:** Diferencial Equation, Dynamic System, DC Motor, Software, Hardware, PI8255A

Untuk dapat meraih suatu tujuan yang dikehendaki, akhir – akhir ini dunia industri diperhadapkan pada suatu persaingan (kompetisi). Kompetisi dapat meliputi kemampuan menggunakan alat – alat yang lebih handal, efisien, akurat, murah dan pengaruhnya terhadap peningkatan produksi. Alat – alat industri yang dipakai sekarang ini masih banyak menggunakan Motor AC dan DC sebagai penggerak utama dalam aktivitasnya, Motor DC saat ini banyak digunakan terutama untuk sistem penggerak dengan daya yang tidak terlalu besar pada industri.

Motor DC merupakan perangkat analog yang komponen utamanya terdiri atas magnet dan kumparan dimana pada waktu arus listrik ini mengalir dapat menyebabkan terjadinya induksi dalam kumparan dengan magnet dalam Motor tersebut sehingga batang poros dapat berputar. Motor juga sebagai *transducer* yang mengubah energi listrik menjadi energi mekanik. Tergantung pada konfigurasi antara beban dan poros Motor, beban akan bergerak secara rotasi maupun translasi, atau mungkin bentuk gerakan yang tak beraturan. Tanpa memadam tipe gerakan, torsi dan kecepatan yang terjadi pada beban dapat dikonversikan menjadi kerja dan daya mekanik (Bahan Pelatihan Nasional, 2002).

Untuk menghindari kerusakan dan kemudahan dalam desain sistem kontrol pada *plant*, Motor DC, diusulkan dalam penelitian ini suatu metode identifikasi sistem atau *plant*. Persamaan model dari sistem atau *plant* merupakan hasil yang didapat dari proses identifikasi. Persamaan tersebut dapat dipakai selanjutnya untuk desain *controller* yang dikehendaki atau sebagai informasi awal yang terpercaya tentang sistem yang hendak di desain *controller*-nya.

Mengetahui perilaku dari sistem yang akan diatur (*plant*) merupakan suatu langkah penting dalam desain sistem pengaturan. Perilaku dari sistem dapat direpresentasikan dengan model atau persamaan. Model dapat diklasifikasikan menjadi dua, model fisik dan model abstrak. Model fisik adalah tiruan dari sistem yang sebenarnya yang disesuaikan dengan fungsi dan cara kerja sistem, namun dalam bentuk kecil atau lazim disebut *prototype*. Model abstrak adalah model yang dibentuk dengan simbol-simbol. Contoh model abstrak adalah model grafik (dalam bentuk gambar), model verbal (dalam bentuk keterangan-keterangan), ataupun dalam bentuk model matematik. Bila dibandingkan dengan model-model yang lain, maka model matematik adalah model yang dapat memberikan deskripsi sistem dan akurasi yang paling baik (Junaedi, A., 1997)

Model matematik dibagi lagi menjadi model deterministik dan model stokastik. Model dikatakan deterministik jika parameter (keluaran) dapat dihitung secara tepat, dalam arti bahwa diasumsikan seluruh parameter dapat dihitung dengan tepat. Model stokastik adalah model yang parameter keluarannya tidak dapat dihitung secara tepat disebabkan adanya unsur random dalam parameter-parameter model yang mempunyai distribusi tertentu.

Dalam sistem diskrit, model masukan keluaran dinamakan dengan fungsi alih pulsa, dan diberikan dengan persamaan:

$$H(z) = \frac{Y(z)}{U(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (1)$$

dimana A dan B adalah polinomial dalam  $z^{-1}$ :

Jika persamaan  $H(z)$  disajikan dalam domain waktu, maka akan diperoleh persamaan beda yang merealisasikan keluaran  $y(t)$  dan masukan  $u(t)$ .

Untuk merealisasikannya dalam domain waktu digunakan operator delay waktu yang mempunyai sifat:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{na} z^{-na} \quad (2)$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{nb} z^{-nb} \quad (3)$$

$$q^{-1}y(t) = y(t-k) \quad (4)$$

$$A(q^{-1})y(t) = B(q^{-1})u(t) \quad (5)$$

$$\begin{aligned} y(t) + a_1 y(t-1) + a_2 y(t-2) + \dots + a_{na} y(t-na) \\ = b_0 u(t) + b_1 u(t-1) + \dots + b_{nb} u(t-nb) \end{aligned} \quad (6)$$

Sehingga persamaan sisi kanan dari  $H(z)$  dapat ditulis menjadi:

Atau dapat pula dituliskan persamaan beda dari model deterministik sebagai:

$$\begin{aligned} y(t) = -a_1 y(t-1) - a_2 y(t-2) - \dots - a_{na} y(t-na) \\ + b_0 u(t) + b_1 u(t-1) + b_2 u(t-2) + \dots + b_{nb} u(t-nb) \end{aligned} \quad (7)$$

Salah satu ciri dari model stokastik adalah dengan masukan acak (*random*). Jika suatu sistem diberi suatu masukan acak, maka model tersebut model *Auto-Regressive Moving Average* dengan masukan eksogen (*Auto Regressive Moving Average Model with eXogenous Inputs*) atau disingkat model ARMAX. Persamaan (8) adalah model ARMAX,

$$A(q^{-1})y(t) = B(q^{-1})u(t) + C(q^{-1})e(t) \quad (8)$$

dimana:

$$A(q^{-1}) = 1 + \sum_{i=1}^{na} a_i q^{-i}$$

$$B(q^{-1}) = \sum_{i=1}^{nb} b_i q^{-i}$$

$$C(q^{-1}) = 1 + \sum_{i=1}^{nc} c_i q^{-i}$$

dan variabel  $q^{-1}$  adalah variabel *delay*, misalnya  $a_3 y q^{-3} = a_3 y(k-3)$ .

Di mana A, B, dan C adalah polinomial dan  $e(t)$  adalah *error* pengukuran. *Error* pengukuran disini dapat terjadi akibat dari ketidaklinearan *plant* (yang diukur dengan alat ukur, dalam hal ini digunakan *tachogenerator*). Model ARMAX biasanya ditulis sebagai ARMAX (Na, Nb, Nc) yang menyajikan bentuk-bentuk umum dari model *transfer function* (filter) digital:

- *Infinite Impulse Response* (IIR) atau ARMAX (Na, Nb, 0)
- *Finite Impulse Response* (FIR) atau ARMAX (1, Nb, 0)

- *Autoregressive (AR)* atau ARMAX (Na,0,1)
  - *Auto Moving- Average (ARMA)* atau ARMAX (Na,0,Nc)
  - *Auto-Regressive Model with eXogenous input (ARX)* atau ARMAX (Na,Nb,1)
- Secara umum bentuk persamaan beda dari model ARMAX adalah:

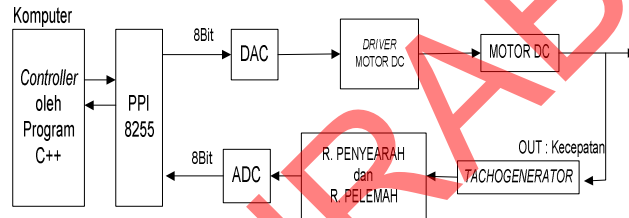
$$y(t) = -\sum_{i=1}^{n_A} a_1 y(t-1) + \sum_{i=0}^{n_B} a_2 y(t-2) + \sum_{i=0}^{n_C} b_1 u(t-1) \quad (9)$$

dimana  $N_A > N_B, N_C$  dan  $e(t)$  adalah *white noise*.

## METODE

### Integrasi Software dan Hardware Sistem

Blok diagram sistem yang diusulkan ditunjukkan pada gambar 1.



Gambar 1. Blok diagram Alat Pengidentifikasi Sistem usulan (Fitrianto, A., 2004)

### Perencanaan dan Pembuatan Software

Karena Borland Turbo C++ 3.0 berbasis pada DOS, maka ukuran program yang akan dihasilkan kecil dan komputer yg di butuhkan juga tidak memerlukan spesifikasi yang tinggi. Pemrograman Turbo C++ ini mendukung mode grafik yang akan menghasilkan suatu tampilan yang lebih bagus di bandingkan dengan mode teks. Program yang dibuat menggunakan dua mode yg ada. Mode teks digunakan untuk menampilkan nilai – nilai yg langsung di dapat dari pengukuran maupun nilai – nilai hasil analisis, sedangkan mode grafik di gunakan untuk tampilan program, menu, grafik, menuliskan fungsi alih, dan sebagainya. Program ini menggunakan pengoperasian *file* sebagai media penyimpanan data. Sehingga data yg di dapat, dapat di analisa secara manual dan dapat di ketahui ke absahan hasil akhir dari data masing-masing. Persamaan dalam *domain* s untuk masing-masing data secara langsung ditampilkan. *File* yang dihasilkan dapat di akses langsung dengan Microsoft Excel sehingga dapat diketahui grafiknya sesuai atau tidak dengan hasil pengukuran.

Program lain yang digunakan adalah program MATLAB. Program ini umum digunakan di dalam analisis dan desain sistem kontrol, selain karena kemudahan penggunaannya, kecepatan eksekusi perintah yang diberikan, MATLAB juga mempunyai banyak fungsi dengan kompleksitas tinggi telah disederhanakan tanpa kekurangan dalam esensinya, seperti pembuatan grafik dari suatu model matematika, pembuatan fungsi dari data-data, dan lain sebagainya. Program ini setidaknya dapat berjalan dalam sistem operasi keluarga Windows, Unix, Linux. Penelitian ini menggunakan sistem operasi Windows'98, sehingga komputer yang digunakan tidak memerlukan spesifikasi *hardware* yang sangat tinggi.

Program Borland C++ 3.0 mendukung untuk pengoperasian *hardware* di luar komputer. Penelitian ini memakai PPI8255 sebagai media penghubung *software* dan *hardware* yang didesain dan di identifikasi.

### Motor DC

Motor adalah *transducer* yang mengubah energi listrik menjadi energi mekanik. Arus listrik yang dicatukan pada motor akan menyebabkan poros motor berputar. Tergantung pada konfigurasi antara beban dan poros motor, beban akan bergerak secara rotasi maupun translasi, atau mungkin sebetuk gerakan yang tak beraturan. Tanpa memandang tipe gerakan, torsi dan kecepatan yang terjadi pada beban dapat dikonversikan menjadi kerja dan daya mekanik. Sebagai sumber daya mekanik, motor telah dipakai untuk begitu banyak keperluan.

Sistem pengaturan Motor DC banyak digunakan karena operasinya yang halus (*smooth*). Untuk penggunaan yang umum, Motor DC diklasifikasikan menurut cara eksitasinya: eksitasi seri, *shunt*, dan eksitasi terpisah yang

mana semuanya tergantung pada cara eksitasi medannya. Tetapi, karakteristik Motor DC *shunt* sangat tidak linear, sehingga untuk aplikasi sistem pengaturan eksitasi terpisah sangat populer dipakai.

Pengaturan Motor DC yang dieksitasi terpisah dibagi menjadi dua bagian yaitu pengaturan medan dan pengaturan jangkar. Untuk motor yang mempunyai magnet permanen biasanya menggunakan pengaturan jangkar. Dalam penelitian ini digunakan Motor DC yang mempunyai magnet permanen, sehingga yang akan dibahas hanya pengaturan jangkar.

Besaran yang dapatnya diteliti oleh para peneliti Motor DC adalah posisi, kecepatan dan percepatan. Pada sistem pengaturan posisi, motor berfungsi sebagai *controller*, artinya jika potensiometer masukan diubah, Motor DC akan berputar menggerakkan potensiometer keluaran dan akan berhenti jika posisi potensiometer keluaran sudah sama dengan posisi potensiometer masukan. Dalam sistem pengaturan posisi, diperlukan motor yang dapat bergerak dua arah (*bidirectional*), sesuai dengan tegangan yang masuk pada jangkar motor. Sebagai contoh bahwa jika tegangan masuk adalah positif, motor berputar ke kiri dan jika tegangan masuk adalah negatif, maka motor harus berputar ke kanan.

Pengaturan kecepatan dapat dilakukan dengan cara memberi sinyal masukan (tegangan) yang sesuai pada motor. Kecepatan motor dipengaruhi oleh besarnya tegangan masukan yang diberikan padanya. Sistem pengaturan yang demikian biasa disebut sistem pengaturan kecepatan *loop* terbuka. Dalam keadaan beban yang konstan atau dalam keadaan tanpa beban, sistem pengaturan dengan *loop* terbuka masih mungkin digunakan. Tetapi dalam keadaan beban yang berubah-ubah, sistem *loop* terbuka tersebut sudah tidak dapat dipertanggungjawabkan lagi keahwalannya.

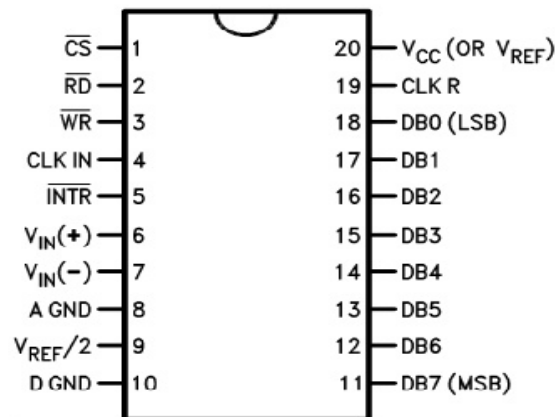
Sistem tersebut dapat diperbaiki dengan memodifikasinya menjadi sistem *loop* tertutup. Artinya kecepatan sesungguhnya di bandingkan dengan kecepatan yang diinginkan. Perbandingan tersebut akan menghasilkan sinyal *error* yang kemudian digunakan untuk menggerakkan motor melalui *servo amplifier* atau melalui *driver* motor. Dengan demikian motor dapat mempertahankan kecepatan yang konstan.

Pengaturan percepatan Motor DC didapat dengan men-*differential*-kan pengaturan kecepatannya terhadap waktu. Meskipun hal ini jarang dipakai tetapi dapat dilakukan.

*Plant* Motor DC terdiri dari suatu himpunan mesin dengan komponen-komponen penyusunnya adalah Motor DC dengan akuisisi nilai *actual* arus, *generator* DC dengan beban yang dapat dihubungkan, *tachogenerator* dengan penguat yang terkopel (*decoupling amplifier*), massa mekanik sentrifugal yang direalisasikan secara elektronik, *Angle-encoder* kanal ganda untuk akuisisi kecepatan dan arah putar, tersedianya fasilitas untuk *optical recognition*, dan *scanning stroboscopic*.

## ADC

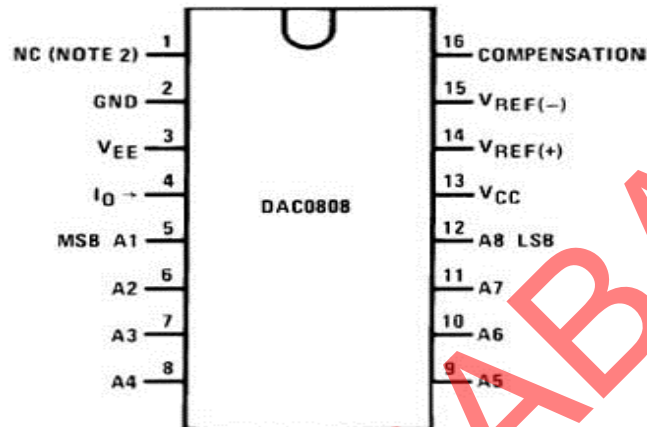
Untuk mengubah sinyal analog menjadi sinyal digital yang dapat diproses oleh komputer maka diperlukan alat untuk mengkonversi sinyal tersebut. Alat tersebut dikenal dengan nama *Analog to Digital Converter* (ADC), yang bentuk fisiknya berupa *Integrated Circuit* (IC). ADC yang dipakai dalam penelitian ini adalah ADC produksi ED Laboratory yang IC utamanya adalah IC ADC 0804. Gambar 2 adalah gambar konfigurasi PIN dari IC ADC 0804.



Gambar 2. Konfigurasi IC ADC 0804

## DAC

Agar dapat digunakan untuk mengendalikan Motor DC, sinyal digital yang berasal dari komputer harus dikonversikan menjadi sinyal analog terlebih dahulu, karena *driver* motor dan motor mengkonsumsi tegangan analog. Alat pengkonversi tersebut dikenal dengan nama DAC (*Digital to Analog Converter*). Gambar 3 merupakan IC DAC yang digunakan dalam penelitian ini.



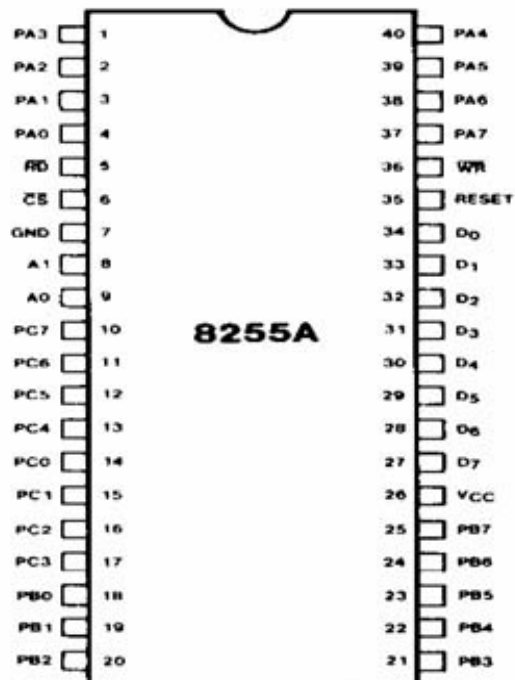
Gambar 3. Konfigurasi IC DAC 0808

## Programmable Peripheral Interface, PPI8255A

PPI 8255A atau secara umum disebut PPI 8255 saja, sesuai namanya bahwa peralatan ini dapat diprogram sesuai keinginan *user* (Gambar 4). Pengoperasian PPI 8255 mempunyai 3 buah mode yaitu: Mode 0, Mode 1, dan Mode 2. Pada mode 0, PPI 8255 bekerja sebagai Masukan / Keluaran biasa, sedang pada Mode 1 Port C digunakan sebagai pengendali dari Port A dan Port B. Mode 1 digunakan bila mikroprocessor berhubungan dengan peralatan yang mempunyai *access time* lambat. Karenanya diperlukan Port C sebagai jalur pemberitahuan pada Port A atau Port B bahwa peralatan luar sudah atau belum siap menerima atau mengirim data. Pada Mode 2, Port A dan Port B berfungsi sebagai data dua arah (*handshake*).

## Tachogenerator

*Tachogenerator* adalah suatu alat yang dapat merubah energi gerak (mekanis) menjadi energi listrik. *Tachogenerator* biasanya sudah terhubung pada bagian ujung (atas) Motor DC, hal ini memudahkan pengguna karena tidak harus menggunakan alat yang banyak untuk mengetahui keluaran Motor DC tersebut, tetapi keluaran dari *Tachogenerator* tersebut masih merupakan tegangan AC, sehingga diperlukan Rangkaian Penyearah untuk mendapatkan tegangan DC.



Gambar 4. Konfigurasi PPI 8255A

### Rangkaian Penyearah Dan Rangkaian Pelemah

Rangkaian penyearah berfungsi untuk merubah tegangan AC menjadi tegangan DC. Rangkaian ini memiliki 2 kelompok, yakni: penyearah setengah gelombang dan penyearah gelombang penuh.

Pada bentuk yang paling sederhana, penyearah setengah gelombang dapat dibuat dengan menghubungkan sebuah Dioda secara seri dengan catu daya AC dan tahanan beban. Disini Dioda hanya mengalirkan arus selama setengah gelombang catu daya AC. Kelemahan dari penyearah jenis ini pada tegangan beban, walaupun satu arah, tapi akan berubah-ubah dengan variasi yang tak dapat diabaikan, dan selain itu tegangan beban berharga nol pada setengah perioda. Keluaran tegangan DC dari rangkaian penyearah diharapkan se-rata mungkin, dan di haruskan tegangan beban tidak nol pada saat setengah gelombang sumber AC. Untuk menghilangkan nilai nol tersebut dapat digunakan sebuah Kapasitor yang di pasang secara paralel dengan beban, sehingga aliran arus akan memberi muatan pada Kapasitor, dan tegangan pada Kapasitor akan bertambah. Selama Dioda tak mengalirkan arus, Kapasitor akan melepaskan muatan melalui Resistor beban dan mencegah tegangan beban turun menjadi nol, sehingga di dapat tegangan DC yang mendekati garis lurus.

Pada penyearah gelombang penuh, sumber AC akan memanfaatkan kedua bagian gelombang, positif maupun negatif. Setengah gelombang negatif akan terbalik, sehingga di dapat gelombang satu arah (keduanya positif). Karenanya untuk menyearahkannya dibutuhkan 2 Dioda serta Kapasitor. Keuntungan yang didapat dengan komponen tersebut adalah lebih efisien, meski hanya sedikit menimbulkan magnetisasi DC, dan tegangan kerutnya mempunyai frekuensi dua kali frekuensi catudaya. Metode lainnya dengan menggunakan Rangkaian Jembatan (*Diode Bridge*), disini dibutuhkan 4 Dioda, Resistor dan Kapasitor.

### HASIL DAN PEMBAHASAN

Perancangan meliputi bagaimana menemukan besaran-besaran yang diperlukan dalam identifikasi sistem, yakni:

#### 1. *Time Delay System.*

Disini dicari nilai *L* (*time delay system*), yaitu waktu yang dibutuhkan oleh *plant* tersebut untuk bereaksi terhadap masukan yang diberikan, dihitung mulai dari titik nol. Algoritmanya seperti berikut :

```
for (int i=0;i< 30;i++)
{
    mid(string1, string2, z1, 5);
    z1 = z1 + 6;
    Vout = atof(string1);
    if ((Vout > 0) && (iL == 0))
        iL = i;
}
L = iL * 0.20;
```

Pertama data sebanyak 31 tersebut dibaca melewati prosedur *mid* kemudian tiap data tersebut saling dibandingkan. Data yang memenuhi syarat adalah data yang bernilai lebih dari nol dan nilai *iL* masih nol. Data inilah yang merupakan data pertama yang lebih dari nol, kemudian *iL* diisi dengan nilai *i* yang merupakan urutan data tersebut kemudian dikalikan dengan *ts* (*time sampling*) dan itulah *L* yang dicari.

#### 2. *Time Konstan dari Sistem*

Disini simbol *T* adalah waktu untuk mencapai tegangan sebesar  $[63,2 \% * \text{Respon Stabil}]$ . Respon stabil disini berarti respon dari Motor DC untuk mencapai nilai *keluaran* tegangan yang stabil. Pertama kali dicari nilai respon stabil tersebut, kemudian langkah selanjutnya adalah mengalikan respon stabil dengan 63,2% sehingga di peroleh tegangan yang bersesuaian dengan *time* konstan, langkah selanjutnya adalah mencari letak *time* konstan tersebut di detik ke berapa.

#### 3. *Time Settling dari Sistem*

Simbol *Ts* menyatakan waktu yang dibutuhkan untuk mencapai nilai respon stabil pertama kali. Disini dicari waktu dimana *Vout* (tegangan keluaran) pertama kali mencapai nilai respon stabil, untuk caranya dapat dilihat program untuk menemukan nilai *time constant*. (digabung dengan program untuk mencari nilai *time* konstan), sebagai :

- if ((VTemp == Vout)&&(i4RS == 0)) i4RS = ii;  
float T4Ts = i4RS\*0.20;
4. Konstanta Penguatan (K)  
Disini dipakai rumus menemukan nilai K sebagai  $K = (\text{Keluaran akhir} - \text{Keluaran awal}) / \text{Amplitudo}$  dari Masukan *Unit Step*. Untuk bagian ini tidaklah sulit dan dapat digabung dengan program untuk mencari nilai *time constant*, caranya sebagai berikut :  
if (ii == 30) V4End = Vout;  
if (ii == 1) V4Start = Vout;  
Gain = (VTemp-V4Start)/Vin;
5. *Ratio* antara *Time Delay* dengan *Time Settling*  
Rumus yang digunakan sebagai berikut :  $L/Ts$ . Disini tidaklah sulit karena nilai L dan Ts sudah didapat dan langsung dapat dipakai, penggalan programnya seperti berikut :  
 $L\text{Div}Ts = L / T4Ts$
6. Tabel STREJC (Strejc, 2004) dipakai dalam penentuan letak *Ratio* yang didapat dari Langkah 5: sebagai jawaban dari pertanyaan 'Lebih dekat ke nilai manakah nilai *Ratio* ?' (dibandingkan dengan kolom Tu/Ta pada Tabel STREJC).  
Tabel 1 adalah Tabel Strejc (2004) yang dipakai setelah menemukan perbandingan antara *time delay* dengan *time settling*.

Tabel 1. Tabel Strejc ( $\tau = 0$ )

N	Tu/Ta	Tu/T	Ta/T	Y1
1	0	0	1	0
2	0.104	0.282	2.718	0.264
3	0.218	0.805	3.695	0.323
4	0.319	1.425	4.465	0.353
5	0.404	2.100	5.119	0.371
6	0.493	2.811	5.699	0.384
7	0.570	3.549	6.226	0.394
8	0.642	4.307	6.711	0.401
9	0.709	5.081	7.164	0.407
10	0.773	5.869	7.590	0.413

Dari Langkah ini didapat nilai-nilai: N = Orde Sistem, [Ta/T], [Tu/T], dan [Tu/Ta]. Langkah untuk mencari nilai yang mendekati dengan yang ada di tabel ini membutuhkan urutan program yang panjang, nilai *ratio* (langkah 5) di bandingkan dengan nilai pada kolom Tu/Ta yang mendekati dari Tabel STREJC tersebut.

7. Nilai *Time Konstan* yang Baru

Disini mempunyai rumus sebagai berikut  $T_{baru} = \text{Time Settling} / [Ta/T]$ , nilai - nilai yang dibutuhkan untuk langkah ini sudah didapat tinggal menghitungnya, potongan programnya adalah sebagai berikut :

$$\text{NewT} = T4Ts / Ta4T;$$

8. Nilai *Time Delay* yang Baru

Disini mempunyai rumus sebagai berikut  $L_{baru} = \text{Time Settling} \times [Tu/T]$ , nilai yang dibutuhkan sudah ada tinggal dimasukkan dalam perhitungan, algoritmanya sebagai berikut :

$$\text{NewL} = T4Ts / Tu4T;$$

9. Nilai Lamda

Disini mempunyai rumus sebagai berikut  $\text{Lamda} = \text{Time Delay} - \text{Time Delay Baru}$ . Potongan programnya sebagai berikut :

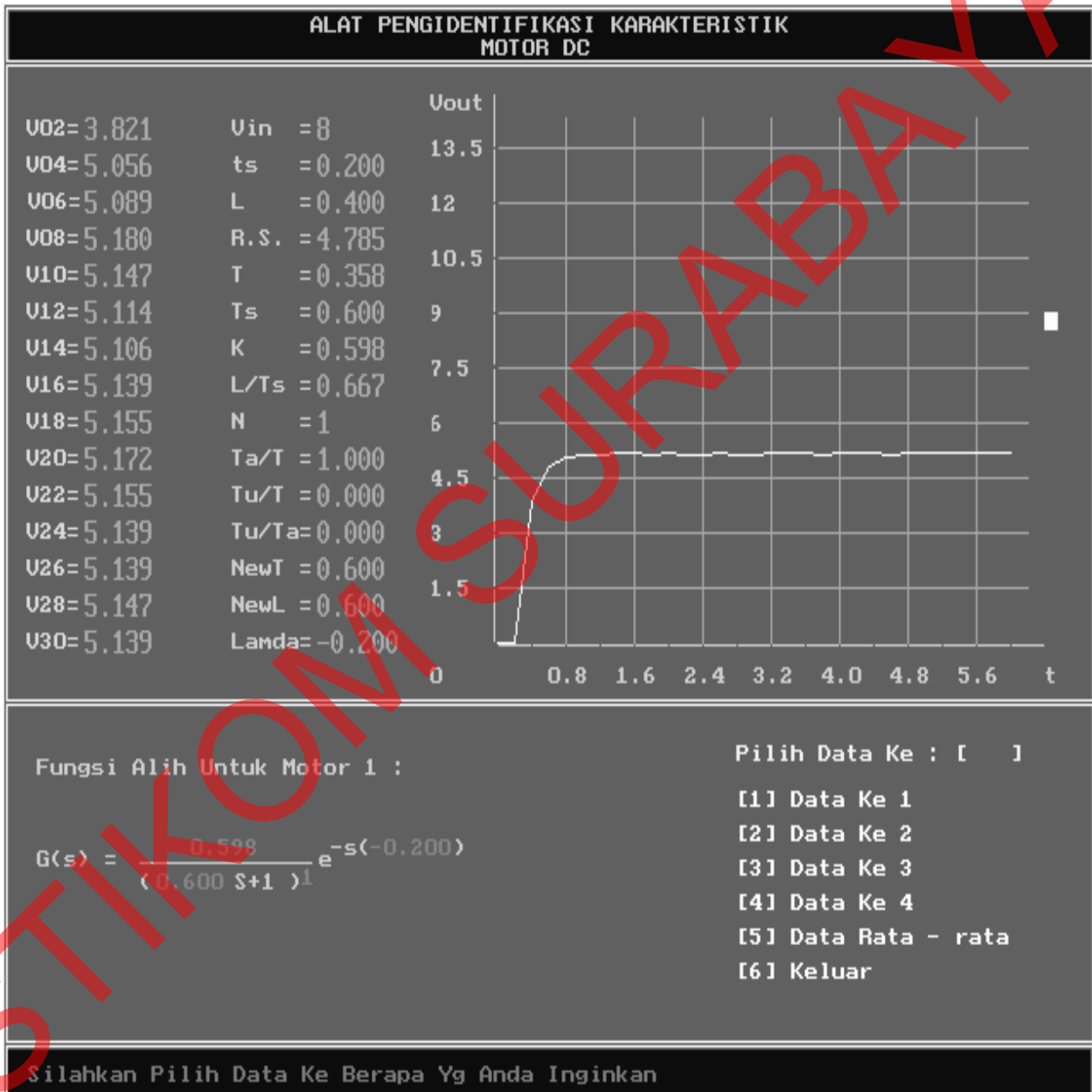
$$\text{Lamda} = L - \text{NewL};$$

10. Pilih Struktur Model yang sesuai sesuai orde yang di dapat.

$$G(s) = \frac{K}{(Ts + 1)^n} e^{-sL}$$

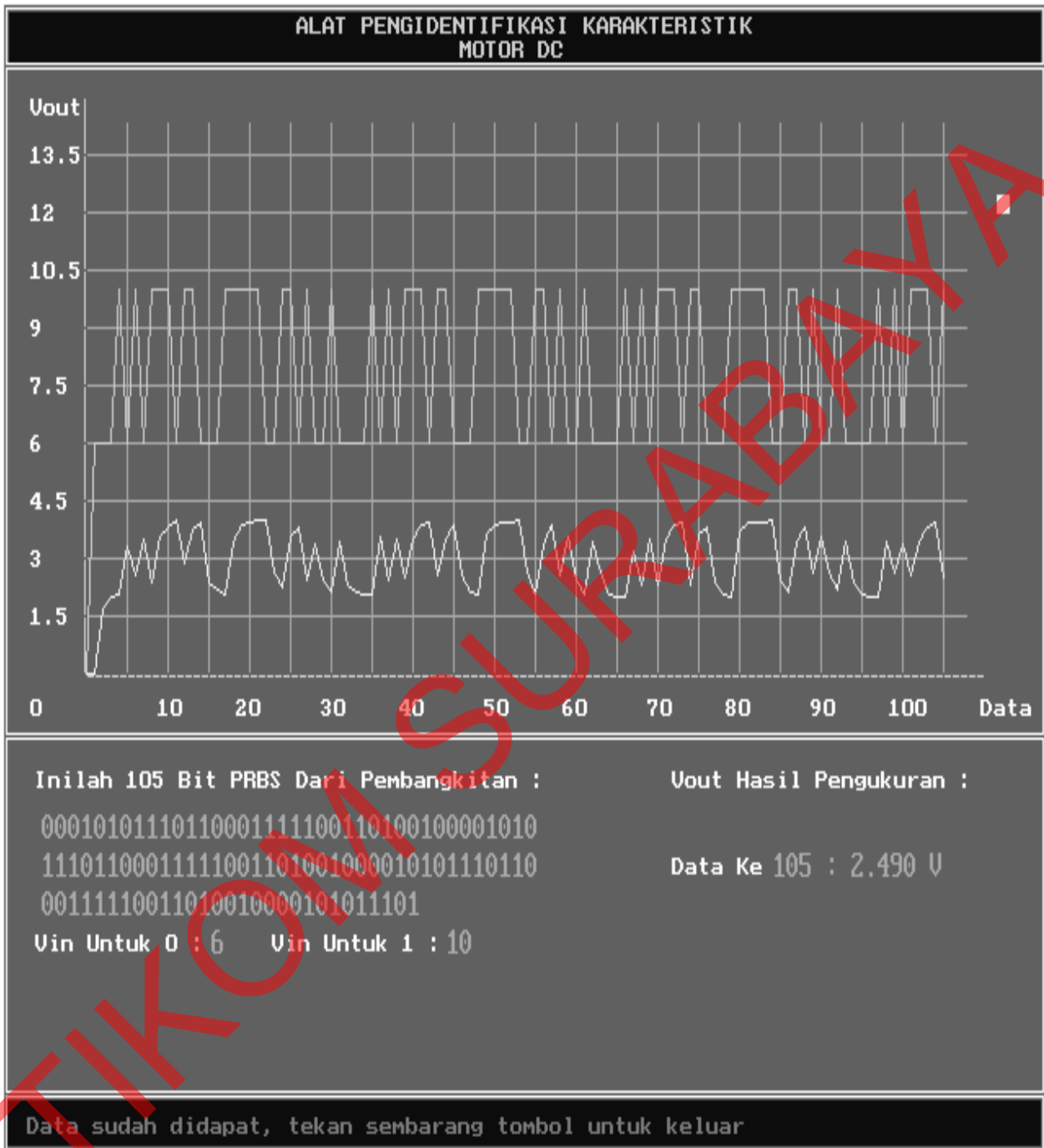
Disinilah langkah dari analisa data berakhir, semua komponen yang membangun suatu persamaan fungsi alih sudah didapat tinggal menempelkan pada bentuk persamaan yang ada, disini bentuk persamaan ditentukan oleh *orde* hasil perhitungan dari tabel Strejc.

Dari gambar 5 dan 6 dapat secara langsung diperoleh persamaan dinamika dari Motor DC yang ditinjau. Desainer kontroler dapat menggunakannya dengan sangat baik.



Gambar 5. Hasil prosedur analisa beserta hasilnya untuk identifikasi statis.





Gambar 6. Hasil prosedur mengambil data untuk identifikasi secara dinamis.

#### SIMPULAN

Telah dirancang dan dibuat sebuah alat pengidentifikasi karakteristik Motor DC, dan telah diujikan untuk mengidentifikasi dua buah Motor DC. Untuk identifikasi secara statis hasilnya dapat langsung di peroleh pada program alat tersebut (visualisasi di monitor).

Alat yang dirancang belum menyertakan nilai performansi dari peralatan yang diintegrasikan, seperti halnya bila membeli sebuah alat ukur dipasaran.

Disarankan agar kiranya juga mengeksplorasikan program komputasi cerdas (*soft computing*) pada tahapan berikut dari penelitian ini yakni: pemakaian sistem fuzzy dan jaringan syaraf tiruan.

## DAFTAR RUJUKAN

- , 2004. *Control Tutorial For MATLAB*, (online), url: <http://www.mame.mu.oz.au/~meg/ctrl301/matlab/ctm>, diakses 20 Juni 2004.
- , 2004. *Euclidean space*, (online), url : [http://www.en.wikipedia.org/wiki/euclidean\\_space](http://www.en.wikipedia.org/wiki/euclidean_space), diakses 23 Juli 2004
- , 2004. *Buku informasi Bahan Pelatihan Nasional, 2002*, url : [http://www.kota-makassar.net/otomotif/logam/automotif/learning&assessment\\_material/electrical\\_/50-003/50-003-3/50-003-3-k.doc](http://www.kota-makassar.net/otomotif/logam/automotif/learning&assessment_material/electrical_/50-003/50-003-3/50-003-3-k.doc), diakses 20 Januari 2004.
- Astrom, K.J., & Wittenmark, B. 1997. *Computer Controlled system theory and design*. New Jersey : Prentice Hall International Editions.
- Franklin, G.F., & Powell, J.D. 1985. *Digital Control of Dynamic System*. Massachusetts: Addison-Wiley Publishing Company.
- Gray, A. 1997. Euclidean Spaces. §1.1 in *Modern Differential Geometry of Curves and Surfaces with Mathematica, 2nd ed.* Boca Raton, FL: CRC Press, pp. 2-5, 1997.
- Green, D.C. 1987. *Pedoman Elektronika 2*. Jakarta : PT. Elex Media Komputindo.
- Jogiyanto. 1993. *Konsep Dasar Pemrograman Bahasa C*. Yogyakarta : Andi Offset.
- Jamsa, K. 1996. *1001 Tip C/C++*. Yogyakarta : Andi Offset
- Junaedi, A. 1997. *Perancangan Kontroler PID Adaptif Melalui Pendekatan Analitik untuk Pengaturan Kecepatan Motor DC*. Surabaya: Tugas Akhir Jurusan Teknik Elektro Fakultas Teknologi Industri ITS.
- Kadir, A. 1995. *Menggunakan TurboC++ dan Borland C++*. Yogyakarta : Andi Offset.
- Malvino, A.P. 1994. *Elektronika Komputer Digital*. Jakarta : Penerbit Erlangga.
- Sedra, A.S. 1990. *Rangkaian mikroelektronik*. Jakarta : Penerbit Erlangga.
- Strejc. 2004. *Methode D'identification De Strejc*, (online), url : <http://www.eudil.fr/eudil/belk/pral.htm>, diakses 23 Juli 2004.
- Tirtamiharja, S. 1996. *Elektronika digital*. Yogyakarta : Andi Offset
- Woollard, B. 1988. *Elektronika Praktis*. Jakarta : PT Pradnya Paramita.