

BAB II

TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian terkait yang telah dilakukan sebelumnya dan terdapat penjelasan tentang dasar teori yang dipakai dalam mengerjakan Tugas Akhir ini. Dasar teori tersebut meliputi : dinamika sistem *Pressure Process Rig* (38-714), permodelan matematis *Pressure Process Rig* (38-714), identifikasi sistem, pengendali *Linear Quadratic Regulator* (LQR), pengendali *Proportional Integral Derivatif* (PID), dan perangkat lunak MATLAB.

2.1 Penelitian Terkait

Dalam penelitian Tugas Akhir ini dilakukan studi literatur yang merupakan pencarian teori serta referensi yang relevan dengan kasus dan permasalahan yang akan diselesaikan, teori dan referensi didapat dari jurnal, *paper*, dan sumber lainnya.

Penelitian yang dilakukan oleh Vector Anggit Pratomo, yang berjudul perancangan pengendali PID pada *Pressure Process Rig* (38-714) berbasis Microcontroller AVR ATmega8535. Pada penelitian ini kelemahan terdapat pada respon waktu yang masih lambat dan masih bisa ditingkatkan lagi saat perubahan tekanan dimana $K_c = 3.1654$, $T_i = 0.2962$, $T_d = 0.0156$ dan juga masih dibutuhkan banyak modifikasi dari block sistem yang dirancang pada simulasi matlab agar respon sistem mendapatkan hasil yang lebih presisi dan didapatkan data berupa $t_r = 1.8s$, $t_s = 0.819s$, $t_d = 1s$. [2]

Penelitian pada sistem *steam drum* untuk mengendalikan *level* menggunakan pengendali optimal *Linear Quadratic Regulator* (LQR) menunjukkan bahwa pengendali optimal LQR didasari pada kemampuannya dalam mempercepat respon sistem, LQR mampu membuat sistem mencapai *set point* pada detik ke 1.5 dari total waktu simulasi selama 10 detik. Namun, masih terdapat error sebesar 0.02% [3].

Pada penelitian yang dilakukan oleh Ikhwanul Kholis yang berjudul *Pemodelan Sistem dan Desain Pengendali PID dengan Metode Ciancone Dengan Menggunakan Matlab Simulink Pada Sistem Pressure Process Rig 38-714* menunjukkan bahwa transien response dan steady state errornya tidak stabil dan membutuhkan waktu selama 1.91 detik untuk mencapai setpoint agar respon waktunya optimal untuk mengikuti setpointnya dimana data diperoleh dari hasil penelitian ini adalah $K_c = 2.79$, $T_i = 0.168$, dan $T_d = 0.105$. [6]

Penelitian selanjutnya membahas tentang pengendalian rasio bahan bakar dan udara pada sistem *steam drum boiler* menggunakan pengendali LQR, yang difokuskan pada pengendalian tekanan dan level pada steam drum boiler. Untuk pengendalian level sangat

cepat sehingga sistem stabil sesaat sudah steam drum boiler mulai bekerja. Artinya pengendali optimal LQR mampu mempercepat respon system. Untuk itu, pengendali LQR bisa diterapkan untuk mengendalikan tekanan pada modul Pressure Process Rig 38-714.

Setelah pengujian dilakukan pada modul training Pressure Process Rig 38-714 menggunakan pengendali LQR secara simulasi, diperoleh respon keluaran belum sesuai karakteristik LQR yang mampu mempercepat respon waktu. Respon pengendalian tekanan yang dihasilkan LQR mempunyai *settling time* (t_s) sebesar 10,645s, *rise time* (t_r) sebesar 6,269s, *delay time* (t_d) sebesar 5,2906 dan *error steady state* (ess) sebesar -18009 dengan total waktu simulasi selama 50 detik. Besar kemungkinan hal ini disebabkan oleh karakteristik sistem yang berbeda. Penyelesaian permasalahan ini dilakukan dengan menambahkan pengendali PID, karena PID memiliki efek Proporsional yang mampu mempercepat respon, memiliki Integral yang mampu menghilangkan error dan memiliki derivatif yang mampu meredam osilasi.

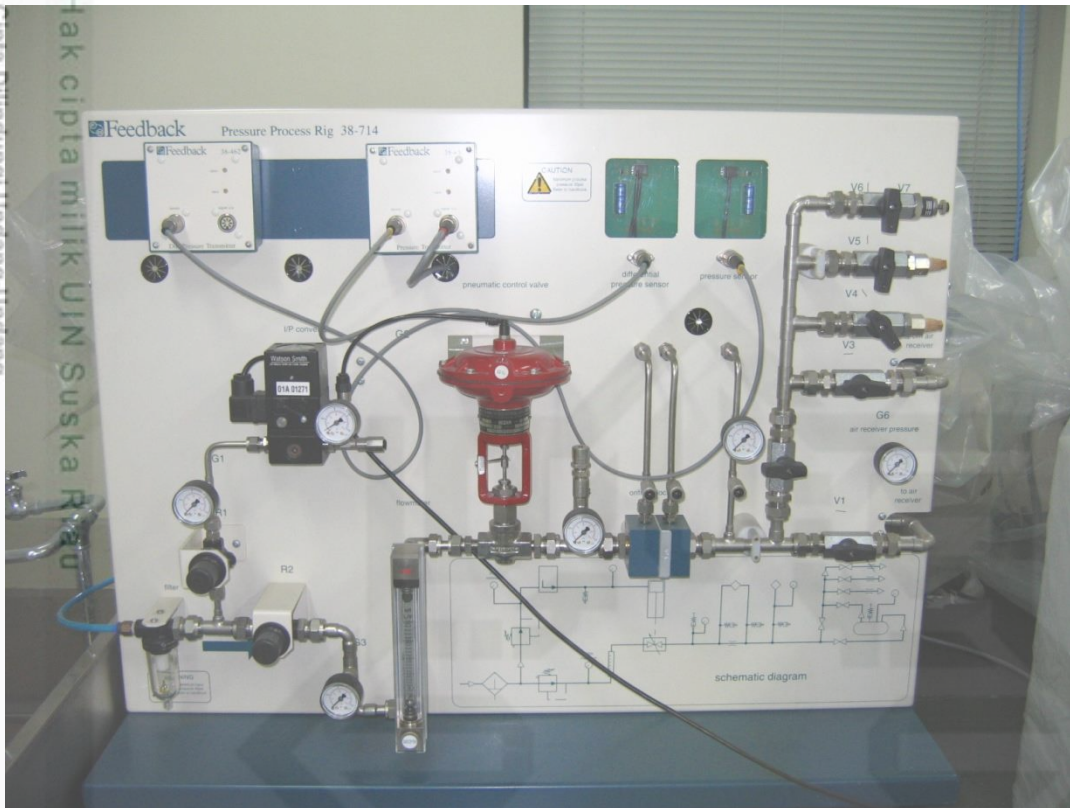
Berdasarkan penelitian terkait dan pengujian sistem modul training Pressure Process Rig 38-714 menggunakan pengendali LQR, menunjukkan bahwa pengendali LQR memiliki kemampuan dalam mempercepat respon sistem, tetapi pada sistem Pressure Process Rig 38-714 respon keluaran yang dihasilkan oleh pengendali LQR belum sesuai karakteristik LQR yang mampu mempercepat respon waktu. Dengan adanya permasalahan tersebut penulis akan mengkombinasikan pengendali optimal LQR dan pengendali PID agar kinerja LQR menjadi lebih baik dan optimal. Optimal difokuskan pada respon waktu yang cepat dengan error minimum. Sehingga penulis tertarik melakukan penelitian dengan judul “Perancangan Kendali LQR-PID Untuk Pengendali Tekanan Pada Modul Training Pressure Process Rig 38-714”.

2.2 Dasar Teori

2.2.1 Pressure Process Rig 38-714

Pressure Process Rig 38-714 merupakan peralatan produksi dari PROCON yang digunakan untuk mengenalkan dan mendemonstrasikan prinsip-prinsip proses pengukuran dan pengendalian. Sistem yang disediakan adalah sistem pneumatic sebagaimana yang digunakan dalam industri. Secara umum Pressure Process Rig 38-714 terdiri atas jalur-jalur pipa yang terhubung pada *Pneumatic Control Valve*, *Orifice Block*, *Flowmeter*, *pressure tapping*, *regulator*, indikator tekanan, dan *valve*. Regulator R1 digunakan untuk mengendalikan tekanan yang diukur oleh G1. Regulator R2 digunakan untuk mengatur tekanan yang diukur oleh G3 atau G4 atau G5. Sementara indikator tekanan G6 digunakan

untuk menunjukkan tekanan pada air receiver. Output yang dikendalikan pada penelitian ini adalah *flow* dari Pressure Process Rig.



Gambar 2.1 Pressure Process Rig 38-714 [Dong-A University Process Lab]

Pressure Process Rig tersusun dari beberapa komponen, yaitu Kompresor, *I/P Converter*, *Control Valve*, *Pressure Sensor*, *Differential Pressure Process*, *Manual Valve*, PC (yang di dalamnya terdapat ADC, DAC, dan PCI Card), *Gauge*, dan *Regulator*. Kompresor merupakan pompa bertekanan yang berfungsi sebagai penyuplai udara ke semua saluran pipa. Kemudian, *I/P Converter* digunakan untuk mengubah arus listrik menjadi tekanan. Aliran udara akan bergerak melewati regulator, *valve*, pipa, *pressure sensor*, dan *gauge*. Tekanan udara yang dibaca oleh *pressure sensor* akan diteruskan ke *I/P Converter* kemudian diubah ke Data Digital dengan ADC sehingga diperoleh data di PC.

Pressure Process Rig digunakan untuk meneliti pengendalian proses, yaitu pengukuran *controlled variable (CV)*, membuat grafik keluaran, dan percobaan kontroler yang telah didesain. CV yang digunakan adalah tekanan dan kecepatan aliran udara. Pada akhirnya, penggunaan Pressure Process Rig dilakukan untuk mencoba kontroler yang didesain agar CV dapat mencapai nilai yang diharapkan dengan *error steady-state* yang kecil.

Procon 38 Series System adalah peralatan yang digunakan untuk menunjukkan segala sesuatu tentang pengendalian proses. Pada panel depan *trainer* yang diperlihatkan pada Gambar 2.1, terdapat skema yang menunjukkan komponen-komponen utama beserta

instalasinya dalam notasi pneumatik. Pada *trainer* terdapat *pneumatic control valve*, *orifice block*, dan *pressure tappings* yang terhubung pada saluran pipa udara. Setelah melewati keseluruhan proses, aliran udara dapat dibuang langsung keluar atau ditampung ke *air receiver* yang terdapat pada bagian belakang *trainer* untuk pengamatan respon proses yang lebih lambat. Pemasangan *air receiver* dapat dilakukan secara seri ataupun paralel. Pengoperasian *process control trainer* 38-714 memerlukan beberapa peralatan pendukung lainnya yaitu *Process Interface* 38-200, *Process Controller* 38-300, *Pressure Transmitter* 38-461, *Differential Pressure Transmitter* 38-462, *Digital Display Module* 38-490 dan *Compressor Unit* 38-820. Katup pneumatik pada *trainer* dioperasikan dari *current to pressure converter* (I/P) yaitu alat pengubah sinyal listrik menjadi sinyal pneumatik. *I/P Converter* menerima sinyal kontrol sebesar 4-20 mA dari *process interface* 38-200 dan mengubahnya menjadi sinyal pneumatik sebesar 3-15 psi. Berikutnya adalah sensor *direct pressure* untuk mengukur tekanan udara dan sensor *differential pressure* sebagai pengukur aliran udara (*flow*). Masing-masing sensor tersebut sudah dilengkapi dengan rangkaian pengkondisi sinyal (*signal conditioning* 38-461 and 38-462) agar sesuai dengan tegangan dan arus kerja *Process Interface* 38-200.

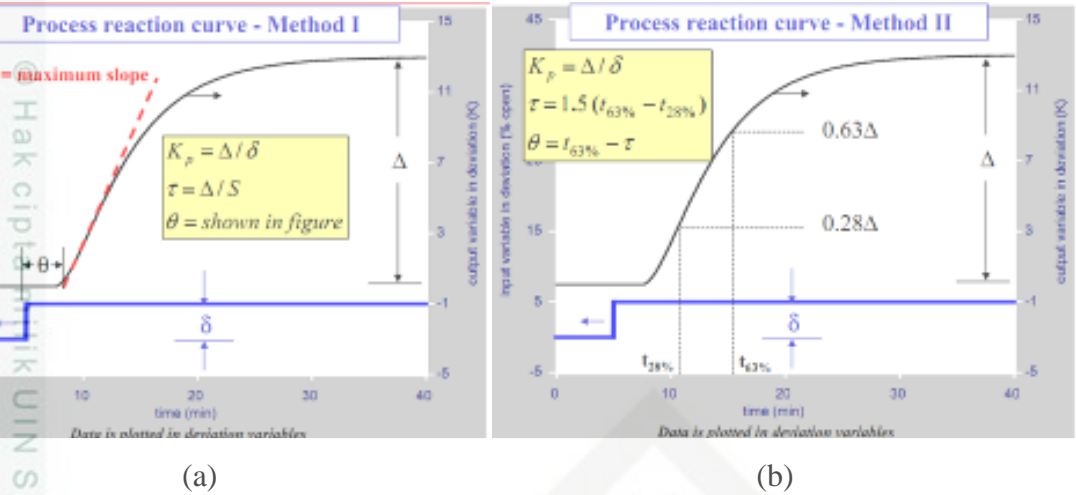
2.2.2 Komponen Sistem

Agar dapat bekerja untuk mengatur tekanan yang diinginkan, *The Pressure Process Rig* 38-714 memiliki beberapa komponen penyusun yang saling berhubungan yang satu dengan yang lainnya yaitu, **kompresor**. Kompresor merupakan suatu komponen yang berfungsi sebagai penyuplay udara bertekanan pada plant pressure. kompresor tersebut dapat diatur tekanan yang akan diperlukan. Tekanan yang diberikan pada sistem tersebut sebesar 35 psi. *I/P Converter* digunakan untuk mengubah masukan berupa arus listrik sebesar 4-20 mA kedalam tekanan udara sebesar 3-15psi. *I/P converter* tersebut memiliki *hysteresis* yang kecil. Keluaran alat ini digunakan untuk menggerakkan *pneumatic control valve*. ***Pneumatic Control Valve*** digunakan sebagai gerbang pembuka atau penutup tekanan udara. Alat ini menerima masukan sebesar 3-15 psi dari *I/P converter*. Mode dari *pneumatic control valve* tersebut adalah *normally open* artinya ketika tidak disuplai tekanan udara maka gerbang akan terbuka. Ketika alat tersebut menerima tekanan udara sebesar 3 psi maka gerbang akan membuka penuh dan jika diberi 15 psi maka gerbang akan menutup penuh. ***Manual Valve*** adalah katup-katup manual yang terpasang di saluran udara untuk mengatur arah aliran udara. ***Manual valve*** ini juga dapat difungsikan untuk keperluan pembebebanan dengan mengatur bukaan *valve* secara manual. ***Pressure Regulator***, Komponen ini digunakan untuk mengatur nilai tekanan udara agar sesuai yang dibutuhkan oleh komponen sesudahnya. Dalam sistem

ini, terdapat dua *pressure regulator*. Regulator pertama diatur sebesar 25 psi untuk menggerakkan *pneumatic control valve* sedangkan regulator kedua sebesar 10 psi untuk keperluan pengaturan tekanan dan kecepatan alir. **Gauge**, alat ini digunakan untuk mengetahui besarnya tekanan udara yang masuk kedalam sistem tersebut. **Flow Meter**, alat ini digunakan untuk mengetahui besarnya kecepatan aliran udara yang masuk kedalam sistem tersebut. **Differential Pressure Sensor**, dua sensor tekanan yang dipasang berdampingan pada jarak tertentu (relatif dekat) dalam satu saluran yang diberi sekat dengan *orifice plate*. Perbedaan tekanan yang terukur adalah informasi untuk mengetahui aliran udara. **Pressure Sensor** merupakan suatu komponen yang berfungsi untuk merubah besaran fisik berupa tekanan udara kedalam besaran listrik berupa tegangan. Besaran listrik tersebut kemudian diteruskan ke *pressure transmitter*. **Air Receiver** adalah penampung udara yang dialirkan dan dapat juga digunakan sebagai tempat yang diukur tekanannya. **Pressure Transmitter** adalah komponen yang digunakan untuk menghasilkan besaran listrik berupa arus sebesar 4-20 mA dari *pressure sensor*. **Differential Pressure Transmitter** adalah komponen yang digunakan untuk menghasilkan besaran listrik berupa arus sebesar 4-20 mA dari *Differential Pressure sensor*. **Digital Display Modul** (DDM) merupakan sesuatu komponen untuk mengetahui nilai dari keluaran terukur suatu sistem. Nilai tersebut berupa arus listrik 4-20 mA atau berupa persentase 0-100%. **Process Controller 38-300**, merupakan suatu komponen yang berfungsi sebagai kontroler. Selain sebagai kontroler, komponen ini berfungsi sebagai *set point* dan juga pengamatan nilai terukur atau sebagai *supervisory control*. **Process Interface 38-200**, merupakan suatu komponen yang digunakan untuk penghubung antara kontroler dengan plant. *Process interface* tersebut mempunyai fitur *span* dan *zero* untuk keperluan kalibrasi. Selain sebagai pengkonversi besaran, komponen tersebut mempunyai fungsi yang lainnya yaitu sebagai *dc supply* untuk mengatur *solenoid valve*. [7]

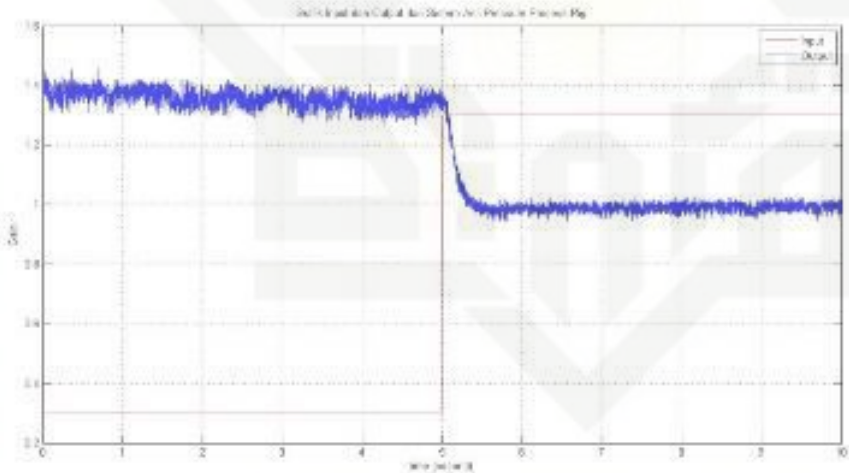
2.2.3 Identifikasi Sistem dengan *Process Reaction Curve*

Process Reaction Curve (PRC) merupakan metode yang digunakan untuk mengidentifikasi suatu sistem dengan menggunakan grafik respon sistem tersebut. Sistem Pressure Process Rig berupa orde satu. Sehingga data persamaan PRC orde satunya yang diambil dari referensi jurnal Ikwanul Kholis sebagai berikut:



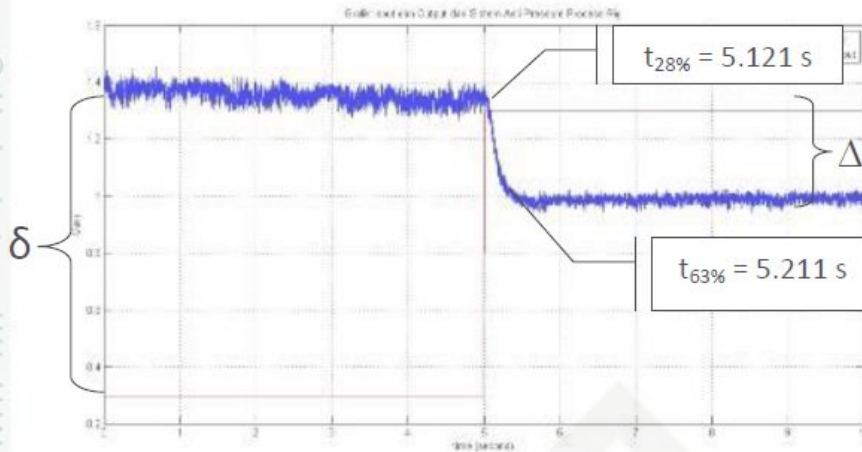
Gambar 2.2 (a) PRC Metode 1, (b) PRC Metode 2[5]

Pengambilan data pada Pressure Process Rig untuk dilakukan permodelan sistem tersebut. Pengambilan data dilakukan dengan memberikan input step dengan initial value sebesar 0.3 dan final value sebesar 1 dengan perubahan step dilakukan pada detik ke-5. Data tersebut diolah menjadi grafik response sistem sebagai berikut.



Gambar 2.3 Respon Sistem Pressure Process Rig [5]

Dari gambar di atas dapat dilakukan pemodelan sistem sebagai berikut :



Gambar 2.4 Permodelan sistem Pressure Process Rig dengan Process Reaction Curve [6]

Dengan menggunakan Gambar di atas, diberikan persamaan Process Reaction Curve untuk memperoleh *transfer function* permodelan system *Pressure Process Rig*.

$$\begin{aligned} \Delta &= \text{Final Value} - \text{Initial Value} & (2.1) \\ &= 1 - 1.351 \\ &= -0.351 \end{aligned}$$

$$\begin{aligned} \text{Value } 63\% &= \text{Initial Value} + (\Delta \times 63\%) & (2.2) \\ &= 1.351 + (-0.351 \times 63\%) \\ &= 1.351 + (-0.22) \\ &= 1.129 \end{aligned}$$

$$\begin{aligned} \text{Value } 28\% &= \text{Initial Value} + (\Delta \times 28\%) & (2.3) \\ &= 1.351 + (-0.351 \times 28\%) \\ &= 1.351 + (-0.098) \\ &= 1.252 \end{aligned}$$

$$\begin{aligned} \tau &= 1.5 (t_{63\%} - t_{28\%}) & (2.4) \\ &= 1.5 (0.211 - 0.121) \\ &= 1.5 (0.09) \\ &= 0.135 \end{aligned}$$

$$\begin{aligned} \theta &= t_{63\%} - \tau & (2.5) \\ &= 0.211 - 0.135 \\ &= 0.076 \end{aligned}$$

$$Kp = \frac{\Delta}{\delta} \quad (2.6)$$

$$\begin{aligned}
 &= \frac{-0.351}{1.3-0.3} \\
 &= \frac{-0.351}{1} \\
 &= -0.351 \\
 G(s) &= \frac{Kp}{\tau s + 1} e^{-\theta s} \\
 &= \frac{-0.351}{0.135s+1} e^{-0.076s}
 \end{aligned} \tag{2.7}$$

Dengan menggunakan persamaan di atas diperoleh data sebagai berikut :

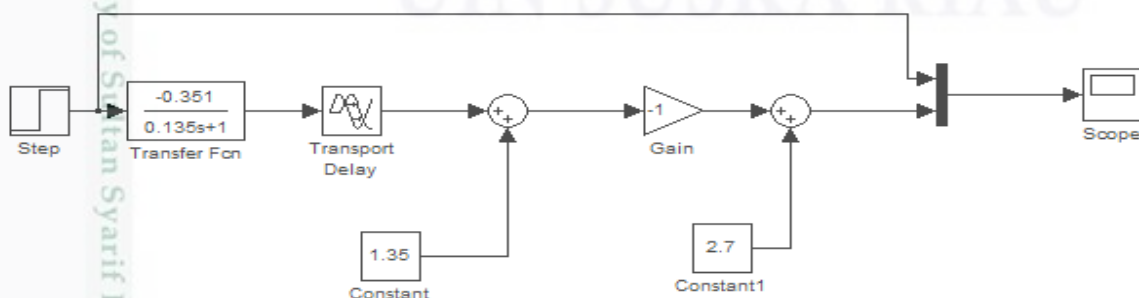
Deskripsi	Nilai
Δ	-0.351
Value(63%)	1.129
Value(28%)	1.252
τ	0.135
t(63%)	0.211
t(28%)	0.121
θ	0.076
Kp	-0.135

Tabel 2.1 Data Perhitungan *Process Reaction Curve* dari Grafik Gambar 2.4

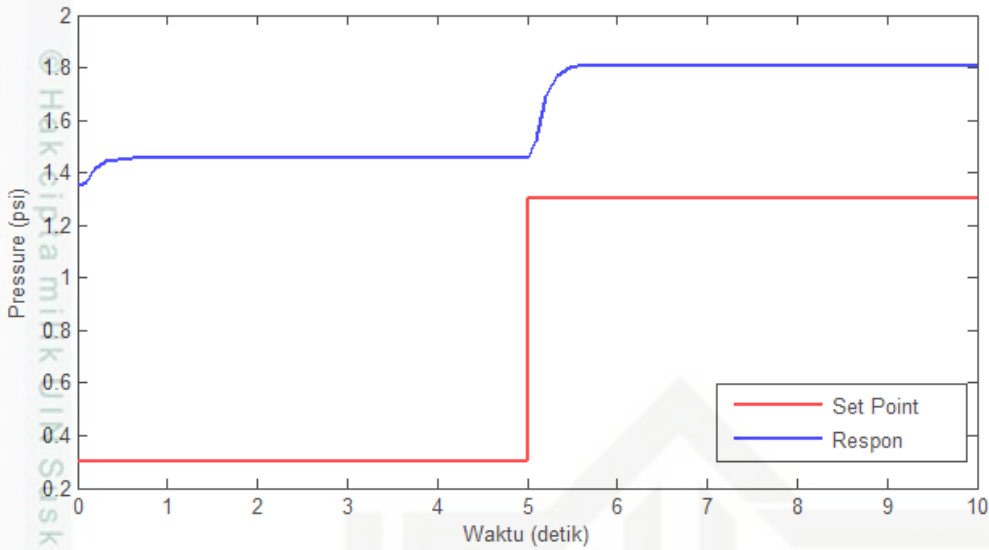
Berdasarkan tabel 1, diperoleh dtransfer function sebagai berikut :

$$\begin{aligned}
 G(s) &= \frac{Kp}{\tau s + 1} e^{-\theta s} \\
 &= \frac{-0.351}{0.135s+1} e^{-0.076s}
 \end{aligned} \tag{2.8}$$

Kemudian dilakukan modifikasi sistem agar output mengikuti grafik input dengan menambahkan konstanta yang memiliki besar dua kali konstanta awal. Hal ini dilakukan agar sistem memiliki *initial value* yang sama dengan yang diperoleh dan sistem menghasilkan *output* yang mengikuti grafik *input*. Berikut adalah gambar blok sistem pemodelan yang telah dimodifikasi;



Gambar 2.5 Diagram block hasil modifikasi



Gambar 2.6 Grafik hasil pada gambar diagram *block* 2.5

Sehingga dari diagram block di atas diperoleh data sebagai berikut:

$$\Delta = \text{Final Value} - \text{Initial Value} \quad (2.9)$$

$$= 1.807 - 1.456$$

$$= 0.351$$

$$\text{Value } 63\% = \text{Initial Value} + (\Delta \times 63\%) \quad (2.10)$$

$$= 1.456 + (0.351 \times 63\%)$$

$$= 1.456 + (0.22)$$

$$= 1.676$$

$$\text{Value } 28\% = \text{Initial Value} + (\Delta \times 28\%) \quad (2.11)$$

$$= 1.456 + (0.351 \times 28\%)$$

$$= 1.456 + (0.098)$$

$$= 1.554$$

$$\tau = 1.5 (t_{63\%} - t_{28\%}) \quad (2.12)$$

$$= 1.5 (0.211 - 0.121)$$

$$= 1.5 (0.09)$$

$$= 0.135$$

$$\theta = t_{63\%} - \tau \quad (2.13)$$

$$= 0.211 - 0.135$$

$$= 0.076$$

$$Kp = \frac{\Delta}{\delta} \quad (2.14)$$

$$\begin{aligned}
 &= \frac{0.351}{1.3-0.3} \\
 &= \frac{0.351}{1} \\
 &= 0.351 \\
 G(s) &= \frac{Kp}{\tau s + 1} e^{-\theta s} \\
 &= \frac{0.351}{0.135s+1} e^{-0.076s}
 \end{aligned} \tag{2.15}$$

Dengan menggunakan persamaan di atas diperoleh data sebagai berikut :

Deskripsi	Nilai
Δ	-0.351
Value(63%)	1.676
Value(28%)	1.554
τ	0.135
t(63%)	0.211
t(28%)	0.121
θ	0.076
Kp	0.135

Tabel 2.2 Data Perhitungan *Process Reaction Curve* dari Grafik Gambar 2.6

Sehingga diperoleh transfer function sebagai berikut:

$$\begin{aligned}
 G(s) &= \frac{Kp}{\tau s + 1} e^{-\theta s} \\
 &= \frac{0.351}{0.135s+1} e^{-0.076s}
 \end{aligned} \tag{2.16}$$

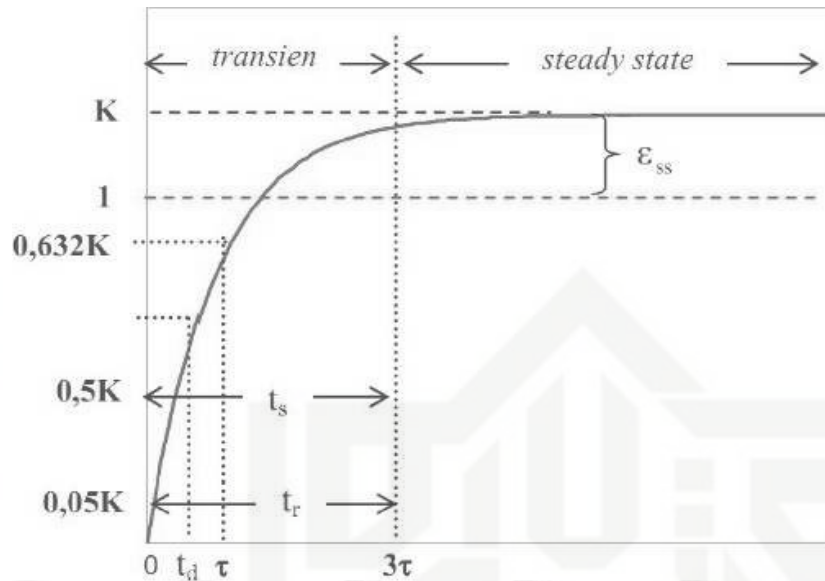
2.2.4 Identifikasi Sistem

Pengaturan proses tekanan merupakan sistem berorde satu, oleh sebab itu metode identifikasi yang dilakukan adalah metode identifikasi statis. Metode identifikasi dilakukan dengan pendekatan grafis, di mana sinyal uji diberikan pada sistem untuk mengetahui respon *open loop* sistem. Dari respon sistem, dapat diketahui karakteristik-karakteristik penting dari system.

Salah satu metode identifikasi statis adalah metode pengamatan respon waktu. Identifikasi sistem dengan metode ini bekerja berdasarkan pengamatan grafis terhadap masukan *step*. Karakteristik respon waktu untuk sistem orde pertama diberikan berdasarkan respon sistem terhadap masukan sinyal *step*. Karakteristik respon waktu sistem orde pertama dibedakan menjadi karakteristik respon transien dan keadaan tunak (*steady*

state). Grafik respon sistem orde pertama untuk $X_{ss} = 1$ dan $Y_{ss} = K$ ditunjukkan pada

Gambar 2.3.



Gambar 2.7 Respon sistem orde pertama [9]

Respon transien terdiri dari:

a) Spesifikasi teoritis:

Konstanta waktu (τ), adalah waktu yang dibutuhkan respon mulai dari $t=0$ sampai dengan respon mencapai 63,2% dari respon *steady state*. Konstanta waktu menyatakan kecepatan respon sistem. Konstanta waktu yang lebih kecil akan mempercepat respon

b) Spesifikasi praktis:

1 Waktu tunak atau *settling time* (t_s), adalah ukuran waktu yang menyatakan bahwa respon sistem telah masuk pada daerah stabil. Jika dihubungkan dengan konstanta waktu τ , maka t_s dapat diformulasikan sebagai berikut:

$$t_s(\pm 3\%) \approx 5\tau$$

$$t_s(\pm 4\%) \approx 2\tau$$

$$t_s(\pm 5\%) \approx 0,5\tau$$

2 Waktu naik atau *rise time* (t_r), adalah ukuran yang menyatakan bahwa respon sistem telah naik dari 5% ke 95% atau 10% ke 90% dari nilai respon pada keadaan tunak (*steady state*). Jika dihubungkan dengan konstanta waktu τ , maka t_r dapat diformulasikan sebagai berikut:

$$t_r(5\% - 95\%) \approx \tau \ln 19$$

$$t_r(10\% - 90\%) \approx \tau \ln 9$$

- 3 Waktu tunda atau *delay time* (t_d), adalah waktu yang dibutuhkan respon mulai $t=0$ sampai respon mencapai 50% dari nilainya pada keadaan tunak (*steady state*). Waktu tunda menyatakan besarnya faktor keterlambatan respon akibat proses *sampling*. Jika dihubungkan dengan konstanta waktu τ , maka t_d dapat diformulasikan sebagai berikut:

$$t_d(\pm 3\%) \approx \tau \ln 12$$

- 4 Karakteristik respon keadaan tunak (*steady state*) sistem orde pertama diukur berdasarkan kesalahan pada keadaan tunak atau *error steady state* (e_{ss}).

$$e_{ss} = R_{ss} - C_{ss}$$

dengan C_{ss} dan R_{ss} masing-masing adalah keluaran dan masukan sistem pada keadaan tunak.

2.3 Proportional Integral Derivative (PID)

PID adalah salah satu pengendali otomatis yang terdiri dari perpaduan tiga aksi dasar kendali yaitu Proporsional, Integral, dan Derivatif. pada masing-masing memiliki aksi yang khas pada tiap-tiap kekurangan tiap pengendali P, I dan D dapat saling menutupi dengan menggabungkan ke tiganya dan merangkai secara paralel menghasilkan pengendali PID. Pada masing- masing aksi keseluruhan memiliki tujuan untuk mempercepat reaksi sebuah sistem yang mampu menghilangkan offset dan menghasilkan perubahan yang besar[8]. Oleh karena itu, pengendali PID merupakan pengendali berumpan balik dengan baik pada beberapa aplikasi proses petroleum, proses polimer, dan lain sebagainya.

Adapun bentuk umum dari aksi kendali PID sebagai berikut :

$$m(t) = k_p e(t) + K_p T_d \frac{de(t)}{dt} + \frac{K_p}{T_i} \int_0^1 e(t) \quad (2.17)$$

$$m(t) = k_p e(t) + K_D \frac{de(t)}{dt} + K_I \int_0^1 e(t) \quad (2.18)$$

Fungsi alihnya adalah

$$\frac{M(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \quad (2.19)$$

Dimana :

k_p = Proporsional Gain

k_i = Integral Gain

k_d = Derivatif Gain

Pengendali proporsional (K_p) akan memberikan efek mengurangi waktu naik, tetapi tidak menghapus *error steady state*. Pengendali integral (K_i) akan memberikan efek menghapus *error steady state* tetapi berakibat buruk pada saat menanggapi respon. Pengendali derivatif (K_d) berdampak pada stabilitas sistem, mengurangi kesalahan keadaan tunak dan menaikkan respon transien.

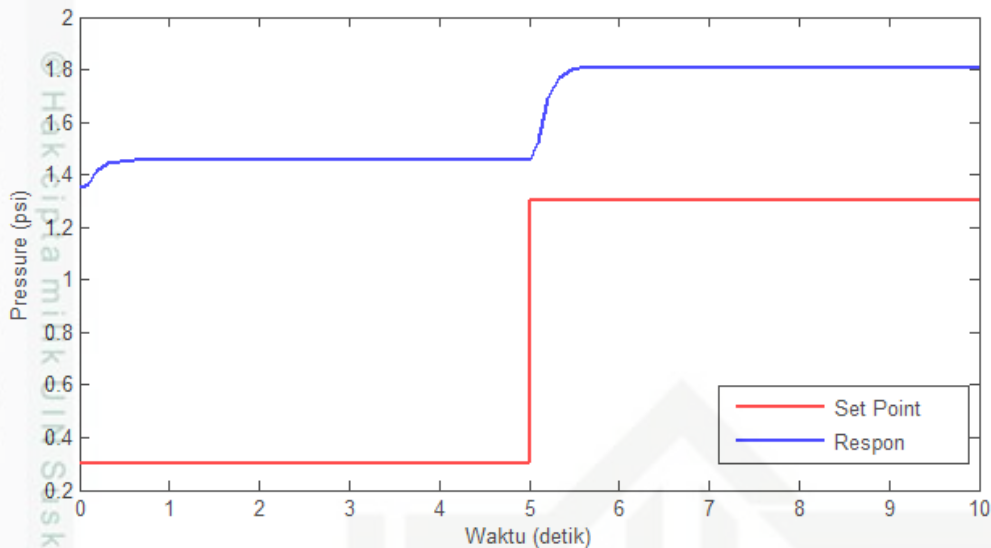
2.3.1 Tuning (Penalaan) Parameter PID Dengan Metode Ciancone

Istilah penalaan (*tuning*) untuk menggambarkan metode-metode yang digunakan dalam memilih pengontrolan agar mendapat unjuk kerja yang diinginkan. Pada penelitian ini, ini dilakukan penalaan pada pengendali P,I dan D dengan menggunakan metode *ciancone*.

Hubungan dari ketiga aksi pengendali tersebut ditunjukkan pada Tabel berikut :

Tabel 2.3 Hubungan antara kendali P, I, dan D [6]

Respon lup tertutup	Waktu naik	<i>Overshoot</i>	Waktu turun	Kesalahan keadaan tunak
K_p	Menurun	Meningkat	Perubahan Kecil	Menurun
K_i	Menurun	Meningkat	Meningkat	Hilang
K_d	Perubahan Kecil	Menurun	Menurun	Perubahan Kecil



Gambar 2.8 Grafik hasil pada gambar diagram *block* 2.5

Untuk mendesain pengendali Ciancone dengan parameter K_p , θ , dan τ , diperlukan *transfer function* dari sistem pada Diagram grafik tersebut. Berdasarkan data yang diperoleh dari tabel 2.2 sehingga di dapatkan persamaan transfer function sebagai berikut:

$$G(s) = \frac{K_p}{\tau s + 1} e^{-\theta s}$$

$$= \frac{0.351}{0.135s + 1} e^{-0.076s} \quad (2.20)$$

Kemudian, berdasarkan transfer function $G(s)$ tersebut, diperoleh nilai $K_p = 0.351$, $\theta = 0.076$, dan $\tau = 0.135$. Kemudian, untuk mencari pengendali Ciancone, digunakan bentuk umum dari Pengendali Ciancone, yaitu

$$MV = K_c \left[E(t) + \frac{1}{T_i} \int_0^\infty E(t) dt - T_d \frac{dCV}{dt} \right] + I \quad (2.21)$$

Untuk menentukan nilai K_c , T_i , dan T_d , digunakan grafik Ciancone Set Point dengan menggunakan MATLAB sehingga diperoleh nilai K_c , T_i , dan T_d masing-masing secara berturut-turut sebesar 2.76, 0.16, dan 0.012.

Untuk menentukan nilai K_p , K_i , dan K_d menggunakan persamaan sebagai berikut:

$$K_p = K_c = 2.76 \quad (2.22)$$

$$K_i = \frac{K_c}{T_i} = \frac{2.76}{0.16} = 17.25 \quad (2.23)$$

$$K_d = K_c T_d = 2.76 \cdot 0.012 = 0.033 \quad (2.24)$$

2.4 Linear Quadratic Regulator (LQR)

Pokok bahasan dalam kendali optimal adalah menentukan sinyal kendali yang akan di proses untuk memenuhi batasan fisik sesuai dengan kriteria performansi yang diinginkan.

Linear Quadratic Regulator (LQR) merupakan salah satu metode kendali optimal dengan kriteria kuadratik untuk menyelesaikan permasalahan regulator [9]. Disebut linier karena model dan bentuk kendalinya berupa sistem linear sedangkan disebut kuadratik karena memiliki *Cost Function* yaitu kuadrat dan karena referensi sistem bukan fungsi waktu maka disebut regulator [10]. Dari hasil linierisasi didapatkan suatu *plant* linear dalam bentuk :

$$\begin{aligned} \dot{x} &= A_x + B_u \\ y &= Cx \end{aligned} \quad (2.25)$$

Dimana,

A : matriks sistem

B : matriks *input*

C : matriks *output*

y : *State output*

x : *State* sistem

u : *State input*

Pada perancangan pengendali optimal LQR, terlebih dahulu menentukan matriks Q dan R yang selanjutnya digunakan untuk menentukan indeks performansi sistem, harga matriks Q dan R ditentukan sesuai dengan kriteria yang diinginkan dengan menggunakan indeks performansi [10].

$$J(t_0) = \frac{1}{2} S(T)x^2(T) + \frac{1}{2} \int_{t_0}^T (Qx^2 + Ru^2) \quad (2.26)$$

Dengan syarat :

$$S(T) \geq 0, \quad Q \geq 0, \quad R > 0$$

dimana

t_0 = waktu awal

T = waktu akhir

x = matriks *State* akhir

Q = matriks semi definit positif

R = matriks definit positif

S = matriks semi definit positif

Sehingga diperoleh persamaan Hamilton

$$H(x, u, \lambda, t) = L(x, u, t) + \lambda^T f(x, u, t)$$

$$\dot{H}(x, u, \lambda, t) = \frac{1}{2} (x^T Qx + u^T Ru) + \lambda^T (Ax + Bu) \quad (2.27)$$

Dari fungsi Hamilton tersebut dapat diperoleh syarat perlu dan syarat batas sebagai berikut:

1. Syarat perlu

a) Persamaan *State*

$$\dot{x} = \left(\frac{\partial x}{\partial x} \right)^T Qx + \left(\frac{\partial Qx}{\partial \lambda} \right)^T u = Ax + Bu \quad (2.28)$$

b) Persamaan *Costate*

$$-\dot{\lambda} = \frac{1}{2} \left[\left(\frac{\partial x}{\partial x} \right)^T Qx + \left(\frac{\partial Qx}{\partial x} \right)^T x + \left(\frac{\partial u}{\partial x} \right)^T Ru + \left(\frac{\partial Ru}{\partial x} \right)^T u \right] + \left(\frac{\partial \lambda}{\partial x} \right)^T (Ax + Bu) + \left(\frac{\partial (Ax + Bu)}{\partial x} \right)^T \lambda$$

$$-\dot{\lambda} = Qx + A^T \lambda \quad (2.29)$$

c) Kondisi stasioner

$$\frac{\partial H}{\partial u} = 0$$

$$\frac{1}{2} \left[\left(\frac{\partial x}{\partial x} \right)^T Qx + \left(\frac{\partial Qx}{\partial x} \right)^T x + \left(\frac{\partial u}{\partial x} \right)^T Ru + \left(\frac{\partial Ru}{\partial x} \right)^T u \right] + \left(\frac{\partial x}{\partial u} \right)^T (Ax + Bu) + \left(\frac{\partial (Ax + Bu)}{\partial u} \right)^T \lambda = 0$$

$$Ru + B^T \lambda = 0$$

$$u = -R^{-1} B^T \lambda \quad (2.30)$$

2. Syarat batas

a. Batas awal

$$t = 0 \quad x(0) = 0$$

b. Batas akhir

$$(\phi_x + \psi_x^T v - \lambda)^T dx|_T + (\phi_t + \psi_t^T v - H) dt|_{t=T} = 0 \quad (2.31)$$

Karena $\psi = 0$ dan

$$\phi(T) = \frac{1}{2} x^T(T) S(T) x(T)$$

$$\phi_x = \frac{\partial \phi}{\partial x} = S(T) x(T)$$

sehingga diperoleh persamaan

$$(S(T)x(T) + v.0 - \lambda)^T dt \Big|_{t=T} = 0$$

$$S(t)x(t) = \lambda(t) \tag{2.32}$$

Dari persamaan *Costate* dan (2.32), keduanya mengandung variabel λ . Bila keduanya disubstitusikan maka akan menghasilkan persamaan:

$$\begin{aligned}
 S(t)x(t) &= \lambda(t) \\
 \dot{\lambda}(t) &= \dot{S}(t)x(t) + S(t)\dot{x}(t) \\
 \dot{\lambda}(t) &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\dot{x}(t) &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\{Ax + Bu\} &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\{Ax + B(-R^{-1}B^T \lambda)\} &= -Qx - A^T \lambda \\
 -\dot{S}x &= (A^T S + SA - SBR^{-1}B^T S + Q)x \\
 -\dot{S} &= A^T S + SA - SBR^{-1}B^T S + Q \tag{2.33}
 \end{aligned}$$

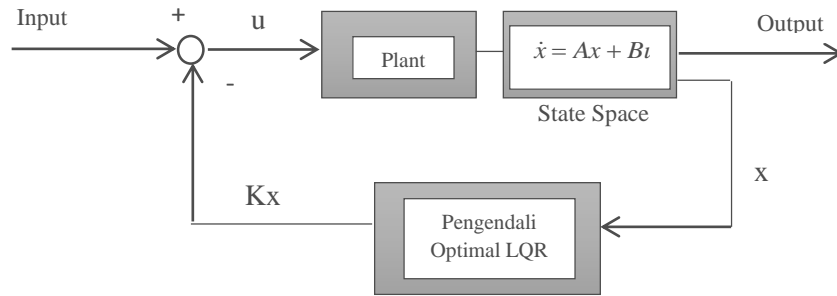
Persamaan (2.33) disebut persamaan differensial Riccati. Untuk $\dot{S} = 0$ persamaannya disebut *Algebraic Riccati Equation*(ARE) [10]. Dengan di dapatkannya matriks S dari persamaan ARE ini sehingga persamaan sinyal kendali pun dapat di hitung:

$$u = -R^{-1}B^T \lambda = -R^{-1}B^T Sx = -Kx \tag{2.34}$$

Maka konstanta umpan balik keadaan K dapat dicari sebelum menghasilkan sinyal kendali pada persamaan (2.35) [10].

$$K = R^{-1}B^T S \tag{2.35}$$

Dimana nilai S adalah unik, solusi semi definit positif untuk persamaan ARE harus memenuhi syarat persamaan (2.28). Selanjutnya dalam perancangan teknik kendali optimal LQR, Setelah nilai S diketahui maka, nilai S tersebut disubstitusikan kedalam persamaan (2.30) sehingga didapatkan nilai matriks optimal K, dengan K adalah *Gain State Feedback*, sehingga sistem kendali optimal LQR hasil desain dapat di perhatikan pada gambar 2.6 berikut:



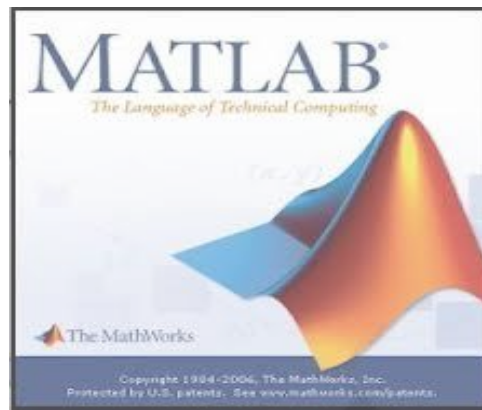
Gambar 2.9 Sistem Kendali dengan Skema Kendali Optimal LQR

Dari penjabaran di atas dapat disimpulkan bahwa langkah-langkah untuk merancang kendali LQR adalah sebagai berikut :

1. Jika persamaan matematis masih dalam bentuk *transfer function*, ubah bentuk *transfer function plant* nonlinier menjadi bentuk *state space* dan dapatkan matriks A, B, C dan D dari *plant* tersebut.
2. Tentukan matriks pembobot Q dan R.
3. Selesaikan persamaan aljabar riccati hingga menghasilkan suatu matriks S yang definit positif. Hitung optimal *gain feedback* K.

2.5 MATRIX LABORATORY (MATLAB)

Matlab adalah singkatan dari *Matrix Laboratory*. Pertama kali dibuat untuk mempermudah penggunaan dua koleksi subrutin pada pustaka *Fortran* yaitu: *Linpack* dan *Eispack*, dalam menangani komputasi matriks. Sejak itu, Matlab berkembang menjadi sebuah sistem yang interaktif sekaligus sebagai bahasa pemrograman untuk keperluan-keperluan ilmiah, komputasi teknis, dan visualisasi. Elemen data dasar Matlab adalah matriks. Perintah-perintah diekspresikan dalam bentuk yang sangat mirip dengan bentuk yang digunakan dalam matematika dan bidang teknik. Contoh persamaan $b = Ax$, dengan A, b, dan x matriks, ditulis: $b = A * x$. Untuk mendapat solusi x dari A dan b, tulis: $x = A \backslash b$. Tidak diperlukan penulisan program khusus untuk operasi-operasi matriks seperti perkalian matriks atau invers matriks. Oleh karena itu bahasa Matlab menyelesaikan masalah tersebut memerlukan waktu lebih cepat dibanding waktu yang dibutuhkan bahasa pemrograman tingkat tinggi lain.



Gambar 2.10 Tampilan Matlab

Pada pertengahan tahun 1970, Cleve Moler dan beberapa rekan tergabung dalam suatu team pengembangan *software* yang dibiayai oleh *The National Science Foundation* untuk tujuan membuat subrutin-subrutin dalam pustaka *Fortran* yang dinamai *Linpack* dan *Eispack*. *Linpack* berisi koleksi subrutin untuk penyelesaian persamaan linear, sementara *Eispack* adalah koleksi subrutin untuk penyelesaian masalah nilai pribadi (*eigenvalue*). Baik *Linpack* maupun *Eispack* pada prinsipnya merupakan program untuk komputasi matriks.

Dipenghujung tahun 1970, Cleve ingin dapat mengajarkan kepada mahasiswa materi aljabar linear di Universitas New Mexico menggunakan *Linpack* dan *Eispack* tanpa harus menulis rutin-rutin program dalam bahasa *Fortran*. Berdasarkan keinginan tersebut, Cleve mulai menulis program untuk memberikan kemudahan akses interaktif pada *Linpack* dan *Eispack*. Cleve menamakan programnya dengan Matlab yang merupakan singkatan dari MATrix LABoratory. Beberapa tahun kemudian, ketika Cleve berkunjung ke universitas lain untuk berbicara, atau sebagai *Visiting Professor*, Cleve meninggalkan duplikasi Matlabnya pada komputer di universitas tersebut. Hanya dalam satu atau dua tahun, Matlab versi pertama ini telah menjadi buah bibir pembicaraan orang, terutama yang berada dalam komunitas matematika terapan.

Dari hasil kunjungan Cleve di Universitas Stanford, sekitar awal tahun 1983, John Little, seorang *engineer*, menampilkan Matlab dengan memperkenalkan penerapan Matlab yang potensial dalam bidang-bidang keteknikan. Karena itu, dalam tahun 1983, Little, Moler, dan Steve Bangert membentuk team untuk mengembangkan Matlab generasi kedua. Matlab versi ini dibuat menggunakan bahasa C dan terintegrasi dengan grafik. *The MathWorks, Inc.* didirikan tahun 1984 untuk memasarkan dan melanjutkan pengembangan Matlab.

Penggunaan Matlab:

1. Adanya *toolbox* yang disebut MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer



2. Komputasi matematik dan komputasi teknik
3. Analisis data.
4. Simulasi dan pemodelan
5. Grafik-grafik perhitungan.
6. Manipulasi matriks
7. Pem-plot-an fungsi dan data
8. Implementasi algoritma
9. Pembuatan antarmuka pengguna / *interface* (visualisasi)
10. Peng-antarmuka-an dengan program dalam bahasa lainnya

Fungsi-fungsi yang digunakan Matlab:

1. Fungsi matematika
2. Fungsi fisika
3. Fungsi statistik
4. Fungsi visualisasi.

Matlab merupakan *software* yang digunakan untuk:

1. Para pendidikan (untuk melakukan riset dalam bentuk komputasi, pemodelan, simulasi dan demonstrasi / visualisasi)
2. Para ilmuwan
3. Matematikawan
4. Pelajar dan mahasiswa terutama para pendidik matematika
5. Praktisi
6. Insinyur.
7. dan untuk siapa saja tanpa ada batasannya

Sebagai sebuah system, Matlab tersusun dari 5 bagian utama:

1. *Development Environment*.

Merupakan sekumpulan perangkat dan fasilitas yang membantu anda untuk menggunakan fungsi-fungsi dan file-file Matlab. Beberapa perangkat ini merupakan sebuah *graphical user interfaces* (GUI). Termasuk didalamnya adalah Matlab desktop dan *Command Window, command history*, sebuah *editor* dan *debugger*, dan *browsers* untuk melihat *help, workspace, files*, dan *search path*.

2. Matlab *Mathematical Function Library*

Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: *sum*, *sin*, *cos*, dan *complex arithmetic*, sampai dengan fungsi-fungsi yang lebih kompleks seperti *matrix inverse*, *matrix eigenvalues*, *Bessel functions*, dan *fast Fourier transforms*.

3. Matlab *Language*

Merupakan suatu *high-level matrix/array language* dengan *control flow statements*, *functions*, *data structures*, *input/output*, dan fitur-fitur *object-oriented programming*. Inimemungkinkan bagi kita untuk melakukan kedua hal baik “pemrograman dalam lingkupsederhana ” untuk mendapatkan hasil yang cepat, dan “pemrograman dalam lingkup yang lebihbesar” untuk memperoleh hasil-hasil dan aplikasi yang kompleks.

4. *Graphics*

Matlab memiliki fasilitas untuk menampilkan vector dan matrices sebagai suatugrafik. Didalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, *image processing*, *animation*, dan *presentation graphics*.Ini juga melibatkan fungsi level rendah yang memungkinkan bagi anda untuk membiasakan diriuntuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan *graphicaluser interfaces* pada aplikasi Matlab anda.

5. Matlab *Application Program Interface (API)*

Merupakan suatu *library* yang memungkinkan program yang telah anda tulis dalam bahasa C dan Fortran mampu berinteraksi dengan Matlab. Ini melibatkan fasilitas untuk pemanggilan *routines* dari Matlab (*dynamic linking*), pemanggilan Matlab sebagai sebuah *computational engine*, dan untuk membaca dan menuliskan *MAT-files*. Kehadiran Matlab memberikan jawaban sekaligus tantangan. Disini kita bisa memilih apa aja yg ingin kita pelajari, mempelajari metoda visualisasi saja, pemrograman saja, atau keduanya di sini intinya Matlab memang dihadirkan bagi mereka yang tidak ingin disibukkan dengan rumitnya sintak dan alur logika pemrograman sementara pada saat yang sama membutuhkan hasil komputasi dan visualisasi yang maksimal untuk mendukung pekerjaannya. Selain itu Matlab juga memberikan keuntungan bagi *programmer* dan *developer* program yaitu menjadi program pembanding yang sangat handal, hal tersebut karena kekayaannya akan fungsi matematika, fisika, statistik dan visualisasi.