

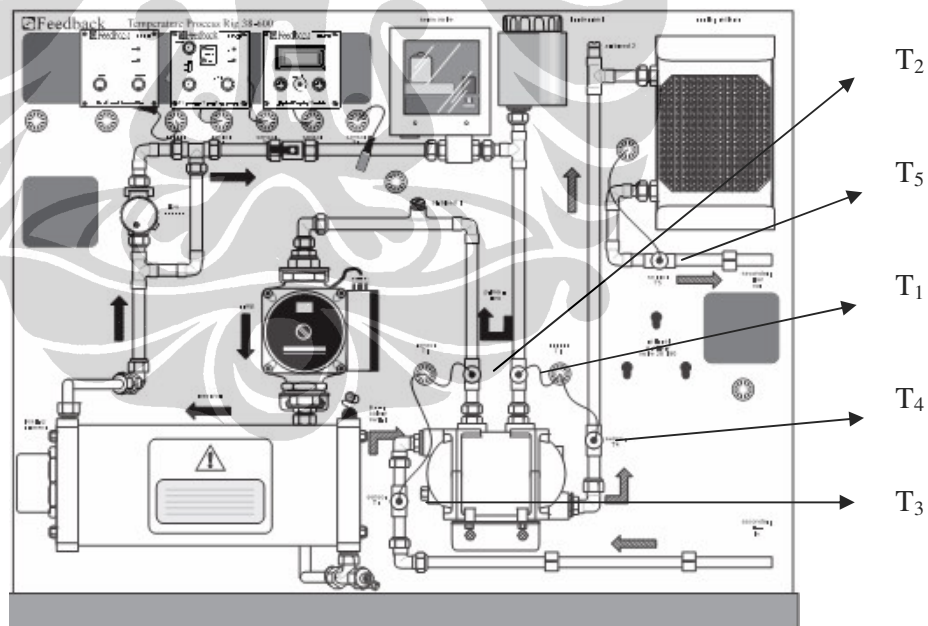
BAB 3

PERANCANGAN PENGENDALI MPC UNTUK SISTEM *HEAT EXCHANGER*

3.1 Pemodelan *Heat Exchanger*

Sistem *Level/Flow and Temperature Process Rig 38-003* merupakan gabungan dari dua buah sistem yang dapat berdiri sendiri yaitu *Basic Process Rig 38-100* dan *Temperature Process Rig 38-600*. *Basic Process Rig 38-100* akan dipakai sebagai penyalur fluida primer ke *Temperature Process Rig 38-600*. *Temperature Process Rig 38-600* tersusun dari beberapa komponen dasar, antara lain : *Closed Primary Hot Water Circuit, Electrical Heater, Heat Exchanger, 2 Motorized Flow Valves, Pulse Flow Sensor, 5 Thermistor Temperature Sensor, Fan-Assisted Cooling Radiator, Signal Conditioning Units*.

Berikut ini diberikan sketsa dari *Temperature Process Rig 38-600* :



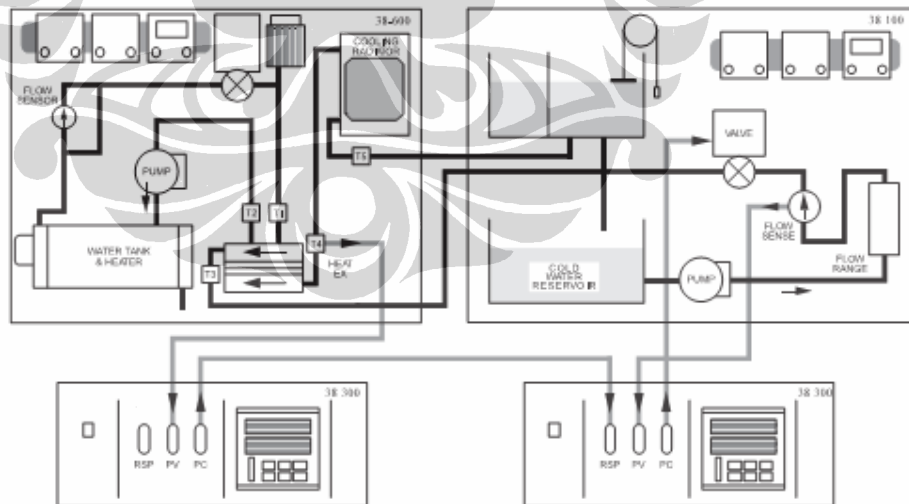
Gambar 3.1 Sketsa *Temperature Process Rig 38-600*

Sistem diatas termasuk kedalam *Heat Exchanger* jenis *Shell and Tube* dengan arah aliran *Counter-current*. Hubungan-hubungan dari kelima thermistor diatas adalah [2]:

1. T_2 seharusnya lebih rendah dari T_1 , panas dihantarkan dari bagian primer ke bagian sekunder.
2. T_3 seharusnya lebih rendah dari T_4 , air dingin dipanaskan didalam *Heat Exchanger*.
3. T_5 seharusnya lebih rendah dari T_4 , air yang tadinya dipanaskan harus didinginkan sebelum kembali ke *sump*.

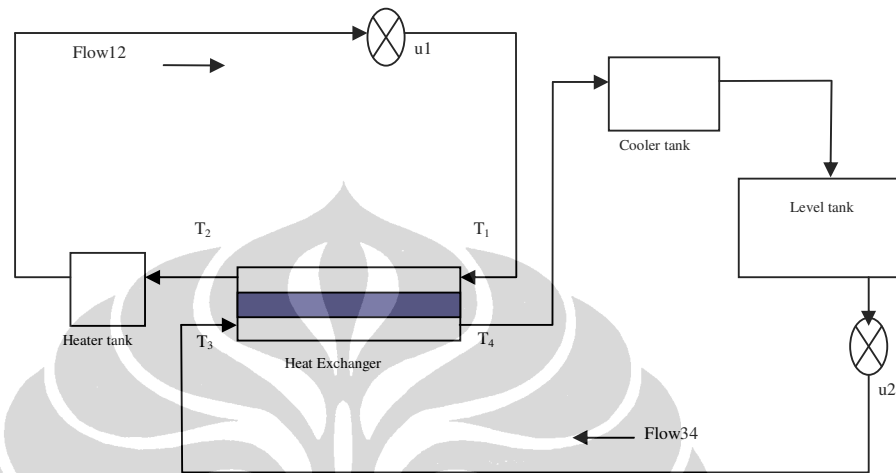
Pada Tesis ini, sumber fluida utama untuk *Temperature Process Rig 38-600* akan disediakan secara langsung oleh *Basic Process Rig 38-100*. Oleh karena itu, *outlet Basic Process Rig 38-100* akan dihubungkan ke *inlet Temperature Process Rig 38-600* sedangkan *outlet Temperature Process Rig 38-600* akan dihubungkan ke *sump Basic Process Rig 38-100*.

Berikut ini adalah sketsa dari gabungan *Basic Process Rig 38-100* dan *Temperature Process Rig 38-600* yang selanjutnya akan dinamakan dengan *Level/Flow and Temperature Process Rig 38-003* :



Gambar 3.2. Sketsa *Level/Flow and Temperature Process Rig 38-003*

Untuk selanjutnya, penelitian tesis ini dibatasi hanya disekitaran sistem *Heat Exchanger*-nya saja. Dengan mengabaikan pemodelan pada sistem pemanas dan sistem pendingin, maka sistem pemanas dan pendingin dianggap telah memiliki nilai konstan. Sketsanya dapat dilihat pada gambar 3.3.



Gambar 3.3 Sketsa Sistem *Heat Exchanger*

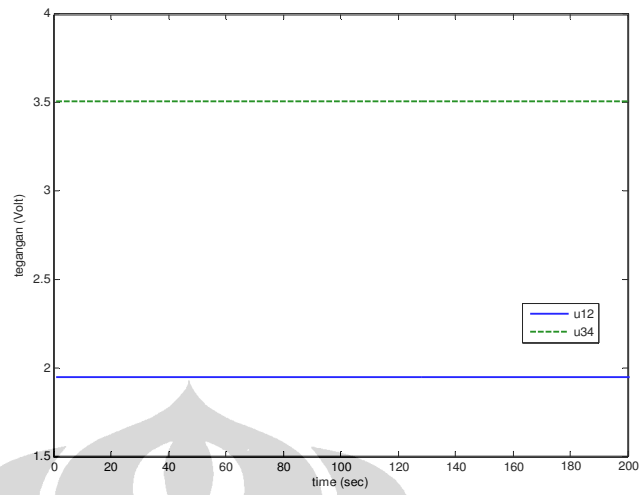
3.2 Model Dinamik Sistem

Dalam kondisi setimbang dari persamaan (2.1) dan (2.2), perubahan suhu pada *Heat Exchanger* dapat dirumuskan dalam persamaan diferensial non linear [5]:

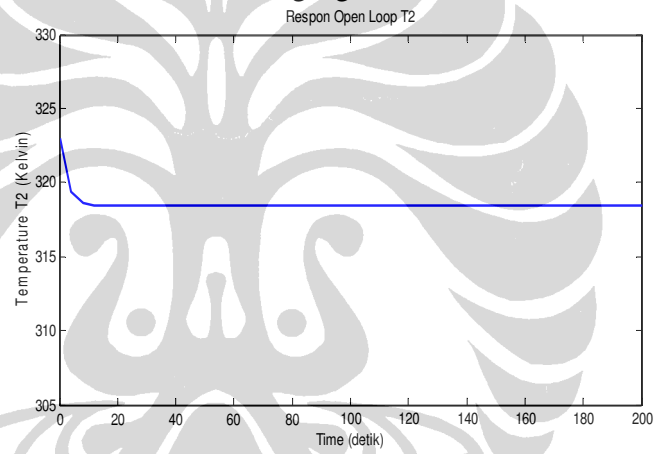
$$\rho_H C_H V_H \frac{dT_2}{dt} = \rho_H C_H f_H (T_1 - T_2) - UA \left(\frac{T_1 + T_2}{2} - \frac{T_3 + T_4}{2} \right) \quad (3.1)$$

$$\rho_C C_C V_C \frac{dT_4}{dt} = \rho_C C_C f_C (T_3 - T_4) + UA \left(\frac{T_1 + T_2}{2} - \frac{T_3 + T_4}{2} \right) \quad (3.2)$$

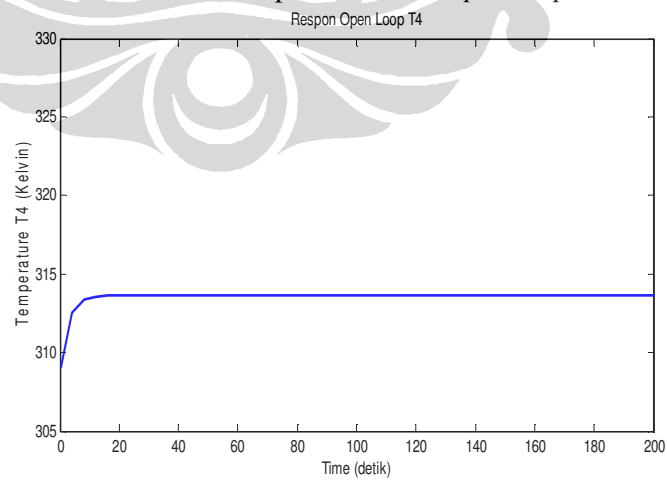
Respons uji *open loop* adalah sebagai berikut:



Gambar 3.4 Tegangan masukan u_1 dan u_2



Gambar 3.5 Respons keluaran *open loop* T_2



Gambar 3.6 Respon keluaran *open loop* T_4

Untuk mendapatkan model linier yang paling tepat mewakili sistem *Heat Exchanger*, maka diperlukan linearisasi sistem disekitaran *operating point*-nya

Untuk persamaan (3.1):

$$\rho_H C_H V_H \frac{dT_2}{dt} = \rho_H C_H f_H (T_1 - T_2) - UA \left(\frac{(T_1 + T_2)}{2} - \frac{(T_3 + T_4)}{2} \right)$$

Linearisasi nya:

$$\rho_H C_H V_H \frac{dT_2}{dt} = \rho_H C_H f_H (T_1 - T_2) - UA \left(\frac{(T_1 + T_2)}{2} - \frac{(T_3 + T_4)}{2} \right)$$

$$\frac{dT_2}{dt} = \frac{\rho_H C_H f_H T_1}{\rho_H C_H V_H} - \frac{\rho_H C_H f_H T_2}{\rho_H C_H V_H} - \left(UA \frac{(T_1 + T_2)}{2\rho_H C_H V_H} - UA \frac{(T_3 + T_4)}{2\rho_H C_H V_H} \right)$$

$$\frac{dT_2}{dt} = \frac{f_H T_1}{V_H} - \frac{f_H T_2}{V_H} - \frac{UAT_1}{2\rho_H C_H V_H} - \frac{UAT_2}{2\rho_H C_H V_H} + \frac{UAT_3}{2\rho_H C_H V_H} + \frac{UAT_4}{2\rho_H C_H V_H}$$

$$F_1 = f_1(u_1, u_2, T_2, T_4)$$

$$\left. \frac{\partial f_1}{\partial u_1} \right|_{u_1=u_{1,ss}} = \frac{Kvq_{12} \cdot T_1}{V_H} - \frac{Kvq_{12} \cdot T_2}{V_H}$$

$$\left. \frac{\partial f_1}{\partial u_2} \right|_{u_2=u_{2,ss}} = 0$$

$$\left. \frac{\partial f_1}{\partial T_2} \right|_{T_2=T_{2,ss}} = -\frac{Kvq_{12} \cdot u_1}{V_H} - \frac{UAT_2}{2\rho_H C_H V_H}$$

$$\left. \frac{\partial f_1}{\partial T_4} \right|_{T_4=T_{4,ss}} = \frac{UAT_4}{2\rho_H C_H V_H}$$

$$\delta \dot{T}_2 = \frac{\partial f_1}{\partial u_1} (u_1 - u_{1,ss}) + \frac{\partial f_1}{\partial u_2} (u_2 - u_{2,ss}) + \frac{\partial f_1}{\partial T_2} (T_2 - T_{2,ss}) + \frac{\partial f_1}{\partial T_4} (T_4 - T_{4,ss}) \quad (3.3)$$

Untuk persamaan (3.2)

$$\rho_C C_C V_C \frac{dT_4}{dt} = \rho_C C_C f_C (T_3 - T_4) + UA \left(\frac{(T_1 + T_2)}{2} - \frac{(T_3 + T_4)}{2} \right)$$

$$\frac{dT_4}{dt} = \frac{\rho_c C_c f_c T_3}{\rho_c C_c V_c} - \frac{\rho_c C_c f_c T_4}{\rho_c C_c V_c} - \left(UA \frac{(T_1 + T_2)}{2\rho_c C_c V_c} - UA \frac{(T_3 + T_4)}{2\rho_c C_c V_c} \right)$$

$$\frac{dT_4}{dt} = \frac{f_c T_3}{V_c} - \frac{f_c T_4}{V_c} + \frac{UAT_1}{2\rho_c C_c V_c} + \frac{UAT_2}{2\rho_c C_c V_c} - \frac{UAT_3}{2\rho_c C_c V_c} - \frac{UAT_4}{2\rho_c C_c V_c}$$

$$F_2 = f_2(u_1, u_2, T_2, T_4)$$

$$\left. \frac{\delta f_2}{\delta u_1} \right|_{u_1=u_{1,ss}} = 0$$

$$\left. \frac{\delta f_2}{\delta u_2} \right|_{u_2=u_{2,ss}} = \frac{Kvq_{34} \cdot T_3}{V_c} - \frac{Kvq_{34} \cdot T_4}{V_c}$$

$$\left. \frac{\delta f_2}{\delta T_2} \right|_{T_2=T_{2,ss}} = \frac{UAT_2}{2\rho_c C_c V_c}$$

$$\left. \frac{\delta f_2}{\delta T_4} \right|_{T_4=T_{4,ss}} = -\frac{Kvq_{34} \cdot u_2}{V_c} - \frac{UAT_4}{2\rho_c C_c V_c}$$

$$\delta \dot{T}_4 = \frac{\partial f_2}{\partial u_1} (u_1 - u_{1,ss}) + \frac{\partial f_2}{\partial u_2} (u_2 - u_{2,ss}) + \frac{\partial f_2}{\partial T_2} (T_2 - T_{2,ss}) + \frac{\partial f_2}{\partial T_4} (T_4 - T_{4,ss}) \quad (3.4)$$

Dari persamaan (3.3) dan (3.4)

$$\delta \dot{T}_2 = \frac{\partial f_1}{\partial u_1} (u_1 - u_{1,ss}) + \frac{\partial f_1}{\partial u_2} (u_2 - u_{2,ss}) + \frac{\partial f_1}{\partial T_2} (T_2 - T_{2,ss}) + \frac{\partial f_1}{\partial T_4} (T_4 - T_{4,ss})$$

$$\delta \dot{T}_4 = \frac{\partial f_2}{\partial u_1} (u_1 - u_{1,ss}) + \frac{\partial f_2}{\partial u_2} (u_2 - u_{2,ss}) + \frac{\partial f_2}{\partial T_2} (T_2 - T_{2,ss}) + \frac{\partial f_2}{\partial T_4} (T_4 - T_{4,ss})$$

Diperoleh bentuk persamaan *state space*:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{Kvq_{42} \cdot u_{1,ss}}{V_H} - \frac{UAT_{2,ss}}{2\rho_H C_H V_H} & \frac{UAT_{4,ss}}{2\rho_H C_H V_H} \\ \frac{UAT_{2,ss}}{2\rho_c C_c V_c} & -\frac{Kvq_{34} \cdot u_{2,ss}}{V_c} - \frac{UAT_{4,ss}}{2\rho_c C_c V_c} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{kvq_{42} T_{2,ss}}{V_H} - \frac{kvq_{42} T_{4,ss}}{V_H} & 0 \\ 0 & \frac{kvq_{34} T_{2,ss}}{V_c} - \frac{kvq_{34} T_{4,ss}}{V_c} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.5)$$

dengan;

x_1 = Temperatur T_2 (*Output hot water*)

x_2 = Temperatur T_4 (*Output cold water*)

u_1 = Tegangan Valve untuk *flow* T_1 (*Input hot water*)

u_2 = Tegangan Valve untuk *flow* T_3 (*Input cold water*)

y = *Output* sistem (T_2 dan T_4)

3.3 Perhitungan Nilai Parameter Model Heat Exchanger

Parameter-parameter yang digunakan pada pemodelan sistem *Heat Exchanger* sesuai dengan parameter pada Tabel 3.1 dan Tabel 3.2

Tabel 3.1 Data Spesifik *Heat Exchanger* [1]

No	Nomenclature	Konstanta	Nilai	Satuan
1	Total panjang pipa	L	0.3	Meter
2	Jumlah pipa parallel	N	12	Batang
3	Diameter luar pipa	Do	0.0095	Meter
4	Diameter dalam pipa	Di	0.0077	Meter
5	Bentuk lintasan air dingin	“S”	3	Segmen parallel
6	Panjang body HE	L/3	0.1	Meter
7	Diameterbody HE	Dhe	0.078	Meter
8	Tebal rata-rata tutup sampung	T	0.0075	Meter
9	Luas penampang alir air dingin	$A_c = \phi/4 * D_i * D_i * n$	0.0005588	m ²
10	Volume air dingin	$V_c = A_c * L + \phi/4 * D_{he} * D_{he} * t * 2$	0.0002393	m ²
11	Volume air panas	$V_h = \phi/4 * (D_{he} * D_{he} - D_o * D_o * n * 3) * L$	0.0002227	m ²
12	Luas dindng luar pipa	$A = \phi * D_o * L * n$	0.1074425	m ²
13	Rapat massa air dingin / panas	Rhoc dan Rhoh	1000	Kg/m ³
14	Kalor spesifik air dingin / panas	Cc dan Ch	1	Kcal/Kg.K
15	Overall heat transfer	U	0.267	Kcal/s.m ² K
16	Flow air dingin maksimal	F_{34init}	5	l/menit
17	Flow air panas maksimal	F_{12init}	9	l/menit
18	Konstanta pengubah tegangan ke debit air dingin	Kv_{q34}	1.666E-5	m ³ /Vs
19	Konstanta pengubah tegangan ke debit air panas	Kv_{q34}	3E-5	M ³ /Vs
20	Suhu air panas maksimal	Theat	343	Kelvin
21	Suhu air dingin maksimal	Tcooler	298	Kelvin

Tabel 3.2 Data Keadaan Tunak Titik Kerja [1]:

No	Data keadaan tunak titik kerja:	Konstanta	Nilai	Satuan
1	Debit air dingin	F_{12ss}	5.833E-05	M^3/s
2	Debit air panas	F_{34ss}	5.833E-05	M^3/s
3	Suhu air panas masuk	T_{1ss}	323	Kelvin
4	Suhu air panas keluar	T_{2ss}	318.3	Kelvin
5	Suhu air dingin masuk	T_{3ss}	309	Kelvin
6	Suhu air dingin keluar	T_{4ss}	313.6K	Kelvin

Dengan memasukkan nilai parameter dari tabel (3.1) dan (3.2) ke persamaan (3.5), maka diperoleh model *state space* hasil linearisasi disekitar *operating point* adalah sebagai berikut:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -0.3263 & 0.06441 \\ 0.05994 & -0.3028 \end{bmatrix}}_A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0.6215 & 0 \\ 0 & -0.3211 \end{bmatrix}}_B \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_C \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.6)$$

dengan;

x_1 = Temperatur T_2 (Output hot water)

x_2 = Temperatur T_4 (Output cold water)

u_1 = Tegangan Valve untuk flow T_1 (Input hot water)

u_2 = Tegangan Valve untuk flow T_3 (Input cold water)

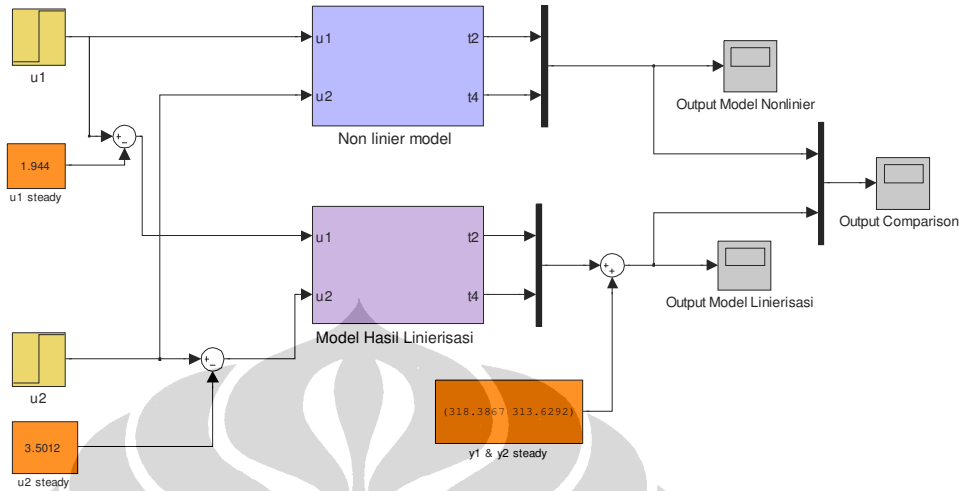
y = Output sistem (T_2 dan T_4)

Selain melalui linierisasi matematis, model linier yang sama dengan persamaan (3.6) juga diperoleh jika dilakukan linierisasi sistem dengan menggunakan fungsi analisa linier pada Simulink.

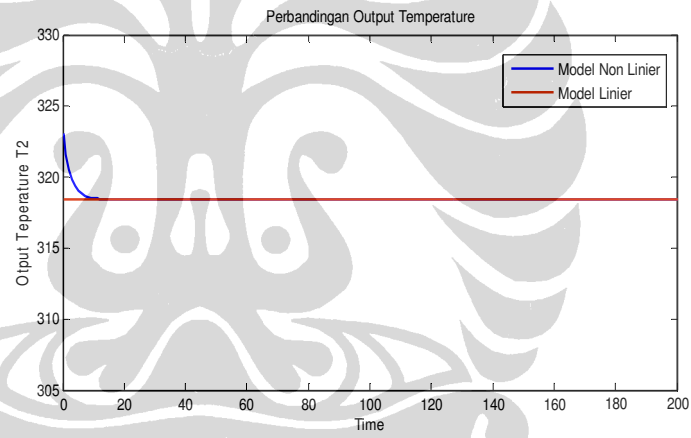
3.4. Validasi Model

Validasi model dilakukan untuk mengetahui kemiripan karakteristik sistem yang sebenarnya (model *non* linier) dengan model hasil linierisasi. Validasi dilakukan pada titik operasi $u_1 = 1.944$ V, $u_2 = 3.5012$ V dan $y_1 = 318.3867$, $y_2 =$

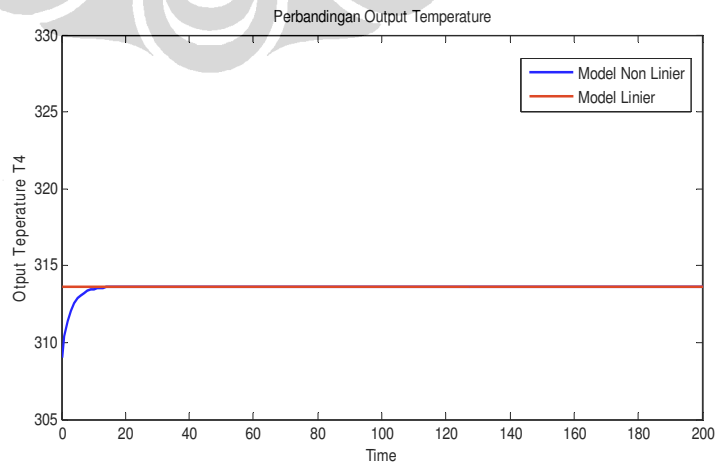
313.6293. Model linier yang digunakan adalah hasil linierisasi pada persamaan (3.6). Diagram blok validasi dengan simulink dapat dilihat pada gambar 3.7.



Gambar 3.7. Simulink validasi model linier pada *operating point*



Gambar 3.8 Output T_2



Gambar 3.9. Output T_4

Dari simulasi yang telah dilakukan, terlihat bahwa output Model Linearisasi memiliki karakteristik yang sama dengan model non linier. Hal ini menunjukkan bahwa model linier ini dapat digunakan dalam simulasi sistem *Heat Exchanger* dan sebagai model dasar dalam perancangan algoritma pengendali *Model Predictive Control*.

3.5. Perhitungan *Sampling Time*

Agar dapat mempresentasikan bagian transient sistem dengan baik yang diwakili oleh 4 sampai 20 data pencuplikan dalam rentang waktu *settling time* [4], maka nilai interval waktu pencuplikan dinyatakan dalam perumusan:

$$\frac{1}{20}T_{95} \leq h \leq \frac{1}{5}T_{95} \quad (3.7)$$

Dengan T_{95} adalah *settling time* dengan kriteria 5%. Respons sistem loop terbuka dengan masukan konstan sebesar $U_{1ss}=1.944$ volt dan $U_{2ss}=3.5012$ volt.

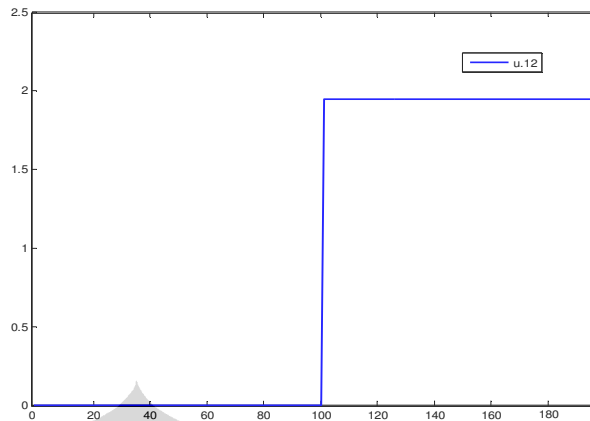
1. Untuk Kriteria *Settling Time* 5% :

- Pada T_2 , Nilai *Operating Point* T_2 adalah 318.3 K , maka batas kriteria 5 % adalah 302.39 K sampai dengan 334.21 K.
- Pada T_4 , Nilai *Operating Point* T_4 adalah 313.6 K, maka batas kriteria 5 % adalah 297.92 K sampai dengan 329.28 K.

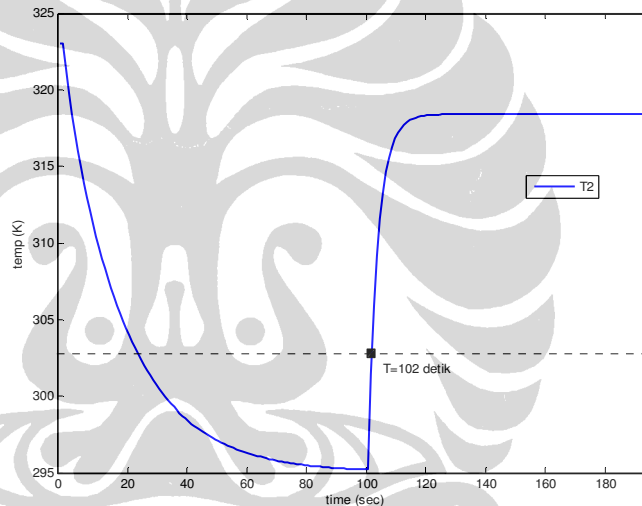
2. Simulasi *Open Loop* Respons terhadap model nonlinier

a. Kondisi 1

Input U_1 *Step* pada detik ke – 100 dari 0 ke nilai *operating point*-nya (1.944) dan input U_2 steady di nilai *operating point*-nya (3.5012).



Gambar 3.10. Input Tegangan *Step* u_{1ss}

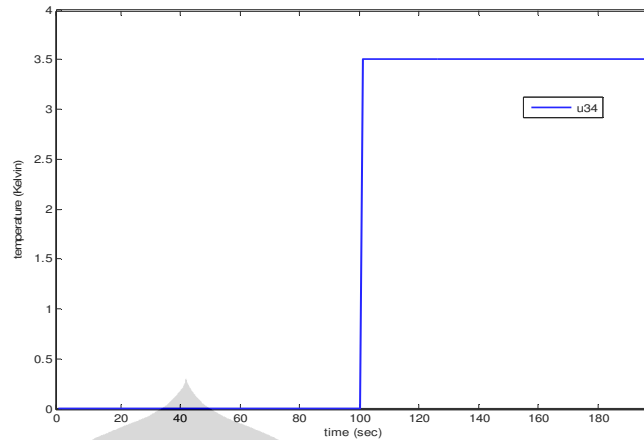


Gambar 3.11 Respon keluaran T_2 untuk input *step* u_{1ss}

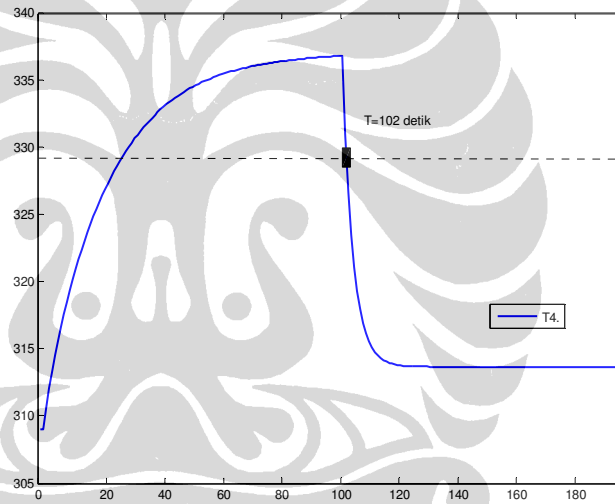
Untuk respon kondisi 1, diperoleh bahwa output T_2 mencapai nilai *settle* minimum 302.39 disekitar detik ke 102. Karena input u_1 *step* pada detik ke 100, maka nilai *Settling Time* = $102 - 100 = 2$ detik.

b. Kondisi 2

Input U_2 *Step* pada detik ke -100 dari 0 ke nilai *operating point*-nya (3.5012) dan input U_1 *steady* di nilai *operating point*-nya (1.944).



Gambar 3.12 Input Tegangan *Step* u_{2ss}



Gambar 3.13 Respon keluaran T_4 untuk input *step* u_{2ss}

Untuk respon kondisi 2, diperoleh bahwa output T_4 mencapai nilai *settle* minimum 329.28 disekitar detik ke 102. Karena input u_{12} *step* pada detik ke 100, maka nilai *Settling Time* = 102 – 100 = 2 detik.

3. Dari langkah atas, diperoleh nilai *Settling Time* (T_s) adalah 2 detik yang digunakan sebagai acuan untuk menentukan besar *Sampling Time* (h)

sistem berdasarkan hubungan persamaan (3.7): $\frac{1}{20} T_s \leq h \leq \frac{1}{5} T_s$

Dari hubungan tersebut, didapatkan rentang *sampling time* yang tepat adalah $0.1 \leq h \leq 0.4$. Untuk simulasi selanjutnya, akan digunakan *sampling time* (h) sebesar 0.25 detik.

Dengan menggunakan *sampling time* 0.25 detik, model diskrit linier dari sistem heat exchanger pada pers. 3.6 adalah:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0.9218 & 0.01489 \\ 0.01385 & 0.9272 \end{bmatrix} x(k) + \begin{bmatrix} 0.1492 & -0.0006134 \\ 0.001105 & -0.07731 \end{bmatrix} (u) \\ y(k) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(k) \end{aligned} \quad (3.8)$$

3.6. Nilai Eigen Sistem

Untuk pengecekan kestabilan sistem, maka diperlukan *eigen value* sistem. *Eigen value* dapat dicari dengan rumus:

$$[\lambda I - A] = 0 \quad (3.9)$$

$$[\lambda I - A] = \begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_3 & \lambda_4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.9218 & 0.01489 \\ 0.01385 & 0.9272 \end{bmatrix} = 0$$

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0.9099 \\ 0.9391 \end{bmatrix}$$

Semua nilai *Eigen* model sistem berada pada skala $0 \leq P \leq 1$ maka sistem adalah stabil.

3.7. Uji Observability dan Controllability Sistem

Pengujian *Observability* dan *Controllability* terhadap suatu model merupakan langkah penting yang harus dilakukan terlebih dahulu sebelum menentukan layak tidaknya sebuah model untuk dijadikan sebagai dasar perancangan sebuah pengendali. Uji *observability* sistem dimaksudkan untuk mengetahui apakah sistem tersebut benar-benar dapat diobservasi dan untuk mengetahui apakah *state-state* yang diamati tersebut dapat mewakili keadaan sistem yang sebenarnya. Sedangkan uji *controllability* akan menentukan apakah sistem yang diwakili oleh model tersebut dapat dikendalikan oleh sebuah pengendali.

Asumsikan bahwa model ruang keadaan sistem *Heat Exchanger* yang ditunjukkan pada persamaan (3.6) dapat diwakili oleh persamaan berikut :

$$x((k+1)T) = Ax(kT) + Bu(kT) \quad (3.10)$$

$$y(kT) = Cx(kT) \quad (3.11)$$

Untuk melakukan pengetesan *observability* dari suatu sistem, langkah yang harus dilakukan adalah membentuk matriks *observability* seperti yang ditunjukkan oleh persamaan berikut :

$$Q_o = \left[C^T : A^T C^T : \dots : (A^T)^{n-1} C^T \right] \quad (3.12)$$

dimana :

- n adalah jumlah *state* yang dimiliki oleh sebuah sistem.
- Sistem *observable* jika matriks *observability* memiliki *rank* sebanyak n (jumlah *state*).

Dengan menggunakan matriks C dan A persamaan (3.8) ke dalam persamaan (3.12), didapatkan matriks *observability* berikut ini;

$$Q_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0.9218 & 0.0149 \\ 0.0138 & 0.9272 \end{bmatrix}$$

Rank dari matriks Q_o adalah 2, dengan Rank matriks A dari model. Hal ini menunjukkan bahwa sistem bersifat *Observable* sempurna.

Untuk melakukan pengetesan *controllability* dari suatu sistem, langkah yang harus dilakukan adalah membentuk matriks *controllability* seperti yang ditunjukkan oleh persamaan berikut :

$$Q_c = [B : AB : \dots : A^{n-1} B] \quad (3.13)$$

dimana :

- n adalah jumlah *state* yang dimiliki oleh sebuah sistem.
- Sistem *controllable* jika matriks *controllability* memiliki *rank* sebanyak n (jumlah *state*).

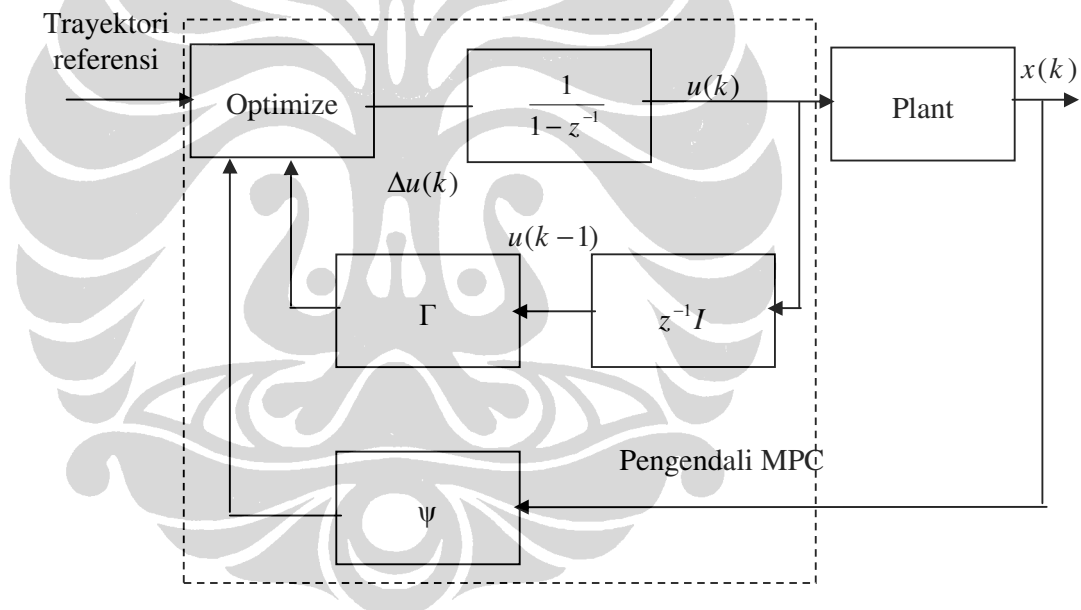
Dengan menggunakan matriks B dan A persamaan (3.8.) ke dalam persamaan (3.12), didapatkan matriks *controllability* berikut ini;

$$Q_c = \begin{bmatrix} 0.1492 & -0.0006 & 0.1375 & -0.0017 \\ 0.011 & -0.0773 & 0.0031 & -0.0717 \end{bmatrix}$$

Rank dari matriks Q_c adalah 2, dengan Rank matriks A dari model. Hal ini menunjukkan bahwa sistem bersifat *controllable* sempurna.

3.8. Algoritma Model Predictive Control dengan Constraints

Struktur pengendali MPC dengan *constraints* untuk model ruang keadaan terdapat pada gambar 3.14. Dari blok diagram tersebut, terlihat bahwa prediksi perubahan sinyal masukan sekarang ($\Delta u(k)$) membutuhkan data dari variabel keadaan sekarang $\underline{x}(k)$ dan masukan satu langkah sebelumnya $u(k-1)$.



Gambar 3.14. Blok Diagram Pengendali MPC dengan *Constraints*.

Algoritma perhitungan perubahan sinyal kendali pada MPC dengan *constraints* adalah sebagai berikut :

1. Parameter pengendali yang terlebih dahulu harus ditentukan antara lain horizon prediksi (H_p), horizon kendali (H_u), matriks faktor bobot kesalahan (Q), dan matriks faktor bobot perubahan sinyal kendali (R).

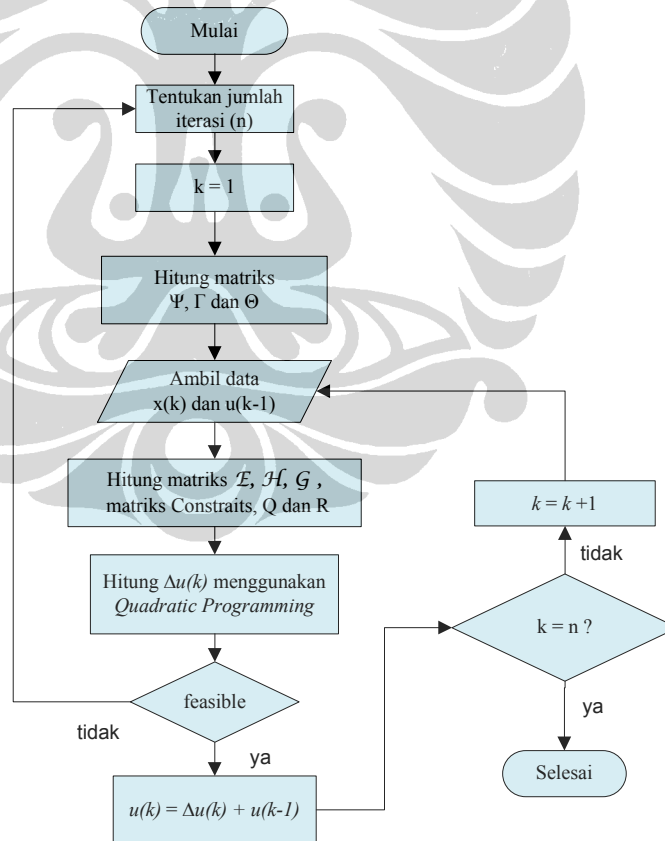
2. Matriks \mathcal{F} dihitung dengan menggunakan persamaan (2.25), serta matriks \mathcal{H} dan \mathcal{G} yang terdapat pada fungsi kriteria persamaan (2.29) dihitung masing-masing dengan menggunakan persamaan (2.31) dan (2.30).
3. Parameter batasan (*constraints*) fisik sistem diubah ke dalam bentuk pertidaksamaan yang memiliki hubungan dengan perubahan sinyal kendali (ΔU).

$$\underline{\Omega} \Delta U(k) \leq \underline{\omega} \quad (3.14)$$

4. Menghitung perubahan sinyal kendali optimal Δu_{opt} dengan menggunakan metode *Quadratic Programming*.
5. Menghitung sinyal kendali $u(k)$ dimana

$$u(k) = \Delta u(k) + u(k-1) \quad (3.15)$$

Diagram alir untuk perhitungan sinyal kendali dengan menggunakan MPC dengan *constraints* adalah seperti gambar 3.15.



Gambar 3.15. Diagram alir algoritma MPC dengan *Constraints*

Metode yang digunakan pada *Quadratic Programming* dalam menghitung nilai $\underline{\Delta U}$ adalah *Active Set*. Dengan menggunakan fungsi *fmincon* pada mfile MATLAB yang berfungsi untuk meminimasi fungsi kriteria dengan alur operasi seperti dijelaskan berikut ini.

1. Fungsi kriteria pada persamaan (2.37), diubah menjadi seperti berikut

$$V(\underline{\Delta U}(k)) = \frac{1}{2} \underline{\Delta U}^T(k) 2\underline{\mathcal{H}} \underline{\Delta U}(k) - \underline{\Delta U}^T(k) \underline{\mathcal{G}} \quad (3.16)$$

berdasarkan *constraints*

$$\underline{\Omega} \underline{\Delta U}(k) \leq \underline{\omega} \quad (3.17)$$

2. Nilai $\underline{\Delta U}_r$ dipilih sedemikian sehingga pertidaksamaan *constraints* (3.17) menjadi sebuah persamaan seperti berikut

$$\underline{\Omega}_r \underline{\Delta U}_r = \underline{\omega}_r \quad (3.18)$$

Elemen yang membuat pertidaksamaan menjadi persamaan disebut elemen aktif .

3. Menghitung nilai \underline{d} yang merupakan pergerakan $\underline{\Delta U}_r$ dalam meminimasi fungsi kriteria sehingga fungsi kriteria pada persamaan (3.16) berubah menjadi

$$\begin{aligned} V(\underline{\Delta U}_r + \underline{d}) &= \frac{1}{2} (\underline{\Delta U}_r + \underline{d})^T 2\underline{\mathcal{H}} (\underline{\Delta U}_r + \underline{d}) - (\underline{\Delta U}_r + \underline{d})^T \underline{\mathcal{G}} \\ &= \frac{1}{2} \underline{d}^T \underbrace{2\underline{\mathcal{H}}}_{\underline{\Phi}} \underline{d} + \underline{d}^T \underbrace{(2\underline{\mathcal{H}} \underline{\Delta U}_r - \underline{\mathcal{G}})}_{\underline{\phi}_r} + V(\underline{\Delta U}_r) \end{aligned} \quad (3.19)$$

Nilai \underline{d} tidak boleh mempengaruhi pertidaksamaan *constraints* (3.17), sehingga persamaan *constraints* untuk persamaan (3.19) adalah

$$\underline{\Omega}_r \underline{d} = \underline{0} \quad (3.20)$$

4. Dari persamaan (3.19) dan (3.20), nilai optimal \underline{d} sepanjang *constraints* yang aktif dapat dihitung dengan menyelesaikan fungsi kuadrat berikut

$$\min \frac{1}{2} \underline{d}^T \underline{\Phi} \underline{d} + \underline{d}^T \underline{\phi}_r \quad (3.21)$$

dengan *constraints*

$$\underline{\Omega}_r \underline{d} = \underline{0} \quad (3.22)$$

Nilai pengali *Lagrange* $\underline{\lambda}_r$ untuk persamaan (3.21) dan (3.22) dihitung berdasarkan kondisi *Karush-Kuhn-Tucker* (KKT) seperti berikut

$$\begin{bmatrix} \underline{\Phi} & \underline{\Omega}^T \\ \underline{\Omega} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{d} \\ \underline{\lambda}_r \end{bmatrix} = \begin{bmatrix} -\underline{\phi}_r \\ \underline{0} \end{bmatrix} \quad (3.23)$$

dimana nilai $\underline{\Delta U}_r$ yang terdapat pada matriks $\underline{\phi}_r$ ditentukan pada langkah (2). Hasil perhitungan \underline{d} dan $\underline{\lambda}_r$ akan mempengaruhi tahapan berikutnya, yaitu :

- a. Jika semua $\underline{\lambda}_r > 0$ dan $\underline{d} = 0$, maka proses komputasi selesai dan nilai $\underline{\Delta U}_r$ merupakan nilai optimal untuk $\underline{\Delta U}(k)$.
 - b. Jika semua $\underline{\lambda}_r > 0$ dan ada nilai $d \neq 0$, maka lanjut ke langkah (5).
 - c. Jika ada nilai $\lambda_r < 0$, maka *constraint* yang memiliki nilai λ_r paling negatif dibuang, kemudian lanjut ke langkah (5).
5. Nilai faktor koreksi pergerakan nilai optimal α_r dihitung dengan menggunakan rumus

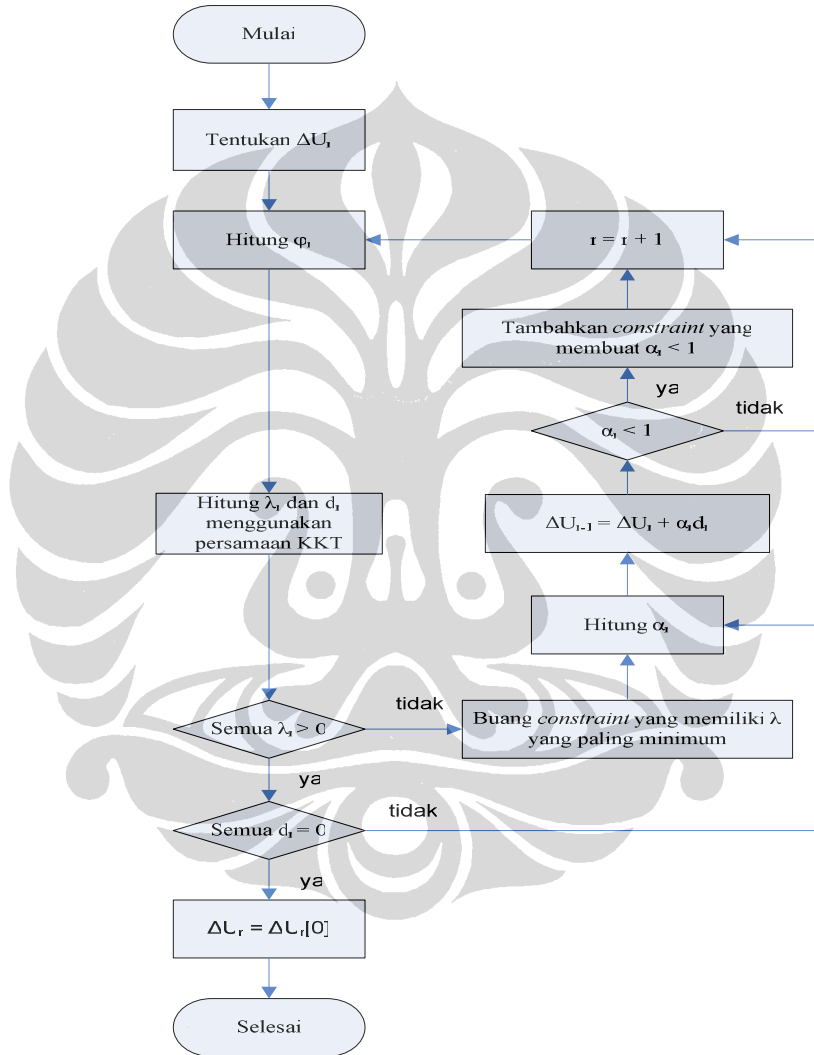
$$\alpha_r = \min \left(1, \min_{\substack{i \in \underline{\Omega}_r \\ a_i d > 0}} \frac{b_i - a_i \underline{\Delta U}_r}{a_i d} \right) \quad (3.24)$$

dengan a_i adalah baris dari pertidaksamaan batasan yang tidak aktif dan b_i adalah batasannya. Selanjutnya, nilai $\underline{\Delta U}_r$ dalam arah \underline{d} dihitung sebagai berikut

$$\underline{\Delta U}_{r+1} = \underline{\Delta U}_r + \alpha_r \underline{d} \quad (3.25)$$

6. Jika nilai $\alpha_r < 1$, maka *constraint* yang membuat nilai $\alpha_r < 1$ ditambahkan ke $\underline{\Omega}_r$.
7. Tetapkan $r = r + 1$ dan kembali ke langkah (3) untuk proses iterasi berikutnya.

Diagram alir metode *Active Set* untuk menyelesaikan *Quadratic Programming* seperti yang terdapat pada gambar 3.16.



Gambar 3.16. Diagram alir metode *Active Set* untuk menyelesaikan *Quadratic Programming*.

3.9. Perhitungan Sinyal Kendali untuk Sistem Heat Exchanger

Berikut ini adalah contoh langkah-langkah yang dilakukan untuk perhitungan sinyal kendali dengan metode MPC dengan *constraints*.

Parameter-parameter pengendali yang digunakan adalah sebagai berikut;

- Nilai *control horizon* (H_u) = 2
- Nilai *prediction horizon* (H_p) = 3
- Faktor Bobot Kesalahan $Q = 100$
- Faktor Bobot perubahan sinyal kendali $R = 0.11$
- Trayektori acuan (set point) $r(k) = [318.3; 313.6]$
- Batasan (constraint) yang digunakan:

$$-0.01 \leq \Delta u(k) \leq 0.01$$

$$0 \leq u(k) \leq 5$$

- Matriks variable keadaan dari persamaan (3.8):

$$x(k+1) = \underbrace{\begin{bmatrix} 0.9218 & 0.01489 \\ 0.01385 & 0.9272 \end{bmatrix}}_A x(k) + \underbrace{\begin{bmatrix} 0.1492 & -0.0006134 \\ 0.001105 & -0.07731 \end{bmatrix}}_B (u)$$

$$y(k) = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_C x(k)$$

Untuk mendapatkan sinyal kendali, algoritma MPC dengan *constraints* melakukan tahapan perhitungan sebagai berikut;

1. Menghitung matriks yang ada dalam persamaan ruang keadaan MPC, yaitu matriks $\underline{C}_y \underline{\Psi}$, $\underline{C}_y \underline{\Gamma}$, $\underline{C}_y \underline{\Theta}$, dengan menggunakan persamaan yang

telah dijabarkan dalam sub bab sebelumnya (2.16), yaitu:

$$\underline{C}_y \underline{\Psi} = \begin{bmatrix} 0.9218 & 0.0149 \\ 0.0138 & 0.0275 \\ 0.8499 & 0.0275 \\ 0.0256 & 0.8599 \\ 0.7838 & 0.0382 \\ 0.0355 & 0.7977 \end{bmatrix} \quad \underline{C}_y \underline{\Gamma} = \begin{bmatrix} 0.1492 & -0.0006 \\ 0.0011 & -0.0773 \\ 0.1375 & -0.0017 \\ 0.0031 & -0.0717 \\ 0.1268 & -0.0026 \\ 0.0048 & -0.0665 \end{bmatrix}$$

$$\underline{C}_y \underline{\Theta} = \begin{bmatrix} 0.1492 & -0.006 & 0 & 0 \\ 0.0011 & -0.0773 & 0 & 0 \\ 0.2867 & -0.0023 & 0.1492 & -0.0006 \\ 0.0042 & -0.149 & 0.0011 & -0.0773 \\ 0.4136 & -0.005 & 0.2867 & -0.0023 \\ 0.009 & -0.2155 & 0.0042 & -0.1490 \end{bmatrix}$$

2. Batasan yang telah ditentukan, terlebih dahulu dinyatakan dalam bentuk $\Delta u(k)$ berdasarkan persamaan dasar berikut;

$$u(k) = \Delta u(k) + u(k-1)$$

$$u(k+1) = \Delta u(k+1) + \Delta u(k) + u(k-1)$$

Untuk k selanjutnya, hanya dilakukan penambahan $\Delta u(k+i)$ hingga $i=Hu$

3. Matriks \underline{G} , dan \underline{H} masing-masing dihitung dengan menggunakan persamaan (2.28), dan (2.29). Dengan membuat nilai matriks \underline{Q} sama dengan \underline{I}_{100} dan matriks \underline{R} bernilai $\underline{I}_{0.1}$, maka matriks \underline{H} dapat dihitung sebagai berikut

$$\underline{H} = (\underline{C}_y \underline{\Theta})^T \underline{Q} \underline{C}_y \underline{\Theta} + \underline{R}$$

$$\underline{H} = \begin{bmatrix} 30.4204 & -0.6008 & 17.7563 & -0.3080 \\ -0.6008 & 8.3111 & -0.3129 & 4.8005 \\ 17.7563 & -0.3129 & 11.5955 & -0.1617 \\ -0.3080 & 4.8005 & -0.1617 & 3.2002 \end{bmatrix}$$

Untuk matriks \underline{G} , akan diperbarui setiap waktu seiring dengan perubahan vektor kesalahan $E(k)$ yang terjadi. Untuk $k = 1$;

$$\underline{G} = 2(\underline{\Theta} \underline{C}_y)^T \underline{Q} \underline{E}(1)$$

$$\underline{G} = \begin{bmatrix} 16.4120 & 0.1216 & 31.5424 & 0.4616 & 45.4946 & 0.9864 \\ -0.0675 & -8.5041 & -0.2563 & -16.3900 & -0.5478 & -23.7045 \\ 0 & 0 & 16.4120 & 0.1216 & 31.5424 & 0.4616 \\ 0 & 0 & -0.0675 & -8.5041 & -0.2563 & -16.3900 \end{bmatrix}$$

Jika pengendali MPC yang digunakan tidak menentukan batasan apapun (MPC *Noconstraints*), maka setelah langkah ini, langsung digunakan persamaan (2.31) untuk perhitungan $\Delta u(k)$ optimal di setiap langkah pencuplikan. Yang diperbarui setiap waktu adalah vektor kesalahan \underline{E} , perubahan sinyal kendali $\Delta u(k)$.

Dari persamaan (2.31)

$$\underline{\Delta U}(k)_{opt} = \frac{1}{2} \underline{\mathcal{H}}^{-1} \underline{\mathcal{G}}$$

$$\underline{\Delta U}(k)_{opt} = \begin{bmatrix} 206.3899 & 3.6398 & 517.1231 & 11.8474 & 803.6792 & 23.7611 \\ -4.3068 & -29.2714 & -10.9685 & -73.3568 & -17.7656 & -114.325 \\ 120.4289 & 1.9914 & 310.1807 & 6.657 & 485.1633 & 13.6109 \\ -2.2227 & -16.8849 & -5.7093 & -43.8606 & -9.3249 & -68.9206 \end{bmatrix}$$

sehingga didapat:

$$V(k) = 1.0e + 0.007 * \begin{bmatrix} 0.2133 & 0.0044 & 0.5382 & 0.0139 & 0.8379 & 0.0273 \\ 0.0044 & 0.0012 & 0.0112 & 0.0032 & 0.0174 & 0.005 \\ 0.5381 & 0.0112 & 1.3582 & 0.0352 & 2.1147 & 0.0689 \\ 0.0139 & 0.0032 & 0.0352 & 0.0081 & 0.0549 & 0.0131 \\ 0.8379 & 0.0174 & 2.1147 & 0.0549 & 3.2923 & 0.1073 \\ 0.0273 & 0.005 & 0.0689 & 0.0131 & 0.1073 & 0.0211 \end{bmatrix}$$

4. Untuk membuat *constraints* pada algoritma MPC, di buat sebuah matriks augmented yang mengandung semua unsur sinyal kendali untuk $\Delta u(k)$ yaitu $\Delta u(k), u(k-1), u(k)$ untuk k di sepanjang rentang *control horizon* yang telah ditentukan untuk semua masukan. Untuk sistem *Heat Exchanger*, terdapat dua buah masukan, sehingga untuk H_u sebesar 2, matriks augmented yang dibuat adalah sebagai berikut:

$$\underline{u}_{aug} = \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k+1) \\ u_1(k) \\ u_2(k) \\ u_1(k+1) \\ u_2(k+1) \\ u_1(k-1) \\ u_1(k-1) \end{bmatrix} \quad (3.25)$$

Berdasarkan matriks augmented ini, dibuat matriks pertidaksamaan dan persamaan yang akan berfungsi sebagai batasan perhitungan sinyal kendali sehingga berlaku hubungan berikut;

$$\begin{aligned} \underline{A} \underline{u}_{aug} &\leq \underline{B} \\ \underline{A}_{eq} \underline{u}_{aug} &= \underline{B}_{eq} \end{aligned} \quad (3.26)$$

Dimana \underline{A} dan \underline{B} merupakan matriks yang berisi nilai batasan maksimum dan minimum untuk semua nilai Δu dan u yang dihitung. Untuk perancangan ini, \underline{B}_{eq} dibuat sama dengan 0, sehingga matriks \underline{A}_{eq} mengandung hubungan antara setiap komponen \underline{u}_{aug} yang menghasilkan nilai 0. Dengan demikian, pengendali MPC akan mengoptimasi sinyal kendali yang dihasilkan agar memenuhi *constraints* persamaan dan pertidaksamaan tersebut.

Untuk pengendalian sistem *Heat Exchanger* yang telah dimodelkan, diberikan *constraints* terhadap *slew rate* sinyal kendali dan amplitudo sinyal kendali yang sama untuk u_1 dan u_2 . Dengan besar *constraints* sebagai berikut :

$$u \text{ minimum} = 0 \text{ volt dan } u \text{ maksimum} = 5 \text{ volt}$$

$$du \text{ minimum} = -0.01 \text{ dan } du \text{ maksimum} = 0.01$$

Berdasarkan *constraints* tersebut, komposisi matriks \underline{A} , \underline{B} , \underline{A}_{eq} , \underline{B}_{eq} di buat agar memenuhi persamaan dan pertidaksamaan *constraints* pada persamaan (3.26).

Untuk pertidaksamaan $\underline{A}u_{aug} \leq \underline{B}$

$$\underline{A}^* \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k+1) \\ u_1(k) \\ u_2(k) \\ u_1(k+1) \\ u_2(k+1) \\ u_1(k-1) \\ u_2(k-1) \end{bmatrix} \leq \begin{bmatrix} -du \text{ min} \\ -du \text{ min} \\ -du \text{ min} \\ -du \text{ min} \\ -u \text{ min} \\ -u \text{ min} \\ -u \text{ min} \\ -u \text{ min} \\ du \text{ max} \\ du \text{ max} \\ du \text{ max} \\ du \text{ max} \\ du \text{ max} \\ u \text{ max} \\ u \text{ max} \\ u \text{ max} \\ u \text{ max} \end{bmatrix} \longrightarrow \underline{A}^* \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k+1) \\ u_1(k) \\ u_2(k) \\ u_1(k+1) \\ u_2(k+1) \\ u_1(k-1) \\ u_2(k-1) \end{bmatrix} \leq \begin{bmatrix} -0.01 \\ -0.01 \\ -0.01 \\ -0.01 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.01 \\ 0.01 \\ 0.01 \\ 0.01 \\ 5 \\ 5 \\ 5 \\ 5 \end{bmatrix}$$

Diperoleh:

$$\underline{A} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Untuk persamaan *constraints* $\underline{A}_{eq} * \underline{u}_{aug} = \underline{B}_{eq}$

$$\underline{A}_{eq} * \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k+1) \\ u_1(k) \\ u_2(k) \\ u_1(k+1) \\ u_2(k+1) \\ u_1(k-1) \\ u_1(k-1) \end{bmatrix} = \text{zeros}(4,1)$$

Sehingga diperoleh :

$$\underline{A}_{eq} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix}$$

- Untuk menghitung $\Delta u(k)$, digunakan metode *quadratic programming* untuk perhitungan sinyal kendali. Pada perancangan ini, digunakan fungsi *fmincon* yang ada dalam Matlab. Fungsi ini meminimumkan sebuah fungsi terhadap *constraints* yang telah ditentukan. Dalam MPC, yang diminimumkan adalah fungsi kriteria. Pada setiap pencuplikan k , akan dihitung semua unsur sinyal kendali yang terdapat pada matriks augmented, namun pada MPC, nilai yang digunakan hanya nilai perubahan sinyal kendali pada saat k ($\Delta u(k)$), sedangkan nilai yang lain tidak digunakan. Untuk memperbarui sinyal kendali, nilai $\Delta u(k)$ inilah yang akan dijumlahkan dengan nilai sinyal kendali sebelumnya ($u(k-1)$). Misalkan, untuk $k=k+1$, perhitungannya sebagai berikut;

$$u(k+1) = \Delta u(k) + u(k-1)$$