



**UNIVERSITAS INDONESIA**

**PENANGANAN GANGGUAN PADA PENGENDALIAN  
KOLOM DISTILASI MENGGUNAKAN *REPRESENTATIVE  
MODEL PREDICTIVE CONTROL***

**SKRIPSI**

**BRAMANTYO**

**0806319324**

**FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK KIMIA  
DEPOK  
JULI 2012**



**UNIVERSITAS INDONESIA**

**PENANGANAN GANGGUAN PADA PENGENDALIAN  
KOLOM DISTILASI MENGGUNAKAN *REPRESENTATIVE  
MODEL PREDICTIVE CONTROL***

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**BRAMANTYO**

**0806319324**

**FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK KIMIA  
DEPOK  
JULI 2012**

## HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Bramantyo

NPM : 0806319324

Tanda Tangan : 

Tanggal : 4 Juli 2012

## HALAMAN PENGESAHAN

Skripsi diajukan oleh :  
Nama : Bramantyo  
NPM : 0806319324  
Program Studi : Teknik Kimia  
Judul Skripsi : Penanganan Gangguan pada Pengendalian Kolom  
Distilasi Menggunakan *Representative Model  
Predictive Control*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Kimia, Fakultas Teknik, Universitas Indonesia.

### DEWAN PENGUJI

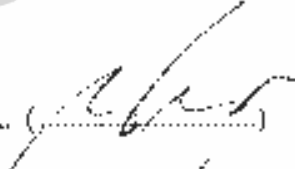
Penyisimbing : Ir. Abdul Wahid, M.T.

  
(.....)


Penguji I : Ir. Mahmud Sudibandriyo, M.Sc., Ph.D.

  
(.....)

Penguji II : Prof. Dr. Ir. Widodo W. Purwanto., DEA.

  
(.....)

Penguji III : Dr. Eny Kusriani, S.Si

  
(.....)

Ditampikan di : Depok

Pada tanggal : 4 Juli 2012

## KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah membimbing penulis, sehingga penulis dapat menyelesaikan skripsi ini dengan judul **“Penanganan Gangguan pada Pengendalian Kolom Distilasi Menggunakan *Representative Model Predictive Control*”** ini dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Departemen Teknik Kimia di Fakultas Teknik Universitas Indonesia.

Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Oleh karena itu, pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada:

- (1) Ir. Abdul Wahid, M.T., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk membimbing saya dalam penyusunan skripsi ini;
- (2) Prof. Dr. Ir. Setijo Bismo, DEA., selaku dosen pembimbing akademik yang telah membimbing dan membantu kehidupan akademik perkuliahan selama ini;
- (3) Para dosen dan karyawan Departemen Teknik Kimia FTUI yang telah berbagi ilmu dan wawasan;
- (4) Keluarga dan Myra yang selalu memberikan dukungan dan semangat;
- (5) Bratanata, Ernest, Afreza, Pindo, Adi, Fami, Victor, Gerry, Rendi, dan semua teman-teman DTK 08 yang tidak dapat disebutkan satu demi satu, yang selalu memberikan dukungan lahir dan bathin;

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini.

Depok, 4 Juli 2012

Bramantyo

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini,

Nama : Bramantyo  
NPM : 0806319374  
Program Studi : Teknik Kimia  
Departemen : Teknik Kimia  
Fakultas : Teknik  
Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Non-eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

**PENANGANAN GANGGUAN PADA PENGENDALIAN KOLOM  
DISTILASI MENGGUNAKAN *REPRESENTATIVE MODEL PREDICTIVE  
CONTROL***

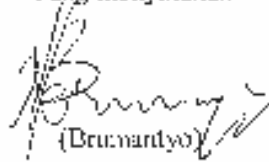
Dengan Hak Bebas Royalti Non-eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 4 Juli 2012

Yang menyatakan

  
(Bramantyo)

## ABSTRAK

Nama : Bramantyo  
Program Studi : Teknik Kimia  
Judul Skripsi : Penanganan Gangguan pada Pengendalian Kolom Distilasi Menggunakan *Representative Model Predictive Control*

---

Untuk menangani gangguan pada proses operasi nonlinear diperlukan suatu bentuk pengendalian. *Representative Model Predictive Control* (RMPC) adalah salah satu cara untuk memperoleh sekumpulan MPC lokal yang dapat merepresentasikan keseluruhan rentang operasi. MPC lokal ini nantinya digunakan pada *Multiple Model Predictive Control* (MMPC) untuk mensimulasikan proses operasi nonlinear multi variabel. Skripsi ini membahas penggunaan RMPC untuk memilih beberapa MPC lokal yang kemudian digunakan sebagai model pada MMPC untuk menangani gangguan. Penelitian ini menggunakan model kolom distilasi biner “Kolom A” yang disimulasikan dengan perangkat lunak MATLAB. Variabel yang dimanipulasi adalah laju refluks dan laju boil up sedangkan variabel yang dikontrol adalah komposisi produk distilat dan komposisi produk bawah. Hasil IAE MMPC dibandingkan dengan IAE kontroler PI konvensional. Untuk gangguan *single step*; MPC terbaik dengan IAE 0,2564, lebih baik dari IAE kontroler PI 0,7494. Sedangkan untuk gangguan *multi step*; MMPC terbaik dengan IAE 0,7730, lebih baik dari IAE kontroler PI 0,9808.

**Kata Kunci:** *Representative Model Predictive Control* (RMPC), *Multi Model Predictive Control* (MMPC), penanganan gangguan, pengendalian kolom distilasi, kolom A

## ABSTRACT

Name : Bramantyo  
Study Program : Chemical Engineering  
Title : Disturbance Handling on Distillation Column Control Using Representative Model Predictive Control

---

In order to handle disturbances in the nonlinear operation some form of control is required. Representative Model Predictive Control (RMPC) is one way to obtain a set of local MPC which able to represent the entire operating range. The local MPC is later used in the Multiple Model Predictive Control (MMPC) to simulate the operation of nonlinear multi-variable process. This thesis discusses the use of RMPC to select some local MPC which is then used as a model for dealing with disturbances in the MMPC. This study uses a model of a binary distillation column "Column A" which is simulated with MATLAB software. The manipulated variable is the rate of reflux and boil-up rate, while the controlled variable is the product composition of the distillate and bottom product composition. MMPC IAE results compared with conventional PI controller IAE. For single step disturbance; the best MPC with IAE 0.2564, is better than PI controller IAE 0.7494. As for the multi-step disturbance; the best MMPC with IAE 0.7730, is better than PI controller IAE 0.9808.

**Keywords:** *Representative Model Predictive Control (RMPC), Multi Model Predictive Control (MMPC), disturbance handling, distillation column control, Column A*



## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS .....	ii
HALAMAN PENGESAHAN .....	iii
KATA PENGANTAR .....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS .....	v
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xi
DAFTAR SINGKATAN .....	xii
DAFTAR LAMPIRAN .....	xiii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Batasan Masalah .....	3
1.5 Sistematika Penulisan .....	3
<b>BAB 2 TINJAUAN PUSTAKA .....</b>	<b>4</b>
2.1 Konsumsi Energi pada Sektor Industri di Indonesia .....	4
2.2 Pengendalian Kolom Distilasi .....	4
2.3 Model Kolom Distilasi .....	5
2.4 Model Ruang Keadaan ( <i>State Space Models</i> ) .....	5
2.5 Pengendalian Berumpan Balik .....	6
2.5.1 Diagram Blok untuk Loop Berumpan Balik .....	6
2.5.2 Algoritma <i>Proportional-Integral-Derivative</i> (PID) .....	8
2.5.2.1 Mode Proporsional .....	8
2.5.2.2 Mode Integral .....	9
2.5.2.3 Mode Derivative .....	9
2.5.2.4 Kontroler PID untuk Pengendalian Multivariabel .....	9
2.6 Model Predictive Control .....	11
2.6.1 Objektif MPC .....	15
2.6.2 Perbedaan Single Variable dengan Multivariable MPC .....	17
2.7 Multi Model Predictive Control .....	18
2.8 Representative Model Predictive Control .....	20
2.9 Disturbance Handling .....	21
<b>BAB 3 METODOLOGI PENELITIAN.....</b>	<b>22</b>
3.1 Alur Penelitian .....	22
3.2 Model dan Perangkat Lunak yang Digunakan .....	23
3.3 Prosedur Penelitian .....	23

<b>BAB 4 HASIL DAN PEMBAHASAN</b> .....	<b>29</b>
4.1 Identifikasi Sistem.....	29
4.2 Performa Kontroler PI.....	32
4.3 Performa Single Linear MPC dan Pemilihan MPC Lokal.....	34
4.3.1 Performa Single MPC dan Pemilihan MPC Lokal pada Skenario 1.....	36
4.3.2 Performa Single MPC dan Pemilihan MPC Lokal pada Skenario 2.....	37
4.4 Performa MMPC Berdasarkan Metode RMPC.....	39
4.4.1 Pengaruh Spesifikasi Switch.....	41
4.4.2 Performa MMPC pada Skenario 1.....	42
4.4.3 Performa MMPC pada Skenario 2.....	46
4.5 Diskusi.....	47
<b>BAB 5 KESIMPULAN DAN SARAN</b> .....	<b>55</b>
5.1 Kesimpulan.....	55
5.2 Saran.....	55
<b>DAFTAR REFERENSI</b> .....	<b>56</b>
<b>LAMPIRAN</b> .....	<b>58</b>



## DAFTAR GAMBAR

Gambar 2.1 Diagram Blok untuk Sistem Pengendali Berumpan Balik.....	7
Gambar 2.2 Struktur <i>Multi-loop</i> Diagonal ketika $m=3$ .....	10
Gambar 2.3 Konsep ‘ <i>Moving Horizon</i> ’ pada MPC .....	13
Gambar 2.4 Struktur <i>Model Predictive Control</i> .....	15
Gambar 2.5 Ilustrasi MMPC .....	19
Gambar 3.1 Diagram Alir Penelitian .....	22
Gambar 3.2 Model Kolom Distilasi Sederhana dengan Pengendali MPC dan MMPC.....	24
Gambar 3.3 <i>Process Reaction Curve</i> .....	25
Gambar 3.4 Grafik Respon CV dengan IAE.....	28
Gambar 4.1 Plant Kolom A.....	30
Gambar 4.2 PRC pengaruh perubahan F terhadap (a) $y_D$ , (b) $x_B$ .....	30
Gambar 4.3 PRC pengaruh perubahan $z_F$ terhadap (a) $y_D$ , (b) $x_B$ .....	30
Gambar 4.4 Plant Kolom A dengan Kontroler PI .....	33
Gambar 4.5 Performa kontroler PI .....	33
Gambar 4.6 Plant Kolom A nonlinear yang dikendalikan oleh kontroler single MPC .....	35
Gambar 4.7 Plant Kolom A dengan MMPC Dua Model.....	40
Gambar 4.8 Plant Kolom A dengan MMPC Tiga Model .....	40
Gambar 4.9 Spesifikasi Blok <i>Switch</i> pada Simulink .....	42
Gambar 4.10 Performa MPC1 pada $T_s = 0,29$ ; $P = 29$ ; $M = 13$ ; Skenario 1 .....	49
Gambar 4.11 Perbandingan MPC2, MPC3, dan MPC2.3; skenario 1.....	50
Gambar 4.12 Perbandingan MPC2, MPC3, MPC4, dan MPC2.3.4; skenario1 ....	50
Gambar 4.13 Perbandingan MPC2, MPC3, dan MPC2.3; skenario 2.....	51

## DAFTAR TABEL

Tabel 2.1 Karakteristik Sistem Pengendalian Multivariabel.....	18
Tabel 4.1 Parameter FOPDT .....	31
Tabel 4.2 Parameter MPC .....	35
Tabel 4.3 Integral Absolute Error dari Single MPC pada Skenario 1 .....	36
Tabel 4.4 Integral Square Error dari Single MPC pada Skenario 1 .....	37
Tabel 4.5 Integral Absolute Error dari Single MPC pada Skenario 2 .....	38
Tabel 4.6 Integral Square Error dari Single MPC pada Skenario 2 .....	39
Tabel 4.7 Kombinasi Model untuk Skenario 1.....	43
Tabel 4.8 IAE dan ISE Terbaik Setiap MMPC Dua Model pada Skenario 1 .....	44
Tabel 4.9 IAE dan ISE Terbaik Setiap MMPC Tiga Model pada Skenario 1 .....	45
Tabel 4.10 Kombinasi Model untuk Skenario 2.....	46
Tabel 4.11 IAE dan ISE Terbaik Setiap MMPC Dua Model pada Skenario 2.....	46
Tabel 4.12Perbandingan IAE dan ISE Kontroler PI, MPC, dan MMPC.....	52

## DAFTAR SINGKATAN

MPC	<i>Model Predictive Control</i>
NMPC	<i>Nonlinear Model Predictive Control</i>
MMPC	<i>Multi Model Predictive Control</i>
RMPC	<i>Representative Model Predictive Control</i>
IAE	<i>Integral Absolute Error</i>
ISE	<i>Integral Square Error</i>
PI	<i>Proportional-Integral</i>
CV	<i>Controlled Variable</i>
MV	<i>Manipulated Variable</i>
SP	<i>Set Point</i>
DMC	<i>Dynamic Matrix Control</i>
FOPDT	<i>First Order Plus Dead Time</i>
PRC	<i>Process Reaction Curve</i>

## DAFTAR LAMPIRAN

Lampiran 1 Program Identifikasi Sistem dengan Perubahan Feed Rate (F).....	58
Lampiran 2 Program Identifikasi Sistem dengan Perubahan Feed Composition (zF).....	61
Lampiran 3 Program Simulasi Kontroler PI .....	64
Lampiran 4 Program Simulasi Single MPC pada Skenario 1 .....	65
Lampiran 5 Respon $x_B$ pada $T_s = 0,72$ ; $P = 66$ ; $M = 26$ ; Skenario 1 .....	68
Lampiran 6 Respon $x_B$ pada $T_s = 0,53$ ; $P = 50$ ; $M = 20$ ; Skenario 1 .....	69
Lampiran 7 Respon $x_B$ pada $T_s = 0,36$ ; $P = 36$ ; $M = 15$ ; Skenario 1 .....	70
Lampiran 8 Respon $x_B$ pada $T_s = 0,29$ ; $P = 29$ ; $M = 13$ ; Skenario 1 .....	71
Lampiran 9 Respon $x_B$ pada $T_s = 0,1$ ; $P = 12$ ; $M = 2$ ; Skenario 1 .....	72
Lampiran 10 Program Simulasi Single MPC pada Skenario 2 .....	73
Lampiran 11 Respon $x_B$ pada $T_s = 0,72$ ; $P = 66$ ; $M = 26$ ; Skenario 2 .....	76
Lampiran 12 Respon $x_B$ pada $T_s = 0,53$ ; $P = 50$ ; $M = 20$ ; Skenario 2 .....	77
Lampiran 13 Respon $x_B$ pada $T_s = 0,36$ ; $P = 36$ ; $M = 15$ ; Skenario 2 .....	78
Lampiran 14 Respon $x_B$ pada $T_s = 0,29$ ; $P = 29$ ; $M = 13$ ; Skenario 2 .....	79
Lampiran 15 Respon $x_B$ pada $T_s = 0,1$ ; $P = 12$ ; $M = 2$ ; Skenario 2 .....	80
Lampiran 16 Program Simulasi MMPC Dua Model pada Skenario 1 .....	81
Lampiran 17 Program Simulasi MMPC Tiga Model pada Skenario 1.....	84
Lampiran 18 Respon $x_B$ pada MMPC Dua Model; Skenario 1 .....	88
Lampiran 19 Respon $x_B$ pada MMPC Tiga Model; Skenario 1 .....	89
Lampiran 20 Respon $x_B$ pada MMPC Dua Model; Skenario 1 .....	92
Lampiran 21 Respon $x_B$ pada MMPC Tiga Model; Skenario 1 .....	93
Lampiran 22 Program Simulasi MMPC Dua Model pada Skenario 2 .....	96
Lampiran 23 Respon $x_B$ pada MMPC Dua Model; Skenario 2 .....	99
Lampiran 24 Respon $x_B$ pada MMPC Dua Model; Skenario 2 .....	100

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Konsumsi energi nasional Indonesia sangat besar mencapai 470 ton oil ekivalen per juta US\$ PDB. Angka ini sangat tinggi dibandingkan Negara-negara Uni Eropa maupun Asia. Sebagai perbandingan, intensitas di Jepang hanya sekitar 92,3 ton oil ekivalen per juta US\$ PDB. Konsumsi energi khususnya dari sektor industri menduduki peringkat pertama, yaitu 40,86% dari keseluruhan konsumsi energi di Indonesia (ESDM, 2010). Sedangkan proses distilasi pada industri kimia, industri makanan, dan lain-lain dapat mengonsumsi energi 30%-50% lebih banyak dari yang biasa digunakan untuk memastikan tercapainya nilai kemurnian produk (Riggs, 2000). Proses distilasi di industri sangatlah kompleks dan pada umumnya bersifat nonlinear dengan sistem yang multivariabel dengan adanya interaksi antara satu variabel dengan variabel yang lain. Oleh karena itu, diperlukan adanya sistem pengendalian yang mampu mengendalikan dua variabel atau lebih dan juga mampu mengakomodasi sistem nonlinear. Pada kasus seperti ini, sistem pengendalian konvensional tidak memberikan hasil yang cukup baik ketika digunakan untuk mengendalikan sistem multivariabel. Padahal dengan pengendalian kolom distilasi yang tepat konsumsi energi dapat turun hingga 15% (Humphrey, 1991).

Untuk mengatasi masalah pengendalian kolom distilasi, Nonlinear MPC (NMPC) telah dikembangkan, namun metode ini sulit diimplementasikan di industri karena permodelan nonlinear yang rumit (Camacho, 2007; Qin, 2002). Sifat nonlinear inilah yang menyebabkan adanya gangguan atau *disturbance*. Oleh karena itu, diperlukan solusi alternatif untuk dapat mengatasi persoalan ini. Multiple MPC (MMPC) dikembangkan untuk dapat memprediksikan model-model linear yang berada dalam rentang operasi variabel yang dikontrol (Palma, 2004) Namun, masalah yang kemudian muncul adalah bagaimana cara menangani gangguan yang muncul dalam MMPC sehingga kolom distilasi dapat beroperasi dengan baik pada *set point* yang ditentukan. Untuk itu diperlukan suatu strategi yang dapat menentukan MPC lokal yang merepresentasikan titik-titik operasi

variabel yang dikontrol dengan baik, MPC lokal tersebut dinamakan *Representative MPC* (RMPC). Penelitian mengenai metode RMPC sejauh ini baru sampai kepada pemilihan MPC lokal yang baik untuk digunakan pada MMPC (Wahid, 2009) dan penggunaan RMPC pada MMPC (Cynthia, 2012) belum pada penggunaan RMPC tersebut untuk menangani gangguan pada kolom distilasi dengan pengendalian MMPC.

Pada penelitian ini, metode RMPC akan dilakukan terlebih dahulu untuk kemudian memilih beberapa lokal MPC dengan penanganan gangguan yang paling baik untuk digunakan pada metode MMPC. Metode RMPC yang digunakan didasarkan pada penelitian sebelumnya (Wahid, 2009). Penelitian ini akan menggunakan model kolom distilasi sistem biner metanol-air. Kemudian keberhasilan penelitian ini diuji dengan menghitung *Integral Absolute Error* (IAE) dari MMPC yang dihasilkan dan membandingkannya dengan nilai IAE dari sistem kontrol *Proportional Integral* (PI).

Penelitian ini akan membuktikan bahwa metode RMPC dapat digunakan sebagai suatu strategi untuk menangani gangguan yang terjadi dalam pelaksanaan metode MMPC. Dengan demikian MMPC dapat diterapkan untuk mengatasi gangguan yang disebabkan ketidaklinearan operasi dengan relatif mudah di industri. Penerapan MMPC sebagai salah satu teknik pengendalian yang *advance* diharapkan akan mampu mencapai tujuan utama dari proses distilasi, yaitu kemurnian produk dan di samping itu juga akan dapat menghemat konsumsi energi dalam proses distilasi.

## 1.2. Perumusan Masalah

Masalah yang hendak diselesaikan pada penelitian ini adalah bagaimana membuat MMPC berdasarkan RMPC pada beberapa rentang operasi untuk digunakan menangani gangguan pada pengendalian kolom distilasi?

## 1.3. Tujuan Penelitian

Membuktikan bahwa IAEMMPC lebih kecil dibandingkan dengan IAE kontroler konvensional PI dalam menangani gangguan pada pengendalian kolom distilasi.



#### 1.4. Batasan Masalah

Penelitian ini dibatasi dengan:

- 1) Kolom distilasi yang digunakan sebagai model adalah kolom A yang dikembangkan oleh Skogestad (1997). Kolom A tersebut memiliki 40 *theoretical stages* dan memisahkan campuran biner dengan volatilitas relatif 1,5 menjadi produk dengan kemurnian 99%.
- 2) Variabel input dan output berukuran 2x2, yaitu dua variabel input (*reflux flow* dan *boil up flow*), dan dua variabel output (fraksi mol produk atas dan bawah)
- 3) Fokus variabel yang dikendalikan adalah komposisi produk bawah ( $x_B$ )

#### 1.5. Sistematika Penulisan

Sistematika penulisan skripsi ini adalah sebagai berikut:

##### BAB I PENDAHULUAN

Berisi pendahuluan tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah dan sistematika penulisan.

##### BAB II TINJAUAN PUSTAKA

Berisi tinjauan pustaka yang membahas tentang pengendalian kolom distilasi, model kolom distilasi, model ruang keadaan, pengendalian berumpan balik, *Model PredictiveControl* (MPC), *Multi Model PredictiveControl* (MMPC), *Representative Model PredictiveControl* (RMPC), dan *disturbance handling*.

##### BAB III METODOLOGI PENELITIAN

Berisi tentang alur penelitian, variabel penelitian, model dan perangkat lunak yang digunakan, serta prosedur penelitian.

##### BAB IV HASIL DAN PEMBAHASAN

Berisi hasil penelitian berupa identifikasi sistem, performa kontroler PI, performa single linear MPC dan pemilihan MPC lokal, performa MMPC berdasarkan metode RMPC, dan diskusi.

##### BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan penelitian dan saran untuk penelitian selanjutnya.

## BAB 2 TINJAUAN PUSTAKA

### 2.1. Konsumsi Energi pada Sektor Industri di Indonesia

Sektor industri di Indonesia mengonsumsi sekitar 40,86% dari total konsumsi energi Indonesia. Angka tersebut merupakan angka konsumsi tertinggi di Indonesia melebihi sektor-sektor lain seperti transportasi (36,87%), sektor rumah tangga (13,26%), sektor komersial (4,73%), dan sektor lainnya (4,28%) (ESDM, 2010). Oleh karena itu, untuk mengkonservasi dan menghemat energi diperlukan sistem pengendalian yang tepat agar operasi industri dapat berjalan optimal

### 2.2. Pengendalian Kolom Distilasi

Kolom distilasi merupakan unit proses yang paling umum digunakan dalam operasi pemisahan zat di industri-industri kimia terutama untuk mendapatkan suatu produk dengan kemurnian tinggi. Namun, untuk mencapai tingkat kemurnian tersebut, seringkali industri melakukan *over-reflux* pada kolom distilasi. Hal ini mengakibatkan pemborosan energi 30-50% lebih besar dari energi yang biasanya dibutuhkan dalam proses distilasi. Telah dibuktikan sebelumnya bahwa dengan menerapkan pengendalian kolom distilasi yang tepat dapat menghemat energi keseluruhan yang digunakan sebesar 15% (Humphrey, 1991).

Pengendalian kolom distilasi sudah menjadi perhatian penting dalam industri kimia. Proses distilasi itu sendiri merupakan sistem *multiple-input multiple-output* (MIMO) nonlinear dengan interaksi antarvariabel yang kuat dan sensitif terhadap gangguan dari luar (Khaisongkram, 2006). Untuk mencari sistem pengendalian kolom distilasi yang terbaik, telah dilakukan perbandingan kinerja antara kontroler konvensional (PI) dan kontroler *advance* (MPC). Hasilnya menunjukkan bahwa MPC menghasilkan performa yang lebih baik dibandingkan PI dengan kondisi salah satu produk lebih penting atau berharga dibandingkan produk lain atau produk sampingnya, seperti kasus-kasus di industri pada umumnya. Performa MPC

juga jauh lebih baik dibandingkan PI karena kemampuannya dalam menangani batasan-batasan yang kompleks secara efektif (Riggs, 2000).

### 2.3. Model Kolom Distilasi

Model kolom distilasi yang digunakan adalah model kolom A yang dikembangkan oleh Skogestad. Kolom A tersebut memiliki 40 *theoretical stages* dan memisahkan campuran biner dengan volatilitas relatif 1,5 menjadi produk dengan kemurnian 99%. Model dinamik kolom ini menggunakan asumsi campuran biner, tekanan konstan, volatilitas relatif konstan, kesetimbangan di semua tingkat, kondenser total, laju molar konstan, tidak ada *holdup* uap, dan dinamika laju cairan yang dilinearisasi. Asumsi ini cocok dengan efek utama yang penting untuk dinamika dan pengendalian, kecuali asumsi tekanan konstan (Skogestad, 1997).

### 2.4. Model Ruang Keadaan (*State Space Models*)

Kebanyakan dinamika proses kimia dimodelkan secara nonlinear. Akan tetapi kebanyakan teknik pengendalian proses didasarkan pada model linear. Maka untuk melinearkan dinamika proses tersebut dapat digunakan model ruang keadaan. Bentuk umum dari model ruang keadaan adalah (Bequette, 1998):

$$\frac{dx_1}{dt} = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + b_{11}u_1 + \dots + b_{1m}u_m$$

sampai

$$(2.1)$$

$$\frac{dx_n}{dt} = a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + b_{n1}u_1 + \dots + b_{nm}u_m$$

$$y_1 = c_{11}x_1 + c_{12}x_2 + \dots + c_{1n}x_n + d_{11}u_1 + \dots + d_{1m}u_m$$

sampai

$$(2.2)$$

$$y_r = c_{r1}x_1 + c_{r2}x_2 + \dots + c_{rn}x_n + d_{r1}u_1 + \dots + d_{rm}u_m$$

Model ini mempunyai  $n$  variabel keadaan ( $x$ ),  $m$  variabel input ( $u$ ), dan  $r$  variabel output ( $y$ ). Variabel keadaan adalah keseluruhan variabel terikat (dipengaruhi oleh perubahan input), baik yang dikendalikan maupun yang tidak dikendalikan. Variabel output adalah beberapa variabel, yang

merupakan bagian dari variabel keadaan (bisa juga sama dengan variabel keadaan), yang dikendalikan dan merupakan fokus utama dalam simulasi. Variabel input adalah variabel bebas, bisa berupa gangguan maupun *manipulated variable*, yang mana perubahannya akan mempengaruhi variabel keadaan dan juga variabel output. Hubungan pada Persamaan 2.1 dan 2.2 biasanya dituliskan dalam bentuk matriks:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \\ \begin{bmatrix} y_1 \\ \vdots \\ y_r \end{bmatrix} &= \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{r1} & c_{r2} & \dots & c_{rn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{r1} & d_{r2} & \dots & d_{rm} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \end{aligned} \quad (2.3)$$

yang mempunyai bentuk (ruang keadaan) umum:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (2.4)$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \quad (2.5)$$

dimana titik di atas variabel keadaan mengindikasikan turunan terhadap waktu.

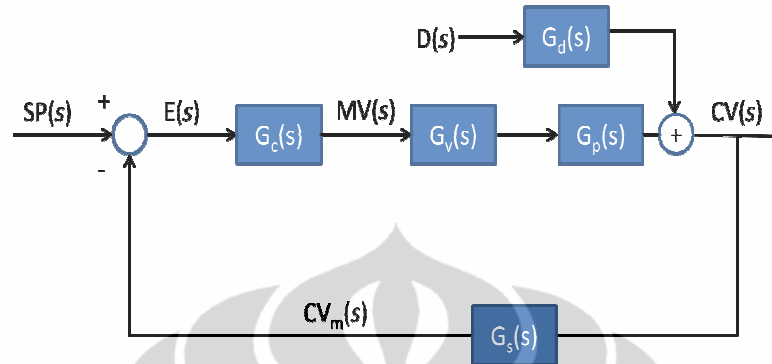
## 2.5. Pengendalian Berumpan Balik

Dalam pengendalian berumpan balik, kita mengenal adanya variabel input dan variabel output. Variabel input merupakan penyebab sedangkan variabel output merupakan akibat. Umpan balik menggunakan keadaan output untuk menentukan cara mempengaruhi input agar dapat menghasilkan variabel output yang sedekat mungkin dengan kondisi terbaik. Hal ini tentu saja bertujuan untuk mencapai objektif pengendalian tertentu yang diharapkan.

### 2.5.1. Diagram Blok untuk Loop Berumpan Balik

Diagram blok digunakan untuk memodelkan kombinasi model-model individu dari proses, instrumentasi, dan kontroler, yang semuanya

mempengaruhi perilaku dinamika sistem. Diagram blok untuk sistem pengendali berumpan balik ditunjukkan oleh Gambar 2.1.



Gambar 2.1. Diagram Blok untuk Sistem Pengendali Berumpan Balik  
(Marlin, 2000)

Keterangan:

**Fungsi Alih**

$G_c(s)$  = Kontroler

$G_v(s)$  = Transmisi, transducer, dan valve

$G_p(s)$  = Proses

$G_s(s)$  = Sensor, transducer, dan transmisi

$G_d(s)$  = *Disturbance* (gangguan)

**Variabel**

$CV(s)$  = *Controlled Variable*

$CV_m(s)$  = Harga terukur dari CV

$D(s)$  = *Disturbance* (gangguan)

$E(s)$  = *Error*

$MV(s)$  = *Manipulated variable*

$SP(s)$  = *Setpoint*

Elemen peralatan pada *loop* berumpan balik terbagi menjadi tiga fungsi alih: valve atau elemen akhir,  $G_v(s)$ ; proses,  $G_p(s)$ ; dan sensor,  $G_s(s)$ . Elemen penghitung adalah kontroler,  $G_c(s)$ . Variabel output proses yang dikendalikan dinamakan *controlled variable*,  $CV(s)$ , dan variabel input proses yang diatur oleh sistem kontrol dinamakan *manipulated variable*,  $MV(s)$ . Harga yang diinginkan, yang harus ditentukan secara independen untuk kontroler, dinamakan *setpoint*,  $SP(s)$ . Selisih antara *setpoint* dengan harga CV terukur disebut sebagai *error*,  $E(s)$ . Suatu input yang berubah karena pengaruh dari luar dan mempengaruhi CV dinamakan *disturbance* (gangguan),  $D(s)$ , dan hubungan antara *disturbance* dan CV adalah fungsi alih *disturbance*,  $G_d(s)$ .

Berikut ini merupakan fungsi alih loop tertutup untuk sebuah loop berumpan balik:

$$\text{Disturbance response: } \frac{CV(s)}{D(s)} = \frac{G_d(s)}{1+G_p(s)G_v(s)G_c(s)G_s(s)} \quad (2.6)$$

$$\text{Setpoint response: } \frac{CV(s)}{SP(s)} = \frac{G_p(s)G_v(s)G_c(s)}{1+G_p(s)G_v(s)G_c(s)G_s(s)} \quad (2.7)$$

### 2.5.2. Algoritma *Proportional-Integral-Derivative*(PID)

Kontroler *Proportional Integral* (PI) adalah penggabungan dua metode kontrol *Proportional* dan *Integral*. Kontrol PI umum digunakan dalam industri, hal ini karena kontroler ini menggabungkan beberapa keuntungan pengendalian, antara lain, respon waktu yang lebih cepat karena kontrol *Proportional* (P), bersamaan dengan *offset* yang dihapuskan oleh kontroler *Integral* (I). Oleh karena itu, kontroler ini digolongkan sebagai kontroler konvensional.

Setiap elemen pada algoritma PID disebut sebagai *mode* dan menggunakan informasi umpan balik yang bergantung pada waktu dalam berbagai cara sesuai dengan namanya masing-masing: *Proportional-Integral-Derivative*.

#### 2.5.2.1. Mode Proporsional

Pada mode proporsional, kontrol (pengaturan terhadap *manipulated variable*) dibuat proporsional terhadap sinyal error, dimana jika error meningkat, pengaturan terhadap MV juga harus meningkat.

$$\text{Mode proporsional: } MV_p(t) = K_c E(t) + I_p \quad (2.8)$$

$$G_c(s) = \frac{MV_p(s)}{E(s)} = K_c \quad (2.9)$$

Gain controller,  $K_c$ , merupakan parameter pertama dari tiga parameter yang dapat diatur untuk membuat controller PID dapat digunakan dalam berbagai aplikasi. Satuan dari  $K_c$  adalah  $[manipulated]/[controlled] \text{ variable}$ . Persamaan untuk mode proporsional mengandung suku konstan,  $I_p$ , yang digunakan pada waktu mula-mula algoritma dijalankan, dimana pada waktu awal ini nilai dari *manipulated variable* tetap.

$$I_p = [MV(t) - K_c E(t)]_{t=0} \quad (2.10)$$

### 2.5.2.2. Mode Integral

Mode integral dapat mengatur *manipulated variable* sampai besarnya error berkurang menjadi nol untuk input step.

$$\text{Mode integral: } MV_I(t) = \frac{K_c}{T_I} \int_0^t E(t') dt' + I_I \quad (2.11)$$

$$G_c(s) = \frac{MV_I(s)}{E(s)} = \frac{K_c}{T_I s} \quad (2.12)$$

Waktu integral,  $T_I$ , merupakan parameter yang dapat diatur untuk mode integral ini. Satuan  $T_I$  adalah satuan waktu.

### 2.5.2.3. Mode Derivatif

Mode derivatif baik digunakan apabila terdapat gangguan yang mempengaruhi *controlled variable* dimana error dan integral error yang mendekati nol menyebabkan dua mode lain tidak melakukan penyesuaian terhadap *manipulated variable*. Padahal, nilai *controlled variable* akan mengalami perubahan dan penyimpangan terhadap *setpoint*-nya di waktu yang akan datang sehingga diperlukan perubahan *manipulated variable* yang dilakukan oleh mode derivatif ini.

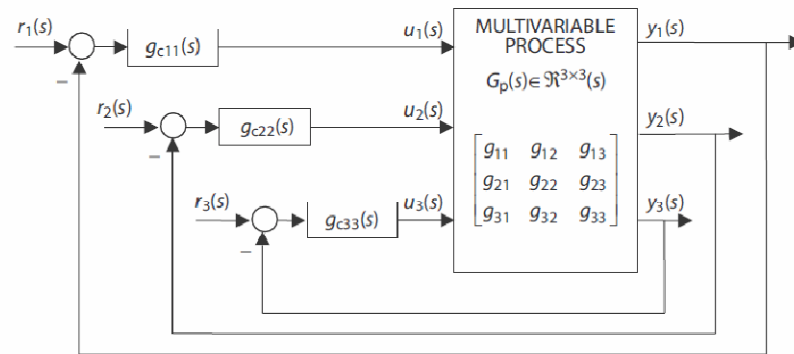
$$\text{Mode derivatif: } MV_d(t) = K_c T_d \frac{dE(t)}{dt} + I_d \quad (2.13)$$

$$G_c(s) = \frac{MV_d(s)}{E(s)} = K_c T_d s \quad (2.14)$$

Waktu integral,  $T_d$ , merupakan parameter yang dapat diatur untuk mode derivatif ini. Satuan  $T_d$  adalah satuan waktu.

### 2.5.2.4. Kontroler PID untuk Pengendalian Multivariabel

Kontroler SISO *multi-loop* sering digunakan untuk mengendalikan proses multivariabel karena strukturnya yang sederhana dan kemudahan dalam menangani kegagalan sebuah loop dengan mengisolir loop tersebut. Struktur *multi-loop* ini disebut juga struktur terdesentralisasi. Kontroler yang dapat digunakan dalam struktur terdesentralisasi antara lain kontroler multivariabel segitiga atas dan bawah, kontroler multivariabel diagonal band, dan kontroler multivariabel diagonal. Struktur diagonal sangat disukai karena struktur loop yang sangat sederhana (Johnson, 2005). Struktur ini ditunjukkan oleh Gambar 2.2.



Gambar 2.2. Struktur *Multi-loop* Diagonal ketika  $m=3$  (Johnson, 2005)

Walaupun struktur *multi-loop* ini sederhana, terdapat interaksi antar-loop yang menyebabkan sulitnya mendesain kontroler yang memenuhi spesifikasi (Huang, 2003). Banyak metode yang telah dipublikasikan untuk desain kontroler. Metode-metode tersebut dapat diklasifikasikan menjadi lima tipe, yaitu:

1. Metode detuning
2. Metode penutupan loop berurutan
3. Metode iteratif atau trial-and-error
4. Metode penyelesaian persamaan secara simultan atau optimisasi
5. Metode independen

Dalam penelitian ini, digunakan dua kontroler PI single-loop sebagai pembanding kinerja MPC. Konfigurasinya adalah  $u_1 u_2$ , dimana  $u_1$  (atau L) digunakan untuk mengendalikan  $y_D$ , dan  $u_2$  (atau V) digunakan untuk mengendalikan  $x_B$ . Jadi, disini akan digunakan dua buah kontroler PI dengan desain yang telah ditentukan bagi Kolom A dengan konfigurasi LV untuk pengendalian komposisi dua titik (Skogestad, 1997). Desain yang digunakan adalah sebagai berikut:

$$k_1(s) = 26,1 \frac{1+3,76s}{3,76s} \quad (y_D - L \text{ loop}) \quad (2.15)$$

$$k_2(s) = -37,5 \frac{1+3,31s}{3,31s} \quad (x_B - V \text{ loop}) \quad (2.16)$$



## 2.6. Model Predictive Control

*Model Predictive Control* (MPC), atau yang biasa disebut dengan *moving/receding horizon control* ditemukan pertama kali pada akhir tahun tujuh puluhan dan telah dikembangkan sejak saat itu sampai sekarang serta menjadi *feedback strategy* yang menarik, terutama untuk proses linear. Istilah MPC bukanlah suatu strategi kontrol yang spesifik, melainkan lebih kepada metode kontrol yang jelas, yang menggunakan sebuah model dari suatu proses untuk mendapatkan sinyal kontrol dari proses tersebut dengan mengurangi fungsi objektifnya. MPC menjadi sering digunakan karena pada kenyataannya MPC merupakan cara yang paling mudah untuk mengatasi masalah pengendalian terhadap waktu. Penggunaan strategi *finite-horizon* memudahkan pengendalian suatu proses dengan menggunakan MPC.

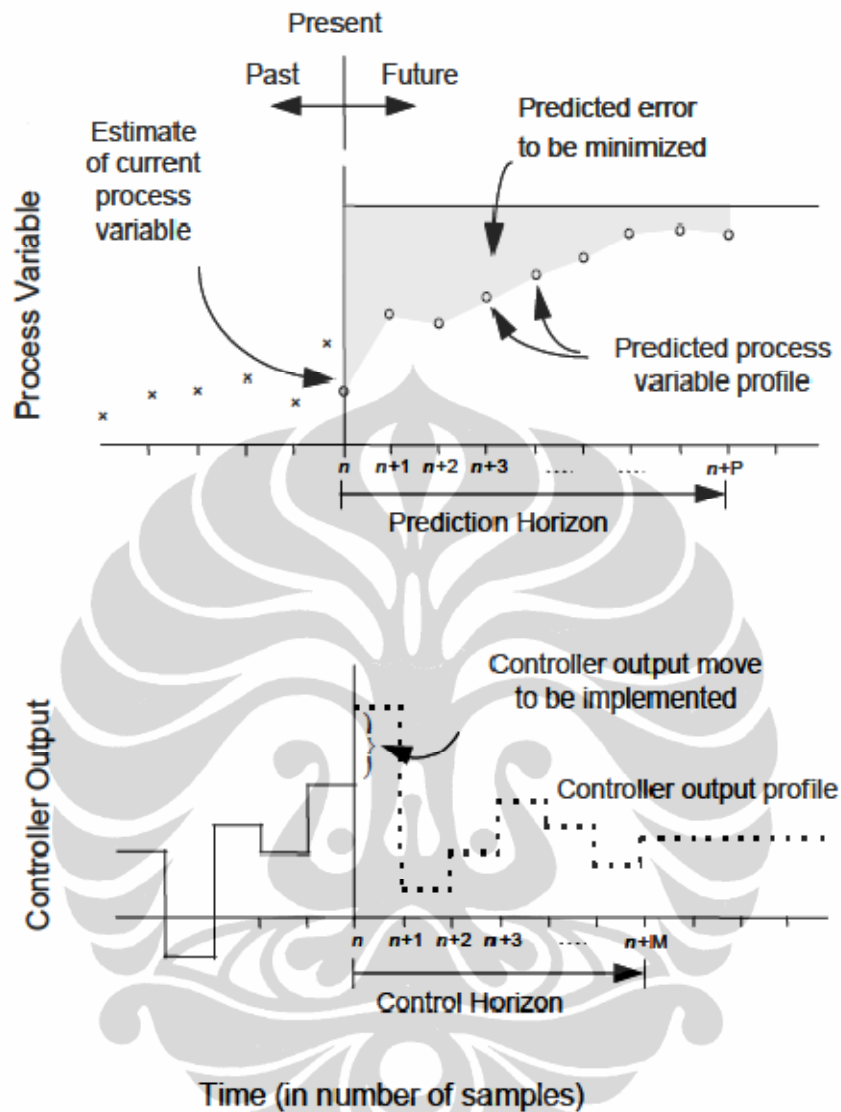
MPC menjadi terkenal, terutama pada industri-industri kimia, dikarenakan algoritma yang digunakan sederhana, serta penggunaan model *impulse response* dan model *step response* yang membutuhkan sedikit informasi. MPC juga dianggap sebagai suatu teknik matang untuk digunakan pada sistem linear dan juga *slow-systems* seperti yang sering ditemui pada industri. Sistem yang lebih kompleks (*nonlinear*, *hybrid*, proses yang sangat cepat) sulit diselesaikan dengan menggunakan MPC. Meskipun begitu, jumlah penggunaan MPC pada sistem *nonlinear* semakin bertambah.

Proses pada industri adalah *nonlinear*, tetapi banyak aplikasi MPC didasarkan pada model linear, dengan alasan pengidentifikasian terhadap model linear dari suatu proses relatif mudah dan model linear menunjukkan hasil yang baik, apabila plant beroperasi disekitar *operating point*. Di banyak situasi, operasi pada sebuah proses membutuhkan pemindahan dari suatu *operating point* ke *operating point* lainnya. Dengan kata lain, proses berjalan secara *nonlinear*. Sehingga, perlu dibuat suatu model *nonlinear* untuk menyelesaikan masalah ini. (Camacho, 2007 )

Pada dasarnya, MPC menggunakan model proses (baik linear maupun *nonlinear*) untuk menghitung perubahan terbaik bagi *manipulated variables* (MV) sehingga dapat mencapai hasil tertentu yang diinginkan pada

*controlled variables* (CV). MPC telah diaplikasikan pada sebagian besar industri *petroleum* pada unit-unit tertentu antara lain: *fluid catalytic crackers*, *hydrocrackers*, dan *petroleum fractionating towers*. Dimana operasi pada unit-unit tersebut bersifat multivariabel, memiliki banyak batasan, dan memproses material dalam volume besar (Luyben, 1997).

MPC adalah salah satu algoritma pengendalian yang menggunakan model eksplisit untuk memprediksikan perilaku sistem di waktu yang akan datang sepanjang horison prediksi yang diperluas. Algoritma ini dirumuskan sebagai sebuah performa fungsi objektif yang didefinisikan sebagai kombinasi lintasan *setpoint* dan usaha kontrol. Fungsi objektif ini diminimalisasi dengan menghitung profil output kontroler yang bergerak di sepanjang horison kontrol. Pergerakan output kontroler yang pertama diterapkan kemudian seluruh prosedur diulang pada sampling berikutnya. Gambar 2.3 mengilustrasikan teknik '*moving horizon*' yang digunakan pada MPC (Dougherty, 2003b).



Gambar 2.3. Konsep 'Moving Horizon' pada MPC (Dougherty, 2003b)

Strategi yang dipergunakan pada MPC adalah sebagai berikut:(Camacho, 2007)

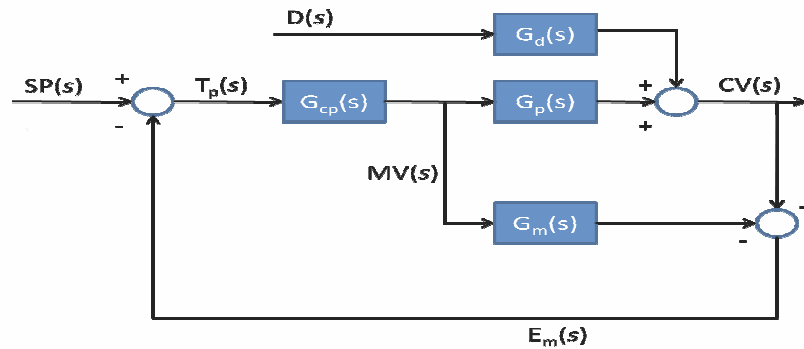
- 1) Output di waktu yang akan datang untuk horizon  $P$  yang ditentukan, disebut horizon prediksi, diprediksikan pada setiap  $n$  menggunakan model proses. Output yang diprediksikan ini tergantung pada harga yang telah diketahui sampai pada  $n$  (input dan output masa lalu) dan sinyal kontrol masa depan, yang semuanya dikirim ke sistem dan dihitung.

- 2) Kumpulan sinyal kontrol di waktu yang akan datang dihitung dengan mengoptimisasi suatu kriteria yang ditentukan dengan tujuan untuk menjaga proses sedekat mungkin dengan sinyal referensi (yang dapat berupa *setpoint*-nya sendiri atau pendekatannya). Kriteria ini biasanya mengambil bentuk fungsi kuadrat dari error antara sinyal output yang diprediksi dan sinyal referensi.
- 3) Sinyal kontrol dikirim ke proses ketika sinyal kontrol yang dihitung selanjutnya ditolak, karena pada waktu sampling berikutnya output telah diketahui dan langkah 1 diulang dengan harga baru ini dan semua urutan perhitungan memiliki nilai baru.

Tiga karakteristik penting pendekatan MPC antara lain:

- Menggunakan model proses untuk menentukan perubahan yang tepat pada MV karena perilaku CV kemudian dapat diprediksikan dari harga MV.
- Informasi *feedback* yang penting adalah selisih respon model prediksi dengan respon proses aktual. Jika selisihnya sama dengan nol, maka pengendalian sudah sempurna dan tidak diperlukan perbaikan.
- Pendekatan dengan feedback ini akan menyebabkan CV mendekati *setpoint* setelah beberapa iterasi.

Gambar 2.4 menunjukkan struktur MPC. Tiga buah fungsi alih merepresentasikan proses sesungguhnya dengan elemen akhir dan sensor,  $G_p(s)$ ; kontroler,  $G_{cp}(s)$ ; dan model dinamik dari proses,  $G_m(s)$ . Seluruh perhitungan pada sistem kendali prediktif selalu dilakukan setiap kali harga dari elemen final ditentukan. Sinyal feedback  $E_m$  merupakan perbedaan harga CV terukur dengan prediksinya. Sinyal feedback dianggap sebagai koreksi dari model yang digunakan untuk mengoreksi *setpoint* untuk memperoleh harga target yang lebih baik (Marlin, 2000).



Gambar 2.4. Struktur Model Predictive Control (Marlin, 2000)

### 2.6.1. Objektif MPC

Tugas dari algoritma kontrol adalah untuk menentukan perubahan *manipulated variable* di waktu yang akan datang yang akan menghasilkan *controlled variable* yang diprediksi dapat kembali dengan cepat ke *setpoint*-nya. Untuk menentukan pengendalian yang terbaik untuk *manipulated variable*, pengukuran terhadap usaha pengendalian perlu dipilih. Usaha pengendalian yang diukur adalah berupa error, yaitu penyimpangan *controlled variable* dari *setpoint*. Error pada waktu tertentu dapat diukur, tetapi error tersebut akan berubah karena adanya pengaturan terhadap *manipulated variable*. Perilaku *controlled variable* tanpa adanya pengaturan di waktu yang akan datang berguna untuk mengetahui error di waktu yang akan datang.

$$CV_i^f = CV_K^* + \sum_{j=1}^{LL} a_{j+i} \Delta MV_{K-j} \quad (\text{tanpa umpan balik}) \quad (2.17)$$

dimana:

$CV_i^f$  = harga *controlled variable* yang diprediksi di waktu yang akan datang yang harganya dipengaruhi oleh perubahan *manipulated variable* di masa lalu

$CV_K^*$  = harga *controlled variable* yang diprediksi di waktu sekarang berdasarkan seluruh input di masa lalu sampai  $K - LL$

$i$  = periode sampel di waktu yang akan datang ( $i = 1$  sampai  $NN$ )

Selisih dari harga *controlled variable* yang diprediksi terhadap *setpoint* digunakan untuk menghitung objektif, yang adalah jumlah kuadrat error yang harus diminimalisasi.

$$OBJ = \sum_{i=1}^{NN} [SP_i - (CV_i^f + CV_i^c)]^2 = \sum_{i=1}^{NN} [E_i^f - CV_i^c]^2 \quad (2.18)$$

dimana:

$SP_i$  = *setpoint* setiap sampel  $i$  di waktu yang akan datang

$CV_i^f$  = diperoleh dari persamaan (2.49) yang tidak dapat dipengaruhi oleh kontroler

$CV_i^c$  = pengaruh pengendalian di waktu yang akan datang pada *controlled variable* pada setiap sampel  $i$

$E_i^f$  = penyimpangan dari *setpoint* yang akan terjadi jika tidak ada pengendalian

$NN$  = batas waktu dimana kerja kontrol akan dievaluasi, disebut horizon output

Usaha optimisasi dilakukan dengan mencari harga pengaturan *manipulated variable* di waktu yang akan datang sehingga nilai objektif sekecil mungkin. Model step respon dapat digunakan untuk menghitung pengaruh perubahan di waktu yang akan datang, sebagai berikut:

$$CV_{i+1}^c = \sum_{j=1}^{i+1} a_j \Delta MV_{i-j+1}^c \quad (2.19)$$

dimana:

$CV^c$  = harga *controlled variable* yang dipengaruhi pengaturan *manipulated variable* di waktu yang akan datang

$\Delta MV^c$  = pengaturan *manipulated variable* di waktu yang akan datang yang dihitung oleh kontroler

Pada situasi multivariabel, objektif kontroler menjadi:

$$OBJ = \sum_{nc=1}^{NC} ww_{nc} \sum_{i=1}^{NN} [E_{nc,i}^f - CV_{nc,i}^c]^2 + \sum_{nm=1}^{NM} qq_{nm} \sum_{i=1}^{MM} (\Delta MV_{nm,i})^2 \quad (2.20)$$

dimana:

$NC$  = jumlah *controlled variable*;  $nc$  adalah urutan *controlled variable* (1 sampai  $NC$ )

$NM$  = jumlah *manipulated variable*;  $nm$  adalah urutan *controlled variable* (1 sampai  $NM$ )

$ww_{nc}$  = parameter yang dapat diatur, yang menunjukkan penyimpangan *controlled variable* ke- $nc$  terhadap *setpoint*

$qq_{nm}$  = parameter yang dapat diatur, yang menunjukkan pengaturan *manipulated variable* ke- $nm$

### 2.6.2. Perbedaan Single Variable dengan Multivariable MPC

Pengendalian multivariabel bertujuan untuk mengendalikan beberapa *controlled variable* pada *setpoint* yang independen. Mengendalikan sistem multivariabel memerlukan analisis yang lebih kompleks dibandingkan dengan pengendalian variabel tunggal. Akan tetapi, metode yang digunakan untuk sistem variabel tunggal masih dapat diaplikasikan untuk sistem multivariabel. Aspek-aspek pada sistem variabel tunggal yang memudahkan atau menyulitkan pengontrolan memiliki pengaruh yang sama untuk sistem multivariabel. Pada sistem multivariabel, interaksi antar variabel sangat penting untuk diperhatikan. Interaksi antar variabel ini menyebabkan satu *manipulated variable* dapat mempengaruhi lebih dari satu *controlled variable*. Tabel 2.1 berikut ini menunjukkan karakteristik sistem pengendalian multivariabel.

Tabel 2.1. Karakteristik Sistem Pengendalian Multivariabel (Marlin, 2000)

No	Karakteristik Pengendalian Variabel Tunggal yang mendukung untuk Pengendalian Sistem Multivariabel	Karakteristik Unik pada Sistem Multivariabel
1.	Proses umpan balik yang cepat ( $\theta + \tau$ yang kecil)	Interaksi antara variabel mempengaruhi stabilitas dan performa pengendalian
2.	Proses umpan balik dengan fraksi <i>dead time</i> ( $\theta/(\theta + \tau)$ ) yang kecil dan tanpa respon balik	Kelayakan pengendalian bergantung pada keseluruhan proses, tidak hanya pada hubungan sebab-akibat tertentu
3.	Gangguan yang kecil yang jauh dari frekuensi kritis	Dalam merancang strategi pengendalian harus memperhatikan sumber dari gangguan bukan hanya besarnya saja
4.	Tidak ada batasan pada <i>manipulated variable</i>	Pasangan variabel terukur dan elemen akhir melalui pengendalian adalah keputusan perancangan
5.	Kontroler digital dengan periode eksekusi relatif cepat	Beberapa proses memiliki jumlah <i>controlled variable</i> dan <i>manipulated variable</i> yang tidak sama
6.	Kontroler berdasarkan model yang akurat	Beberapa desain pengendalian multivariabel sangat sensitif terhadap kesalahan pemodelan
7.	Kontroler menggunakan penunjang-penunjang yang tepat	

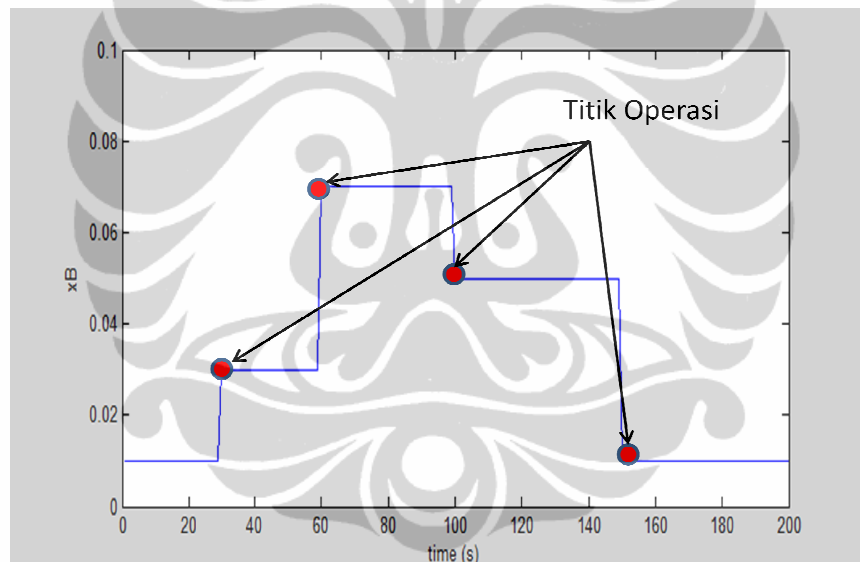
## 2.7. Multi Model Predictive Control

*Nonlinear dynamics* hampir terdapat pada semua aplikasi *engineering*. Membuat suatu model yang memadai untuk suatu proses nonlinear sangat sulit dilakukan dan di dunia ini tidak ada model yang cocok untuk merepresentasikan proses nonlinear. Untuk mengatasi masalah pengendalian



kolom distilasi, Nonlinear MPC (NMPC) telah dikembangkan, namun metode ini sulit diimplementasikan di industri karena permodelan nonlinear yang rumit (Camacho, 2007; Qin, 2002). Oleh karena itu, diperlukan solusi alternatif dan salah satunya adalah Multiple MPC (MMPC). Prinsip dari MMPC ini adalah membuat beberapa model linear dari berbagai operating point pada suatu proses industri. Model-model linear yang telah dibuat tadi kemudian digunakan secara bersama-sama dalam pengendalian proses sehingga model nonlinear dari proses industri tersebut mampu direpresentasikan, serta mampu mengurangi hal-hal yang tidak diinginkan dari proses nonlinear. (Palma, 2004).

Ilustrasi untuk MMPC ditunjukkan oleh Gambar 2.5 berikut ini.



Gambar 2.5. Ilustrasi MMPC

Sebuah contoh sistem nonlinear digambarkan oleh ilustrasi Gambar 2.5 di atas. Suatu *controlled variable*, misalnya  $x_B$ , memiliki rentang operasi antara 0,01 sampai 0,1. Di antara rentang operasi ini kita dapat memilih beberapa titik operasi misalnya 0,01; 0,03; 0,05; dan 0,07. MMPC mendekati sistem nonlinear – yang seharusnya dimodelkan dengan model nonlinear – dengan beberapa model linear berbeda di setiap titik operasi yang telah ditentukan. Jadi, jika sistem berada pada titik operasi 0,01 misalnya, maka

model linear pada titik 0,01 yang bekerja dengan baik, sama halnya untuk titik-titik operasi lainnya. Maka, apabila sistem berubah sesuai input step dari titik operasi satu ke titik operasi lainnya, model linear yang digunakan juga akan berubah sesuai dengan titik operasi masing-masing model linear tersebut. Dengan demikian MMPC mampu mendekati ketidaklinearan sistem dengan menggunakan beberapa model linear lokal.

Penelitian mengenai penggunaan MMPC pada proses separasi biner menunjukkan hasil yang lebih baik daripada penggunaan *single linear* MPC. Pada penelitian tersebut digunakan enam model linear yang diperoleh pada titik-titik operasi yang berbeda pada sistem. Adapun kekurangan yang ditemui dari metode ini adalah waktu komputasi yang relatif lebih lama dibandingkan dengan waktu komputasi kontroler konvensional. Hal ini dikarenakan untuk menggunakan algoritma MMPC kita harus mengidentifikasi model-model linear lokal yang merepresentasikan sistem pada berbagai titik operasi (Porfirio, 2003).

## 2.8. Representative Model Predictive Control

Seperti yang telah dijelaskan sebelumnya, masalah dalam penggunaan MMPC adalah bagaimana mendapatkan model-model linear lokal untuk setiap titik operasi dalam keseluruhan rentang operasi sehingga MMPC dapat merepresentasikan keseluruhan rentang operasi variabel yang dikontrol, model linear lokal tersebut dinamakan MPC lokal atau *Representative* MPC (RMPC). Penelitian mengenai penggunaan algoritma RMPC telah dilakukan sebelumnya dengan menggunakan sepuluh model (MPC lokal) berdasarkan rentang perubahan *setpoint*. Kemudian masing-masing model tersebut diuji untuk melihat responnya terhadap perubahan *setpoint*. Hasil penelitian tersebut menyarankan tiga dari sepuluh buah MPC lokal yang diuji untuk digunakan pada MMPC. Adanya perbedaan respon dari masing-masing MPC lokal menunjukkan MPC lokal dapat benar-benar merepresentasikan keseluruhan rentang operasi (Wahid, 2009).

## 2.9. Disturbance Handling

Proses distilasi yang nonlinear tidak akan terpisahkan dari adanya gangguan. Beberapa gangguan seperti laju aliran, suhu, atau komposisi aliran masuk (dan kadang-kadang yang keluar dari) proses, tidak bisa dengan bebas untuk dimanipulasi. Gangguan yang muncul tergantung kepada kondisi bagian hulu atau hilir dari pabrik. Sistem kontrol harus dapat mengendalikan keadaan pabrik walaupun efek gangguan ini berdampak pada proses operasi (Luyben, 1997).

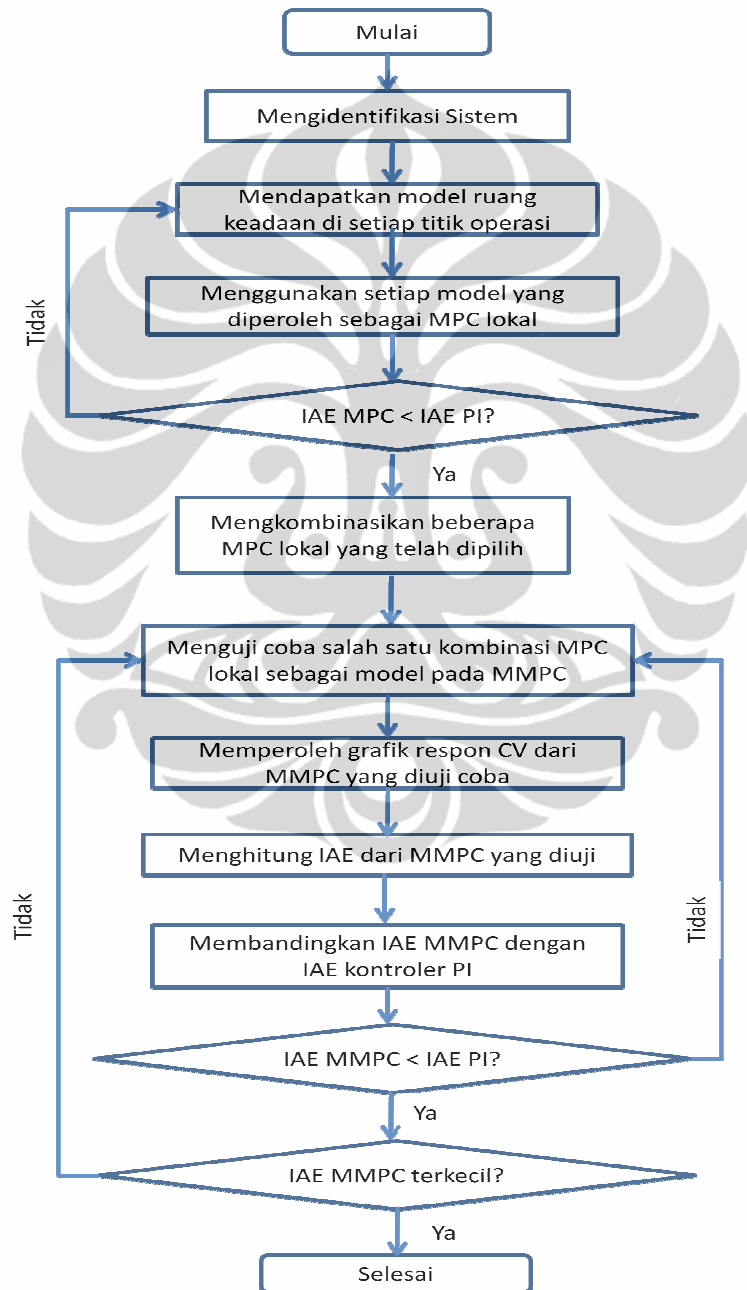
Tujuan kebanyakan kontroler adalah mengatur efek deterministik gangguan pada variabel yang terkendali. MPC standar belum dapat menghasilkan hasil yang baik dalam menekan dampak dari gangguan karena biasanya mengasumsikan bahwa semua sinyal termodelkan masa depan tetap sama dengan kesalahan prediksi saat ini (Zhao, 2004). Oleh karena itu, MPC perlu dibenahi agar dapat mengatasi gangguan dengan lebih efektif.

Penelitian yang akan dilakukan sekarang bersifat melanjutkan penelitian mengenai penggunaan RMPC dalam mendukung penerapan MMPC. Dari beberapa MPC lokal yang dipilih untuk digunakan pada MMPC akan diuji coba apakah MPC lokal tersebut dapat mengatasi gangguan yang terjadi dalam operasi. Keberhasilan penelitian akan diukur berdasarkan besarnya *integral absolute error* (IAE) dari penerapan MMPC berdasarkan RMPC yang dipilih dan kemudian membandingkan besar IAE tersebut dengan IAE dari kontroler konvensional (PI).

### BAB 3 METODOLOGI PENELITIAN

#### 3.1. Alur Penelitian

Alur penelitian yang akan dilakukan ditunjukkan oleh Gambar 3.1 di bawah ini.



Gambar 3.1. Diagram Alir Penelitian

### 3.2. Model dan Perangkat Lunak yang Digunakan

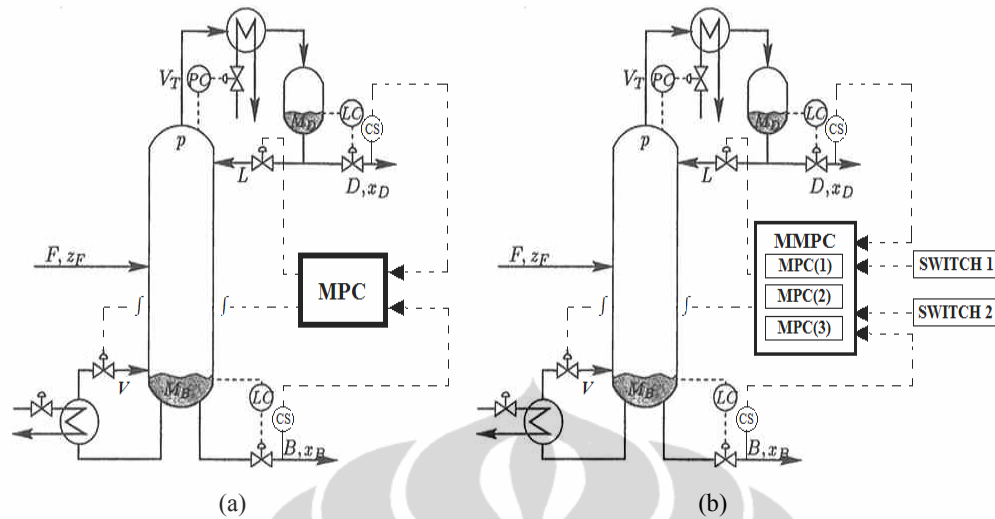
Penelitian ini menggunakan model kolom distilasi, yaitu kolom A yang dikembangkan oleh Skogestad, sebagai sistem yang akan dikendalikan. Kolom A dipilih sebagai model dasar karena merupakan model kolom distilasi biner yang realistis (Skogestad, 1997).

MATLAB<sup>®</sup> adalah bahasa pemrograman tingkat tinggi dan program interaktif yang memungkinkan penghitungan dan penjalanan perintah lebih cepat dibandingkan bahasa pemrograman tradisional seperti C, C++, dan Fortran. Di dalam perangkat lunak MATLAB ini terdapat juga Simulink. Simulink<sup>®</sup> adalah program untuk simulasi multidomain dan *Model-Based Design* untuk sistem dinamik dan mendalam. Dengan Simulink, dapat disimulasikan model kolom distilasi yang akan dikendalikan. Dalam Simulink terdapat berbagai blok yang dapat ditambahkan pada desain sistem sesuai kebutuhan. Pada penelitian ini akan digunakan blok MPC untuk memperoleh model linear MPC lokal, kemudian blok MMPC untuk menguji kombinasi beberapa MPC lokal tersebut.

### 3.3. Prosedur Penelitian

- Variabel Penelitian
  - Variabel bebas: variabel bebas pada penelitian ini adalah variabel input kontrol yaitu *reflux flow* dan *boil up flow (LV-configuration)* serta *feed rate (F)* dan *feed composition (zF)* sebagai gangguan.
  - Variabel terikat: variabel terikat pada penelitian ini adalah variabel yang dikendalikan yaitu fraksi mol produk atas dan fraksi mol produk bawah.

Gambar 3.2. adalah model kolom distilasi sederhana dengan *LV-configuration* dilengkapi dengan pengendali MPC dan MMPC yang terhubung dengan *manipulated variable* dan *controlled variable* pada penelitian ini.



Keterangan:

PC = Pressure Control

LC = Level Control

CS = Composition Sensor

Gambar 3.2. Model Kolom Distilasi Sederhana dengan Pengendali (a) MPC dan (b) MMPC

- Prosedur Percobaan

1. Mengidentifikasi sistem yang akan dikendalikan

Sistem yang akan diidentifikasi adalah kolom distilasi biner “Kolom A”. Sistem ini dikembangkan oleh Skogestad (1997). Asumsi yang digunakan adalah pemisahan biner, tekanan konstan, holdup uap diabaikan, kondenser total, laju molar konstan, kesetimbangan di setiap tingkat dengan volatilitas relatif yang konstan, dan dinamika laju cairan yang dilinearisasi. Data-data yang digunakan sebagai harga awal untuk kolom A adalah sebagai berikut (Skogestad, 1997):

$N = 40$ (jumlah stage teoritis)	$D = 0.5$ (laju produk distilat [kmol/min])
$N_t = 41$ (jumlah stage total)	$L = 2.706$ (reflux flow [kmol/min])
$N_F = 21$ (nomor stage masuk umpan)	$V = 3.206$ (boil up flow [kmol/min])
$F = 1$ (laju feed [kmol/min])	$y_D = 0.99$ (komposisi produk distilat)
$z_F = 0.5$ (komposisi feed)	$x_B = 0.01$ (komposisi produk bawah)
$q_F = 1$ (fraksi liquid di feed)	$M_i = 0.5$ (holdup cairan di stage-i)
$\alpha = 1.5$ (volatilitas relatif)	$\tau_L = 0.06$ (konstantawaktu)

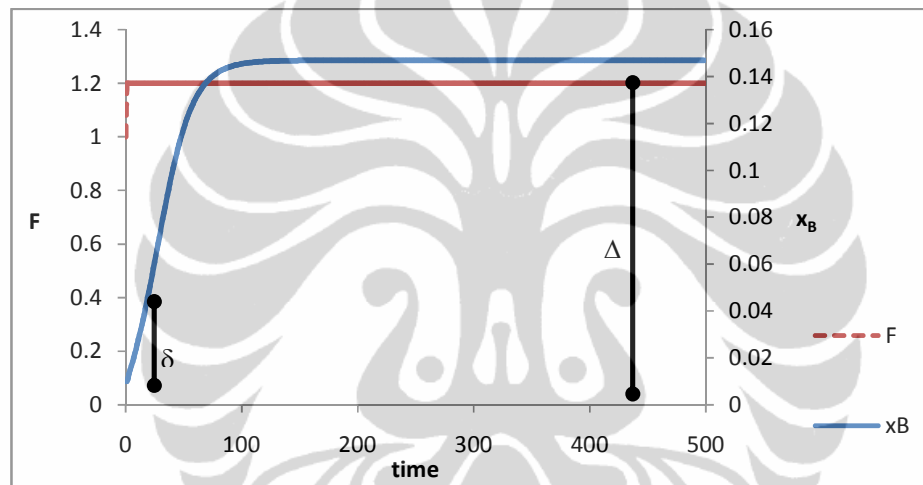
Kemudian model kolom A tersebut dibuat menjadi sebuah file simulink.

Dengan melakukan perubahan terhadap  $F$  dan  $zF$ , akan diperoleh *Process Reaction Curve* dari  $y_D$  dan  $x_B$ . Contoh PRC ditunjukkan oleh Gambar 3.3. Kemudian akan dihitung parameter FOPDT dengan cara berikut:

$$Kp = \frac{\Delta}{\delta} \quad (3.1)$$

$$\tau = 1,5(t_{63\%} - t_{28\%}) \quad (3.2)$$

$$\theta = t_{63\%} - \tau \quad (3.3)$$



Gambar 3.3. *Process Reaction Curve*

## 2. Mendapatkan model di setiap titik dalam rentang operasi

Jenis model yang akan digunakan adalah model ruang keadaan (*state space*), yaitu berupa matriks A, B, C, dan D. Model ruang keadaan akan diperoleh dengan menggunakan syntax `linmod('sys')` dimana 'sys' merupakan nama file simulink yang telah dibuat sebelumnya. Rentang operasi variabel yang dikendalikan yaitu dari 0.01 sampai 0.1 untuk  $x_B$ .

## 3. Menggunakan setiap model yang telah diperoleh sebagai MPC lokal

Model ruang keadaan di setiap titik operasi yang telah diperoleh akan digunakan sebagai MPC lokal. Kontroler MPC dijalankan dengan menggunakan toolbox MPC pada MATLAB.

4. Memilih beberapa MPC lokal dengan penanganan gangguan yang baik

Setelah menjalankan kontroler MPC untuk masing-masing model, maka akan diperoleh grafik respon *controlled variable* yang dapat dihitung besarnya nilai *Integral Absolute Error* (IAE) untuk masing-masing MPC lokal. Nilai IAE ini kemudian dibandingkan dengan nilai IAE dari kontroler konvensional Proporsional-Integral (PI). Dari seluruh MPC lokal yang diperoleh hanya akan dipilih beberapa buah MPC lokal yang memiliki nilai IAE kurang dari IAE PI.

5. Mengaplikasikan beberapa lokal MPC yang telah dipilih untuk digunakan dalam MMPC

Beberapa MPC lokal yang telah dipilih akan dikombinasikan satu sama lain untuk digunakan pada MMPC sehingga akan diuji coba beberapa MMPC yang masing-masingnya dapat terdiri dari dua atau lebih MPC lokal yang dikombinasikan. Untuk menjalankan MMPC, perlu ditambahkan sebuah blok kontroler MMPC pada file simulink.

6. Membandingkan IAE dari setiap MMPC yang dijalankan dengan IAE dari kontroler konvensional (PI)

Untuk masing-masing MMPC yang dijalankan, akan diperoleh grafik respon CV. Dari grafik respon CV ini akan dapat dihitung *integral absolute error*-nya. Sedangkan simulasi pengendalian kolom A dengan kontroler PI dijalankan dengan menggunakan simulink dan Matlab file yang dapat diperoleh dari internet (Skogestad). Dengan menjalankan simulasi untuk kontroler PI, dapat diperoleh juga grafik respon CV yang dapat dihitung IAE-nya dan kemudian dibandingkan dengan IAE dari setiap MMPC yang dijalankan. Satu atau lebih MMPC yang memiliki IAE kurang dari IAE kontroler PI dipilih, dan MMPC yang memiliki IAE terkecil diantara MMPC terpilih lainnya ditetapkan sebagai MMPC (kombinasi MPC lokal) terbaik.



- **Prosedur Pengambilan Sampel**

Sampel yang akan diambil dan dianalisis pada penelitian ini adalah grafik respon CV. Grafik respon CV ini diperoleh setelah menjalankan program MPC, MMPC, dan kontroler PI. Dengan menggunakan grafik respon CV dapat dihitung IAE masing-masing kontroler.

- **Prosedur Analisis**

Penelitian yang dilakukan akan mendapatkan keluaran berupa grafik respon CV untuk setiap MMPC yang diuji coba dimana dalam setiap MMPC terdapat kombinasi antara dua atau lebih MPC lokal yang telah dipilih sebelumnya. Dari grafik respon CV tersebut dapat dihitung besar IAE-nya yang menunjukkan penyimpangan respon variabel yang dikendalikan terhadap perubahan gangguan. Selain grafik respon CV dari MMPC yang diuji coba, juga akan didapatkan grafik respon CV dari kontroler PI sebagai pembanding yang akan dihitung juga besar IAE-nya. Kemudian besar IAE dari MMPC dibandingkan dengan besar IAE dari kontroler PI. Kontroler yang baik akan menghasilkan nilai IAE terkecil.

- **Prosedur Perhitungan**

- **Tuning parameter MPC**

Untuk menjalankan program MPC dan menghitung IAE dari MPC maka sebelumnya perlu ditentukan nilai dari tiga parameter MPC, yaitu  $T$  (waktu sampel),  $P$  (*prediction horizon*), dan  $M$  (*control horizon*). Nilai untuk ketiga parameter tersebut dapat dihitung dengan pendekatan strategi tuning DMC non-adaptif (Dougherty, 2003a) sebagai berikut:

1. Pendekatan dinamika proses output kontroler untuk pasangan-pasangan variabel proses terukur dengan model FOPDT:

$$\frac{y_r(s)}{u_s(s)} = \frac{K_{rs}e^{-\theta_{rs}s}}{\tau_{rs}s+1} \quad (r = 1,2, \dots, R; s = 1,2, \dots, S) \quad (3.4)$$

2. Memilih waktu sampel sedekat mungkin dengan:

$$T_{rs} = \text{Max} (0.1\tau_{rs}, 0.5\theta_{rs}), \quad (r = 1,2, \dots, R; s = 1,2, \dots, S) \quad (3.5)$$

$$T = \text{Min}(T_{rs}) \quad (3.6)$$

3. Menghitung  $P$  (*prediction horizon*):

$$P = \text{Max} \left( \frac{5\tau_{rs}}{T} + k_{rs} \right) \quad (3.7)$$

$$\text{dimana } k_{rs} = \left( \frac{\theta_{rs}}{T} + 1 \right), \quad (r = 1, 2, \dots, R; s = 1, 2, \dots, S) \quad (3.8)$$

4. Menghitung  $M$  (*control horizon*):

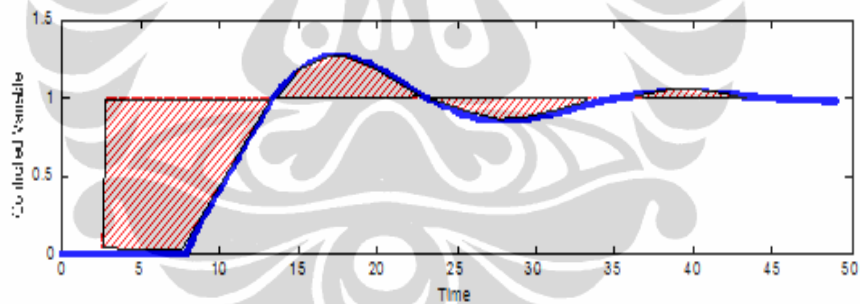
$$M = \text{Max} \left( \frac{\tau_{rs}}{T} + k_{rs} \right), \quad (r = 1, 2, \dots, R; s = 1, 2, \dots, S) \quad (3.9)$$

- Menghitung *Integral Absolute Error*

IAE dihitung dengan:

$$IAE = \int |SP(t) - CV(t)| dt \quad (3.10)$$

Jika dilihat pada grafik respon CV, maka IAE adalah luas daerah absolut dari selisih luas grafik *setpoint* dengan luas grafik respon CV. Semakin kecil IAE berarti CV semakin mendekati *setpoint* nya yang menandakan kontroler yang digunakan semakin baik. Seperti diperlihatkan pada contoh grafik respon CV berikut ini, daerah yang diarsir adalah besar *Integral Absolute Error*.



Gambar 3.4. Grafik Respon CV dengan IAE

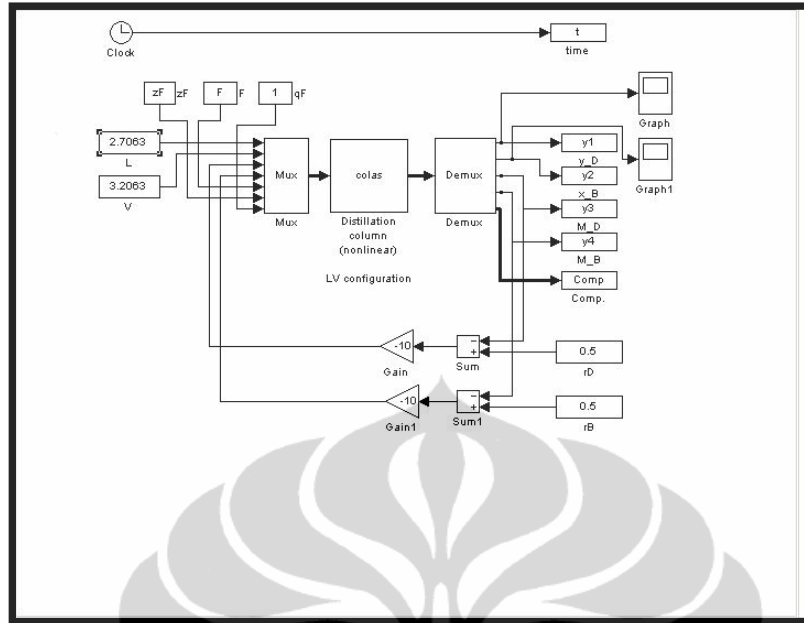
## BAB 4

### HASIL DAN PEMBAHASAN

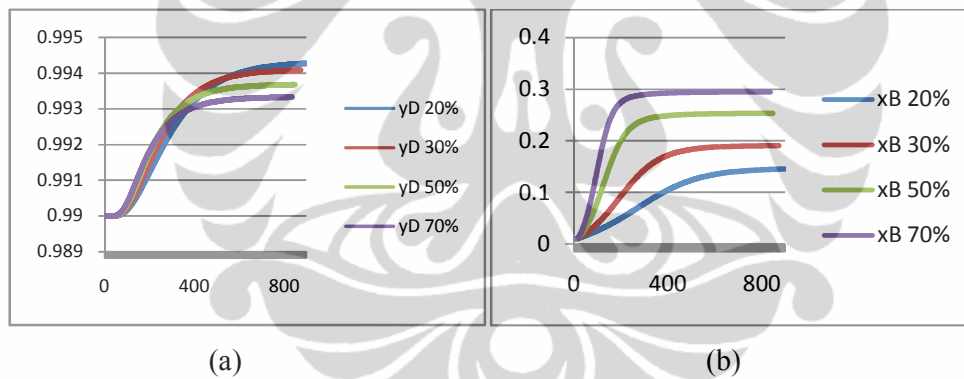
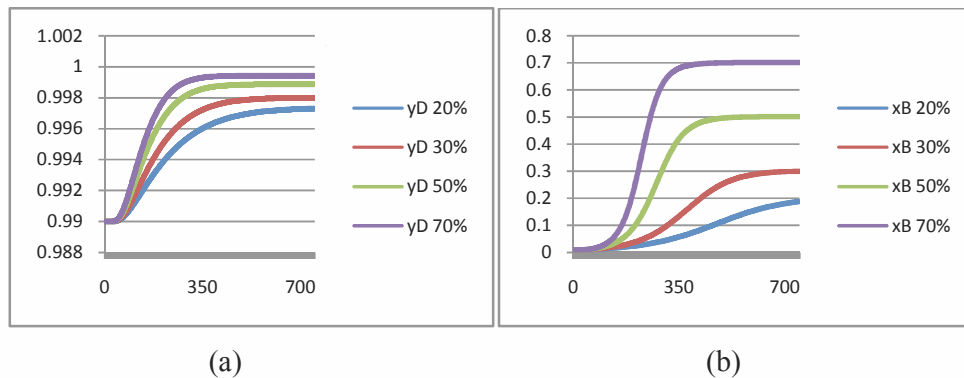
#### 4.1. Identifikasi Sistem

Model kolom A yang digunakan memiliki dua output kontroler dan dua variabel proses terukur. Output kontroler adalah reflux flow (L) dan boil up flow (V) sedangkan variabel proses terukur adalah komposisi produk atas ( $y_D$ ) dan komposisi produk bawah ( $x_B$ ). Dengan melakukan perubahan pada masing-masing output kontroler akan mempengaruhi variabel proses terukur. Perubahan pada masing-masing output kontroler dipengaruhi oleh gangguan berupa perubahan feed rate (F) dan feed composition (zF). Oleh karena itu, gangguan juga mempengaruhi variabel proses terukur. Pengaruh perubahan output kontroler terhadap variabel proses terukur disebut *sub-proses*. Pada percobaan ini terdapat empat buah sub-proses yaitu F terhadap  $y_D$ , F terhadap  $x_B$ , zF terhadap  $y_D$ , dan zF terhadap  $x_B$ .

Perubahan yang dilakukan terhadap output kontroler adalah kenaikan sebesar 20%, 30%, 50%, dan 70% dari harga awalnya. Perubahan dilakukan untuk satu output kontroler sedangkan output kontroler lain tetap (perubahan F sedangkan zF tetap dan perubahan zF sedangkan F tetap). Perubahan ini disebut perubahan aliran eksternal. Sedikit saja perubahan pada aliran eksternal akan berpengaruh besar terhadap perubahan variabel proses terukur. Perubahan variabel proses, baik  $y_D$  maupun  $x_B$ , oleh karena perubahan output kontroler akan menghasilkan kurva reaksi proses (*Process Reaction Curve / PRC*) yang ditunjukkan oleh Gambar 4.2 dan Gambar 4.3. PRC diperoleh dengan menjalankan simulasi dari file Simulink Plant Kolom A seperti yang ditunjukkan oleh Gambar 4.1. Kemudian, file Simulink tersebut dijalankan dengan program m-file yang terdapat pada Lampiran 1 dan Lampiran 2.



Gambar 4.1. Plant Kolom A

Gambar 4.2. PRC pengaruh perubahan F terhadap (a)  $y_D$ , (b)  $x_B$ Gambar 4.3. PRC pengaruh perubahan zF terhadap (a)  $y_D$ , (b)  $x_B$

Berdasarkan PRC yang dihasilkan, sistem diidentifikasi dengan menggunakan persamaan 3.1 sampai 3.3 sehingga diperoleh parameter FOPDT seperti yang ditunjukkan oleh Tabel 4.1.

Tabel 4.1. Parameter FOPDT

	F to y1	F to y2	zF to y1	zF to y2
Process variable values	20 %	20 %	20 %	20 %
FOPDT model fit parameters				
$K_p$	0.0193	0.9435	0.8675	0.0155
$\tau_p$	4.3551	7.1619	4.8206	1.8804
$\theta_p$	1.2341	9.4152	9.7691	1.0350
Process variable values	30 %	30 %	30 %	30 %
FOPDT model fit parameters				
$K_p$	0.0133	0.7838	0.6439	0.0111
$\tau_p$	2.9118	5.2448	3.3639	1.1053
$\theta_p$	1.2506	8.4499	7.3450	1.1103
Process variable values	50 %	50 %	50 %	50 %
FOPDT model fit parameters				
$K_p$	0.0076	0.5332	0.4675	0.074
$\tau_p$	1.9002	3.6407	2.1214	0.6930
$\theta_p$	1.1556	6.5385	5.1503	1.1416
Process variable values	70 %	70 %	70 %	70 %
FOPDT model fit parameters				
$K_p$	0.0056	0.4124	0.3599	0.0045
$\tau_p$	1.5225	2.9019	1.6706	0.5913
$\theta_p$	1.1111	5.6263	3.9917	1.1183

#### 4.2. Performa Kontroler PI

Kontroler PI digunakan untuk mengendalikan plant Kolom A nonlinear dengan perubahan gangguan. Hasilnya akan digunakan sebagai pembanding dalam pemilihan MPC lokal yang akan digunakan pada MMPC. Struktur pengendalian yang digunakan untuk mengendalikan komposisi produk atas dan komposisi produk bawah adalah pengendalian loop tunggal (*single loop control*). Dengan demikian, reflux flow digunakan untuk mengendalikan komposisi produk atas (L mengendalikan  $y_D$ ) dan boil-up flow digunakan untuk mengendalikan komposisi produk bawah (V mengendalikan  $x_B$ ).

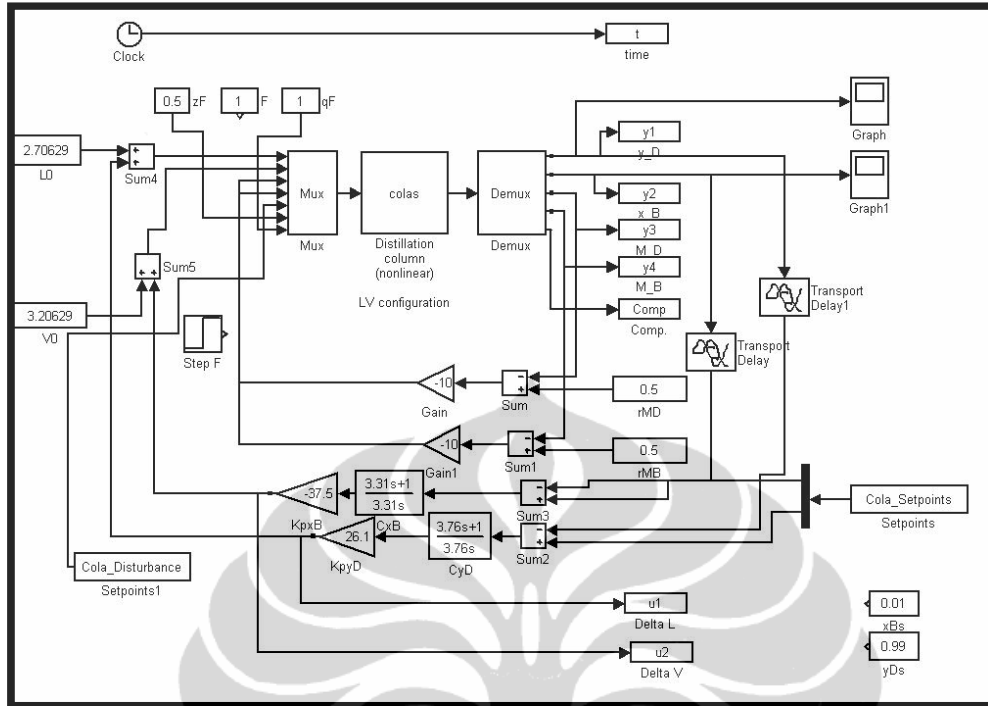
Sesuai dengan struktur pengendalian loop tunggal, maka digunakan dua buah kontroler PI dengan pengaturan parameter-parameternya yang berbeda satu sama lain. Parameter kontroler PI yang digunakan adalah sebagai berikut:

$$k_1(s) = 26,1 \frac{1+3,76s}{3,76s} (y_D - \text{L loop}) \quad (2.15)$$

$$k_2(s) = -37,5 \frac{1+3,31s}{3,31s} (x_B - \text{V loop}) \quad (2.16)$$

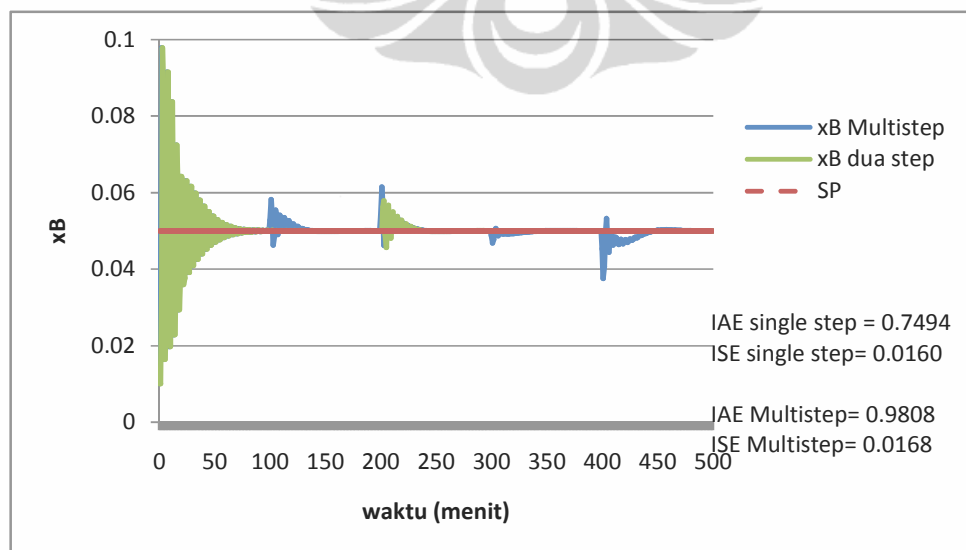
Parameter-parameter tersebut digunakan untuk Kolom A dengan konfigurasi LV sehingga dapat mengendalikan komposisi produk atas dan komposisi produk bawah.

Plant Kolom A nonlinear dengan perubahan gangguan yang dikendalikan dengan kontroler PI ditunjukkan oleh Gambar 4.4. Program untuk menjalankan simulasi pengendalian Kolom A dengan kontroler PI dapat dilihat pada Lampiran 3. Simulasi dijalankan untuk dua skenario yang akan dijelaskan pada bagian 4.3. Hasil simulasi adalah berupa nilai IAE dan ISE yang nantinya akan dibandingkan dengan nilai IAE dan ISE single MPC untuk memilih MPC lokal yang akan digunakan pada MMPC.



Gambar 4.4. Plant Kolom A dengan Kontroler PI

Simulasi kontroler PIDengan *setpoint* 0,05 menghasilkan IAE sebesar 0,7494 dan ISE sebesar 0,0160 untuk skenario 1 dan IAE sebesar 0,9808 dan ISE sebesar 0,0168 untuk skenario 2. Sedangkan pada *setpoint* lebih besardari 0,05 kontroler PI tidak mampu mengendalikan. Gambar 4.5 menunjukkan performa kontroler PI terhadap komposisi produk bawah.



Gambar 4.5. Performa kontroler PI

### 4.3. Performa Single Linear MPC dan Pemilihan MPC Lokal

Untuk mendapatkan MPC lokal yang akan digunakan pada MMPC, terlebih dahulu Kolom A dikendalikan dengan single MPC, yaitu menggunakan satu model lokal untuk seluruh rentang operasi. Model lokal yang diuji berjumlah 10 yang berbeda kondisi awalnya dari 0,01 (MPC1) sampai 0,1 (MPC10) dengan *set point* tetap yaitu 0,05. Dalam hal ini yang dikendalikan secara langsung dengan perubahan gangguan adalah komposisi produk bawah. Untuk komposisi produk atas, kontroler tidak mengaturnya secara langsung. Nilai  $y_D$  akan bervariasi seperlunya untuk mencapai  $x_B$  yang sesuai *setpoint*. Akan tetapi nilai  $y_D$  akan tetap diukur dan memengaruhi hasil pengendalian. Pengaturan seperti ini akan konsisten digunakan dalam setiap pengujian sampai akhir.

Perubahan *step* gangguan akan mensimulasikan gangguan yang dapat memengaruhi kerja kolom distilasi. MPC akan digunakan untuk menangani gangguan tersebut agar komposisi produk dapat kembali ke *set point* yang diminta. Untuk skenario 1 perubahan gangguan yang diuji untuk CV, dalam hal ini  $x_B$ , ada satu *step* yaitu, 1,0 ke 1,2 pada *set point* 0,05 dimana pada rentang operasi ini, kontroler PI masih dapat digunakan walaupun memberikan hasil pengendalian yang tidak stabil. Pemilihan MPC lokal yang akan digunakan sebagai model pada MMPC didasarkan pada nilai IAE MPC yang kurang dari nilai IAE PI dengan nilai 0,7494.

Perubahan gangguan untuk skenario 2 adalah dengan *multistep* yaitu, 1,0; 1,2; 1,5; 1,4; 1,0 dengan *set point* yang sama yaitu 0,05 untuk mensimulasikan gangguan yang lebih variatif dan banyak dimana pada skenario ini sistem akan diuji kemampuannya untuk dapat mengembalikan komposisi produk ke *set point* yang ditentukan. Pemilihan MPC lokal yang akan digunakan sebagai model pada MMPC didasarkan pada nilai IAE MPC yang kurang dari nilai IAE PI dengan nilai 0,9808.

Single MPC yang digunakan untuk mengendalikan Kolom A menggunakan parameter-parameter MPC sesuai dengan Tabel 4.2 ditambah dengan satu *set* parameter tambahan yang ternyata memberikan nilai IAE yang lebih baik daripada hasil tuning DMC non-adaptif, yaitu  $T_s=0,1$ ;  $P=12$ ;

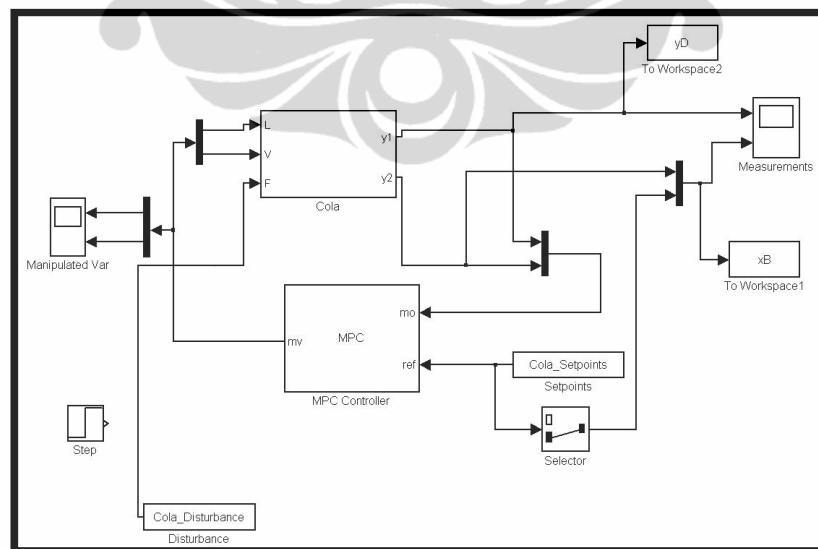


$M=2$ . Dengan parameter MPC ini, 9 dari 10 model memberikan hasil IAE yang lebih baik dibandingkan dengan parameter MPC yang lain, kecuali untuk model MPC1 yang memberikan hasil lebih buruk. *Set* parameter MPC tambahan inilah yang akan digunakan juga nantinya dalam pengujian menggunakan MMPC. Hasil lebih baik yang diberikan oleh penggunaan parameter MPC optimum ini menunjukkan bahwa persamaan tuning parameter MPC untuk DMC non-adaptif kurang cocok digunakan untuk plant nonlinear dan memungkinkan untuk adanya pengembangan persamaan sehingga diperoleh parameter MPC yang optimum untuk plant yang nonlinear.

Tabel 4.2. Parameter MPC

Tuning Parameter	Process variable values			
	20 %	30 %	50 %	70 %
Ts	0.72	0.53	0.36	0.29
P	66	67	69	70
M	26	27	29	30

Plant Kolom A nonlinear yang dikendalikan oleh kontroler single MPC ditunjukkan oleh Gambar 4.6 yang selanjutnya file Simulink ini akan dipanggil oleh program pada m-file yang sesuai dengan skenarionya masing-masing.



Gambar 4.6. Plant Kolom A nonlinear yang dikendalikan oleh kontroler single MPC

#### 4.3.1. Performa Single MPC dan Pemilihan MPC Lokal pada Skenario 1

Pada skenario 1, performa single linear MPC dapat dibandingkan dengan performa kontroler PI. Simulasi juga dilakukan untuk setiap nilai parameter MPC hasil tuning pada bagian 4.1 dan juga untuk parameter optimum ( $T_s = 0,1$ ;  $P = 12$ ;  $M = 2$ ). Program untuk menjalankan simulasi single MPC skenario 1 terdapat pada Lampiran 4. Program yang terdapat pada Lampiran 4 adalah untuk menjalankan MPC2 pada parameter MPC  $T_s = 0,1$ ;  $P = 12$ ;  $M = 2$ . Kondisi awal  $x_B$  dapat diubah-ubah dari 0,01 sampai 0,1 untuk mendapatkan model lainnya (MPC1 sampai MPC10). Untuk menguji parameter MPC lainnya juga dengan mengubah  $T_s$ ,  $P$ , dan  $M$  pada m-file.

Setelah menjalankan simulasi untuk setiapset parameter MPC dan juga untuk sepuluh model (MPC1 sampai MPC10) maka akan diperoleh nilai IAE dan ISE untuk masing-masing single linear MPC yang ditunjukkan oleh Tabel 4.3 dan Tabel 4.4. Grafik respon  $x_B$  hasil simulasi skenario 1, untuk setiap kondisi awal dan setiapset parameter MPC, dapat dilihat pada Lampiran 5 sampai Lampiran 9.

Tabel 4.3. Integral Absolute Error dari Single MPC pada Skenario 1

Kontroler	$T_s = 0.72$	$T_s = 0.53$	$T_s = 0.36$	$T_s = 0.29$	$T_s = 0.1$
	$P = 66$	$P = 67$	$P = 69$	$P = 70$	$P = 12$
	$M = 26$	$M = 27$	$M = 29$	$M = 30$	$M = 2$
MPC1	0.9310	0.9781	0.8728	0.8031	1.5175
MPC2	0.3766	0.3753	0.3249	0.2950	0.2564
MPC3	0.4391	0.4277	0.4010	0.3851	0.3501
MPC4	0.5218	0.5162	0.4945	0.4810	0.4517
MPC5	0.5857	0.5840	0.5646	0.5530	0.5312
MPC6	0.6347	0.6374	0.6224	0.6118	0.5912
MPC7	0.6731	0.6804	0.6699	0.6611	0.6400
MPC8	0.7060	0.7166	0.7102	0.7033	0.6812
MPC9	0.7340	0.7473	0.7453	0.7400	0.7147
MPC10	0.7585	0.7740	0.7763	0.7728	0.7427

Tabel 4.4. Integral Square Error dari Single MPC pada Skenario 1

Kontroler	Ts = 0.72 P = 66 M = 26	Ts = 0.53 P = 67 M = 27	Ts = 0.36 P = 69 M = 29	Ts = 0.29 P = 70 M = 30	Ts = 0.1 P = 12 M = 2
MPC1	0.0188	0.0275	0.0278	0.0263	0.0755
MPC2	0.0057	0.0063	0.0053	0.0046	0.0036
MPC3	0.0075	0.0076	0.0068	0.0062	0.0052
MPC4	0.0095	0.0098	0.0090	0.0085	0.0076
MPC5	0.0113	0.0118	0.0111	0.0106	0.0098
MPC6	0.0130	0.0136	0.0130	0.0125	0.0118
MPC7	0.0145	0.0153	0.0147	0.0142	0.0136
MPC8	0.0159	0.0168	0.0163	0.0159	0.0153
MPC9	0.0172	0.0183	0.0178	0.0174	0.0170
MPC10	0.0184	0.0196	0.0192	0.0188	0.0185

Berdasarkan nilai IAE MPC yang kurang dari IAE PI, MPC2 hingga MPC10 telah memenuhi kriteria yang diinginkan. Namun, dalam penelitian ini hanya dipilih MPC2, MPC3, MPC4, MPC5, MPC6 untuk dikombinasikan sebagai model pada MMPC skenario 1 karena nilai IAE MPC2 hingga MPC6 merupakan yang terkecil diantara sembilan MPC dengan nilai IAE kurang dari IAE PI.

#### 4.3.2. Performa Single MPC dan Pemilihan MPC Lokal pada Skenario 2

Pada skenario 2, performa single linear MPC dapat dibandingkan dengan performa kontroler PI. Simulasi juga dilakukan untuk setiap nilai parameter MPC hasil tuning pada bagian 4.1 dan juga untuk parameter optimum ( $T_s = 0,1$ ;  $P = 12$ ;  $M = 2$ ). Program untuk menjalankan simulasi single MPC skenario 2 terdapat pada Lampiran 10. Program yang terdapat pada Lampiran 10 adalah untuk menjalankan MPC2 pada parameter MPC  $T_s = 0,1$ ;  $P = 12$ ;  $M = 2$ . Kondisi awal  $x_B$  dapat diubah dari 0,01 sampai 0,1 untuk mendapatkan model lainnya (MPC1

sampai MPC10). Untuk menguji parameter MPC lainnya juga dengan mengubah  $T_s$ ,  $P$ , dan  $M$  pada m-file.

Setelah menjalankan simulasi untuk setiapset parameter MPC dan juga untuk sepuluh model (MPC1 sampai MPC10) maka akan diperoleh besar IAE dan ISE untuk masing-masing single linear MPC yang ditunjukkan oleh Tabel 4.5 dan Tabel 4.6. Grafik respon  $x_B$  hasil simulasi skenario 2, untuk setiap kondisi awal dan setiapset parameter MPC, dapat dilihat pada Lampiran 11 sampai Lampiran 15.

Tabel 4.5. Integral Absolute Error dari Single MPC pada Skenario 2

Kontroler	$T_s = 0.72$	$T_s = 0.53$	$T_s = 0.36$	$T_s = 0.29$	$T_s = 0.1$
	$P = 66$	$P = 67$	$P = 69$	$P = 70$	$P = 12$
	$M = 26$	$M = 27$	$M = 29$	$M = 30$	$M = 2$
MPC1	1.7899	1.7666	1.6111	1.5191	2.1758
MPC2	1.0792	1.0354	0.9464	0.8995	0.8016
MPC3	1.0899	1.0431	0.9831	0.9519	0.8568
MPC4	1.1903	1.1560	1.1034	1.0752	0.9853
MPC5	1.2818	1.2529	1.2026	1.1762	1.0919
MPC6	1.3537	1.3291	1.2845	1.2595	1.1735
MPC7	1.4087	1.3902	1.3505	1.3273	1.2402
MPC8	1.4551	1.4400	1.4057	1.3843	1.2954
MPC9	1.4905	1.4791	1.4518	1.4342	1.3386
MPC10	1.5201	1.5126	1.4932	1.4787	1.3745

Tabel 4.6. Integral Square Error dari Single MPC pada Skenario 2

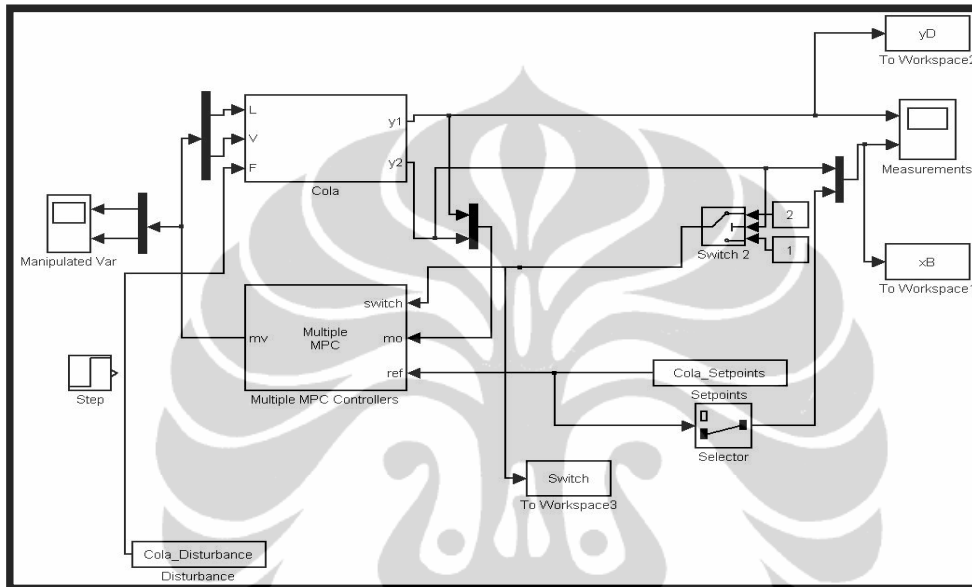
Kontroler	Ts = 0.72	Ts = 0.53	Ts = 0.36	Ts = 0.29	Ts = 0.1
	P = 66	P = 67	P = 69	P = 70	P = 12
	M = 26	M = 27	M = 29	M = 30	M = 2
MPC1	0.0313	0.0381	0.0367	0.0345	0.0819
MPC2	0.0162	0.0156	0.0135	0.0122	0.0094
MPC3	0.0179	0.0170	0.0152	0.0142	0.0115
MPC4	0.0201	0.0195	0.0178	0.0169	0.0142
MPC5	0.0222	0.0218	0.0202	0.0193	0.0168
MPC6	0.0241	0.0239	0.0224	0.0215	0.0191
MPC7	0.0258	0.0258	0.0243	0.0234	0.0212
MPC8	0.0274	0.0275	0.0261	0.0252	0.0231
MPC9	0.0287	0.0290	0.0277	0.0269	0.0249
MPC10	0.0300	0.0304	0.0292	0.0284	0.0265

Berdasarkan nilai IAE MPC yang kurang dari IAE PI, hanya dua model MPC yang memenuhi syarat yaitu MPC2 dan MPC3. Kedua model ini akan dikombinasikan sebagai model pada MMPC.

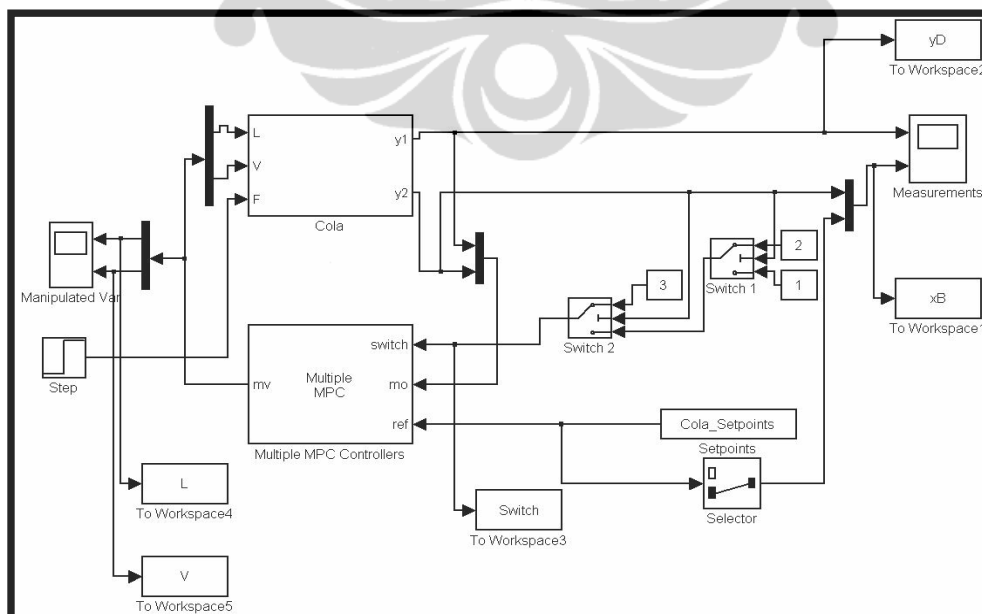
#### 4.4. Performa MMPC Berdasarkan Metode RMPC

Model yang digunakan untuk MMPC adalah MPC lokal (RMPC) yang disarankan pada bagian sebelumnya. Di sini juga akan diuji dua skenario seperti pada percobaan single MPC. Untuk setiap skenario ada lima MPC lokal namun pada MMPC hanya akan digunakan kombinasi dua atau tiga buah MPC lokal. Hal ini karena semakin banyak model yang digunakan pada MMPC tidak menjamin bahwa respon akan semakin baik karena setiap model pasti memiliki error sehingga semakin banyak model yang digunakan akan semakin besar pula error yang dihasilkan. Selain itu, implementasi MMPC dengan terlalu banyak model yang digunakan tidak aplikatif pada dunia praktis karena akan memerlukan lebih banyak perhatian khusus untuk menangani model-model tersebut.

Model yang telah dipilih sebelumnya akan dikombinasikan satu sama lain membentuk kumpulan model yang terdiri dari dua atau tiga buah model. Penyusunan kontroler MMPC yang terdiri dari dua model pada Plant Kolom A ditunjukkan oleh Gambar 4.7, sedangkan Plant Kolom A dengan MMPC yang terdiri dari tiga model ditunjukkan oleh Gambar 4.8.



Gambar 4.7 Plant Kolom A dengan MMPC Dua Model



Gambar 4.8 Plant Kolom A dengan MMPC Tiga Model

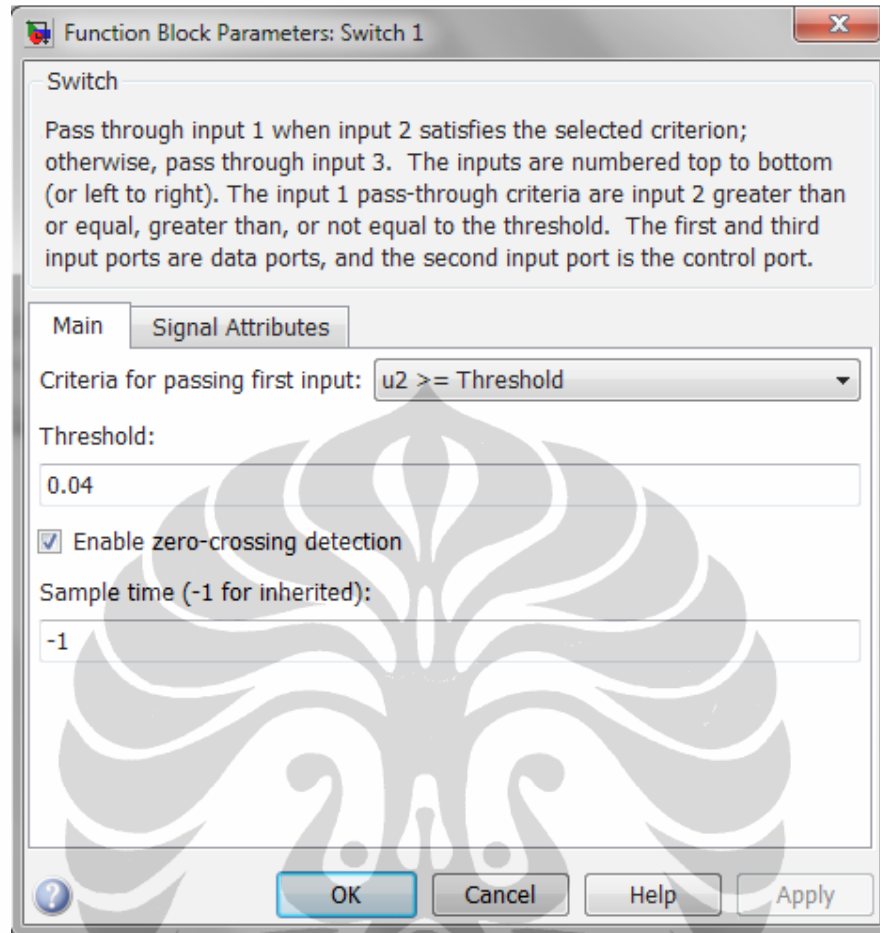
#### 4.4.1. Pengaruh Spesifikasi *Switch*

Penggunaan MMPC memerlukan perhatian khusus pada spesifikasi *switch* karena perbedaan spesifikasi *switch* akan berpengaruh pada performa kontroler walaupun untuk kombinasi model yang sama. Spesifikasi *switch* ditunjukkan oleh blok *switch* pada file Simulink seperti yang terdapat pada Gambar 4.7 dan 4.8.

Spesifikasi *switch* ini menunjukkan transisi atau perubahan model yang digunakan pada nilai CV terukur. Untuk MMPC yang terdiri dari dua model, terdapat satu blok *switch* sedangkan untuk MMPC yang terdiri dari tiga model, terdapat dua blok *switch*.

Pada blok *switch* dispesifikasikan batasan penggunaan satu model dan model lainnya. Sebagai contoh, apabila ditentukan spesifikasi seperti pada Gambar 4.9 artinya model kedua akan digunakan apabila CV terukur  $\geq 0,04$ . Jika terdapat dua blok *switch*, *switch* pertama mempunyai batasan yang lebih kecil daripada batasan *switch* kedua karena model dijalankan sesuai urutan dari kondisi awal terkecil ke kondisi awal terbesar. Model dengan kondisi awal yang kecil akan bekerja dengan baik pada rentang CV terukur yang kecil pula.

Perbedaan hasil respon akan dijelaskan pada bagian selanjutnya. Adanya perbedaan ini menunjukkan pentingnya menggunakan model yang tepat pada saat yang tepat pula. Model yang digunakan pada rentang operasi yang tepat tentu akan membuat performa pengendalian menjadi lebih baik karena variabel proses dapat menjangkau *set point* dengan lebih cepat. Pengendalian yang baik tentunya akan membuat kinerja kolom distilasi menjadi optimal sehingga dapat mengkonservasi energi.



Gambar 4.9. Spesifikasi Blok *Switch* pada Simulink

#### 4.4.2. Performa MMPC pada Skenario 1

MPC lokal yang dipilih untuk digunakan pada MMPC adalah MPC2, MPC3, MPC4, MPC5, dan MPC6. Kelima model ini dikombinasikan satu sama lain membentuk suatu *set* model yang terdiri dari dua atau tiga buah MPC lokal. Kombinasi yang dihasilkan oleh kelima model terpilih ditunjukkan oleh Tabel 4.7.



Tabel 4.7. Kombinasi Model untuk Skenario 1

Kombinasi 2 Model		Kombinasi 3 Model	
Model	Simbol	Model	Simbol
MPC2, MPC3	MPC2.3	MPC2, MPC3, MPC4	MPC2.3.4
MPC2, MPC4	MPC2.4	MPC2, MPC3, MPC5	MPC2.3.5
MPC2, MPC5	MPC2.5	MPC2, MPC3, MPC6	MPC2.3.6
MPC2, MPC6	MPC2.6	MPC2, MPC4, MPC5	MPC2.4.5
MPC3, MPC4	MPC3.4	MPC2, MPC4, MPC6	MPC2.4.6
MPC3, MPC5	MPC3.5	MPC2, MPC5, MPC6	MPC2.5.6
MPC3, MPC6	MPC3.6	MPC3, MPC4, MPC5	MPC3.4.5
MPC4, MPC5	MPC4.5	MPC3, MPC4, MPC6	MPC3.4.6
MPC4, MPC6	MPC4.6	MPC3, MPC5, MPC6	MPC3.5.6
MPC5, MPC6	MPC5.6	MPC4, MPC5, MPC6	MPC4.5.6

File Simulink seperti pada Gambar 4.7 dan 4.8 kemudian dijalankan dengan program m-file. Untuk MMPC dengan dua model, program m-file yang dijalankan adalah sebagaimana yang terdapat pada Lampiran 16. Program m-file tersebut dapat digunakan untuk mensimulasikan MMPC untuk seluruh kombinasi dua model pada skenario 1. Contoh program pada Lampiran 16 adalah untuk MPC2.3, untuk kombinasi lainnya dapat disimulasikan dengan mengubah kondisi awal pada *yp*. Sedangkan untuk simulasi MMPC kombinasi tiga model pada skenario 1, program m-file yang digunakan terdapat pada Lampiran 17. Kedua program tersebut dijalankan untuk seluruh kombinasi model yang dipilih untuk skenario 1.

Simulasi MMPC dilakukan untuk semua kombinasi model, baik kombinasi dua model maupun kombinasi tiga model, dan juga untuk variasi spesifikasi *switch*. MMPC yang terdiri dari kombinasi dua model hanya diperlukan satu spesifikasi *switch*. Variasi spesifikasi *switch* yang diuji pada MMPC kombinasi dua model skenario 1 adalah dari 0,01 sampai 0,06. Nilai IAE dan ISE untuk seluruh simulasi MMPC dua model pada skenario 1 terdapat pada Lampiran 18. Tabel 4.8 di bawah ini

merangkum setiap kontroler MMPC dua model pada skenario 1 beserta spesifikasi *switch* yang menghasilkan nilai IAE dan ISE terbaik.

Tabel 4.8. IAE dan ISE Terbaik Setiap MMPC Dua Model pada Skenario 1

Kontroler	<i>Switch</i>	IAE	ISE
MPC2.3	0.05	0.2582	0.0036
MPC2.4	0.06	0.2606	0.0037
MPC2.5	0.06	0.2624	0.0038
MPC2.6	0.06	0.2639	0.0038
MPC3.4	0.06	0.3524	0.0053
MPC3.5	0.06	0.3543	0.0053
MPC3.6	0.06	0.3559	0.0054
MPC4.5	0.06	0.4624	0.0078
MPC4.6	0.06	0.4742	0.0082
MPC5.6	0.05	0.5433	0.0102

Dari tabel lengkap yang terdapat pada Lampiran 18, dapat dilihat bahwa untuk kontroler yang sama akan menghasilkan IAE dan ISE yang berbeda pada spesifikasi *switch* yang berbeda. Sedangkan dari Tabel 4.8 di atas, kontroler yang memberi hasil terbaik adalah MPC2.3 pada *switch* 0,05 yaitu dengan IAE 0,2582 dan ISE 0,0036 kemudian disusul oleh MPC2.4 pada *switch* 0,06 yang menghasilkan IAE 0,2606 dan ISE 0,0037 dan MPC2.5 pada *switch* 0,06 yang menghasilkan IAE 0,2624 dan ISE 0,0038. Semua MMPC dua model ini belum bisa mengalahkan hasil IAE dari single MPCterbaik, yaitu MPC2 dengan IAE 0,2564. Grafik respon  $x_B$  pada MMPC dua model skenario 1 terdapat pada Lampiran 20.

Berbeda dengan MMPC dua model, MMPC tiga model memerlukan dua spesifikasi *switch*. Variasi spesifikasi *switch* dilakukan dari 0,01 sampai 0,06. Hasil simulasi MMPC tiga model untuk seluruh variasi *switch* pada skenario 1 terdapat pada Lampiran 19. Dari hasil tersebut dapat dilihat bahwa variasi kedua spesifikasi *switch* tetap

memberikan pengaruh walaupun tidak terlalu signifikan terhadap IAE dan ISE. Oleh karena itu pada Tabel 4.9 berikut ini ditampilkan kedua spesifikasi *switch* yang digunakan untuk setiap kontroler yang menghasilkan IAE dan ISE terbaik. Dari Tabel 4.9 dapat dilihat bahwa hasil terbaik diberikan oleh kontroler MPC2.3.4 dengan IAE 0,2605 dan ISE 0,0037. Hasil ini diperoleh pada spesifikasi *switch* 1 0,05 dan spesifikasi *switch* 2 0,06. Artinya, ketika CV terukur bernilai  $< 0,05$ , sistem dikendalikan dengan MPC2, ketika  $0,05 \leq CV < 0,06$ , sistem dikendalikan dengan MPC3, dan ketika  $CV \geq 0,06$ , sistem dikendalikan dengan MPC4. Hasil terbaik selanjutnya diberikan oleh kontroler MPC2.3.5, MPC2.3.6, dan MPC2.5.6. Namun, nilai IAE lima MMPC tiga model ini belum dapat mengalahkan nilai IAE terbaik yang dapat dicapai oleh MMPC dua model. Grafik respon  $x_B$  pada MMPC tiga model skenario 1 terdapat pada Lampiran 21.

Tabel 4.9. IAE dan ISE Terbaik Setiap MMPC Tiga Model pada Skenario 1

Kontroler	Switch 1	Switch 2	IAE	ISE
MPC2.3.4	0.05	0.06	0.2605	0.0037
MPC2.3.5	0.05	0.06	0.2624	0.0038
MPC2.3.6	0.05	0.06	0.2639	0.0038
MPC2.4.5	0.05	0.06	0.2911	0.0040
MPC2.4.6	0.05	0.06	0.2927	0.0040
MPC2.5.6	0.05	0.06	0.2639	0.0038
MPC3.4.5	0.05	0.06	0.3735	0.0056
MPC3.4.6	0.05	0.06	0.3751	0.0057
MPC3.5.6	0.05	0.06	0.4019	0.0062
MPC4.5.6	0.05	0.06	0.4797	0.0083

#### 4.4.3. Performa MMPC pada Skenario 2

MPC lokal yang dipilih untuk digunakan pada MMPC hanya MPC2 dan MPC3. Hanya ada satu kombinasi yang mungkin untuk performa MMPC pada skenario 2. Kombinasi kedua model terpilih ditunjukkan oleh Tabel 4.10.

Tabel 4.10. Kombinasi Model untuk Skenario 2

Kombinasi 2 model	
Model	Simbol
MPC2, MPC3	MPC2.3

File Simulink seperti pada Gambar 4.7 kemudian dijalankan dengan program m-file. Untuk MMPC dengan dua model, program m-file yang dijalankan adalah sebagaimana yang terdapat pada Lampiran 22. Program m-file tersebut dapat digunakan untuk mensimulasikan MMPC untuk kombinasi dua model pada skenario 2.

Simulasi MMPC dilakukan untuk kombinasi dua model dengan variasi spesifikasi *switch*. MMPC yang terdiri dari kombinasi dua model hanya diperlukan satu spesifikasi *switch*. Variasi spesifikasi *switch* yang diuji pada MMPC kombinasi dua model skenario 2 adalah dari 0,01 sampai 0,06. Nilai IAE dan ISE untuk simulasi MMPC dua model pada skenario 2 terdapat pada Lampiran 23. Tabel 4.11 di bawah ini adalah performa kontroler MMPC dua model pada skenario 2 beserta spesifikasi *switch* yang menghasilkan nilai IAE dan ISE terbaik.

Tabel 4.11. IAE dan ISE Terbaik MMPC Dua Model pada Skenario 2

Kontroler	<i>switch</i>	IAE	ISE
MPC2.3	0.04	0.7730	0.0096

Dapat dilihat dari tabel 4.11 bahwa performa MPC2.3 untuk mengatasi gangguan multi step sangat baik. MPC2.3 dapat mengalahkan

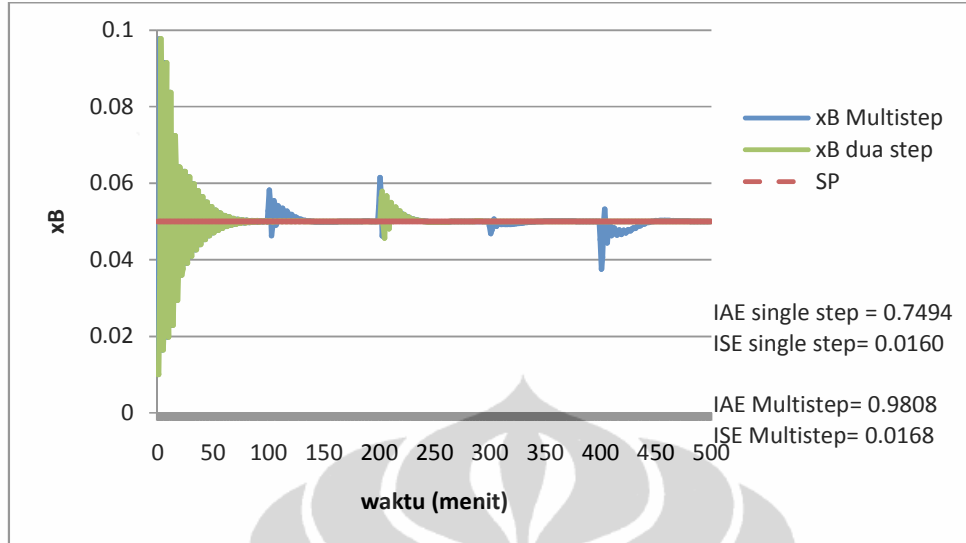
performa single MPC terbaik untuk skenario 2 yaitu MPC2 dengan nilai IAE sebesar 0,8016.

Dari tabel lengkap yang terdapat pada Lampiran 23, dapat dilihat bahwa untuk kontroler yang sama akan menghasilkan IAE dan ISE yang berbeda pada spesifikasi *switch* yang berbeda. Grafik respon  $x_B$  pada MMPC dua model skenario 2 terdapat pada Lampiran 24.

#### 4.5. Diskusi

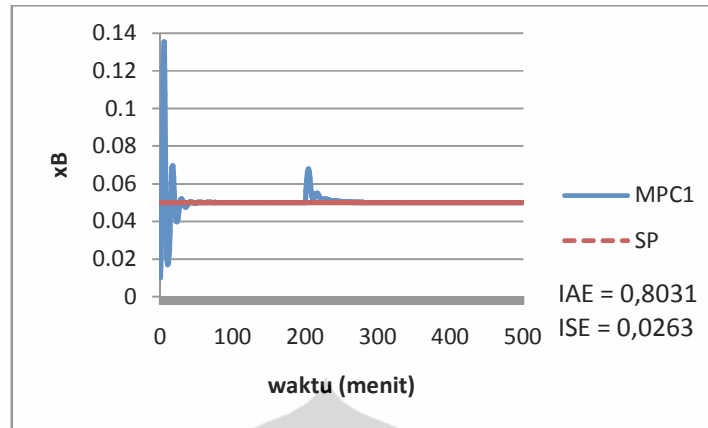
Simulasi yang telah dilakukan adalah mengatasi gangguan plant Kolom A yang nonlinear dengan tiga macam kontroler lalu membandingkan hasil yang diperoleh. Tiga macam kontroler tersebut adalah kontroler PI, single MPC, dan MMPC. Hasil respon dari komposisi produk bawah yang diberi gangguan telah ditunjukkan pada bagian 4.2 sampai 4.4.

Simulasi dengan kontroler PI menggunakan parameter tuning yang sama untuk kedua skenario. Parameter yang digunakan adalah seperti yang ditunjukkan oleh persamaan 2.15 dan 2.16. Dengan menggunakan parameter tersebut, *set point* yang digunakan adalah 0,05 karena pada *set point* tersebut kontroler PI sudah mencapai batas pengendalian. Hal ini dapat dilihat dari performa kontroler PI untuk menangani gangguan pada gambar 4.5. Kontroler PI tidak dapat mengendalikan komposisi produk bawah dengan baik pada *setpoint* 0,05. Walaupun dengan waktu yang lebih panjang, respon akan membaik dan amplitudo osilasi terus menurun mencapai kestabilan. Pemberian gangguan pada skenario 1 tidak begitu mengganggu performa kontroler PI dalam mengendalikan komposisi produk bawah. Hal ini dapat dilihat dengan kembalinya komposisi bawah ke *set point* yang ditentukan. Pemberian gangguan yang bervariasi untuk simulasi skenario 2 juga membuktikan kontroler PI masih mampu untuk mengembalikan komposisi produk bawah ke *set point* yang ditentukan. Hal ini menunjukkan bahwa untuk *set point* yang tidak melebihi batas pengendalian kontroler PI, kontroler PI menunjukkan hasil yang cukup baik dalam menangani gangguan pada operasi kolom distilasi kolom A.



Gambar 4.5. Performa kontroler PI

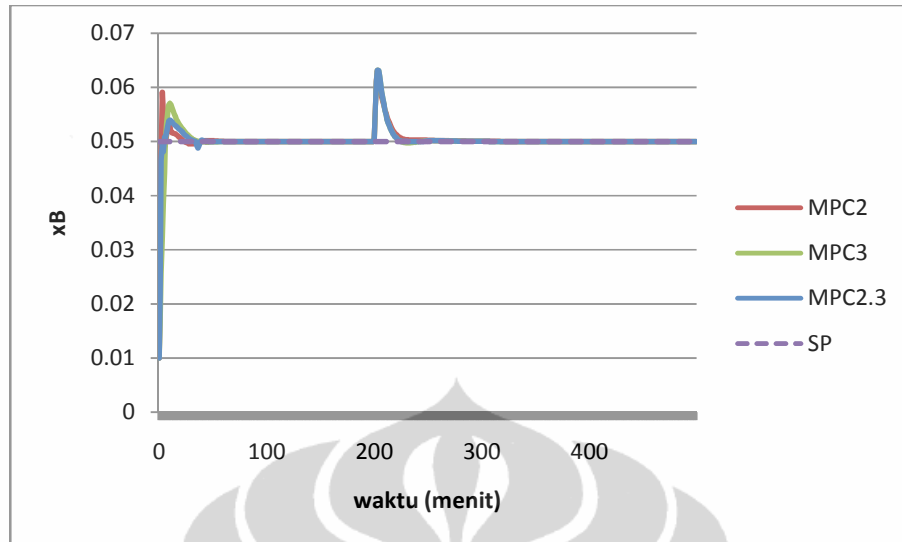
Berbeda dengan kontroler PI yang bekerja dengan mengolah informasi umpan balik berupa error yang dihasilkan, MPC bekerja dengan melakukan prediksi atas output di waktu mendatang. Pada dasarnya MPC akan memberikan respon yang lebih cepat untuk mencapai *setpoint* tergantung dari titik awal operasi MPC tersebut. Gambar 4.10 menunjukkan respon MPC1 pada  $T_s = 0,29$ ;  $P = 70$ ;  $M = 30$  dengan skema perubahan gangguan skenario 1. Respon yang diberikan oleh MPC1 ini adalah contoh ketidakmampuan single MPC dalam menjangkau perubahan gangguan pada *setpoint* yang jauh dari kondisi awalnya yang adalah 0,01. Namun setelah sampai pada *set point* yang diminta, MPC1 memberikan respon yang baik dalam menangani gangguan. Hal ini dapat dilihat dari performa MPC1 di  $t=200$ , dimana MPC1 diberi gangguan sebesar 20% lebih dari nilai awal. MPC1 berhasil menangani gangguan dan mengembalikan komposisi produk bawah ke *set point* yang diminta. Jadi, kontroler single MPC terbukti dapat digunakan untuk menangani gangguan, hanya saja terdapat sedikit kesulitan untuk menjangkau *setpoint* operasi yang terlalu jauh dari kondisi awalnya.



Gambar 4.10. Performa MPC1 pada  $T_s = 0,29$ ;  $P = 70$ ;  $M = 30$ ;  
Skenario 1

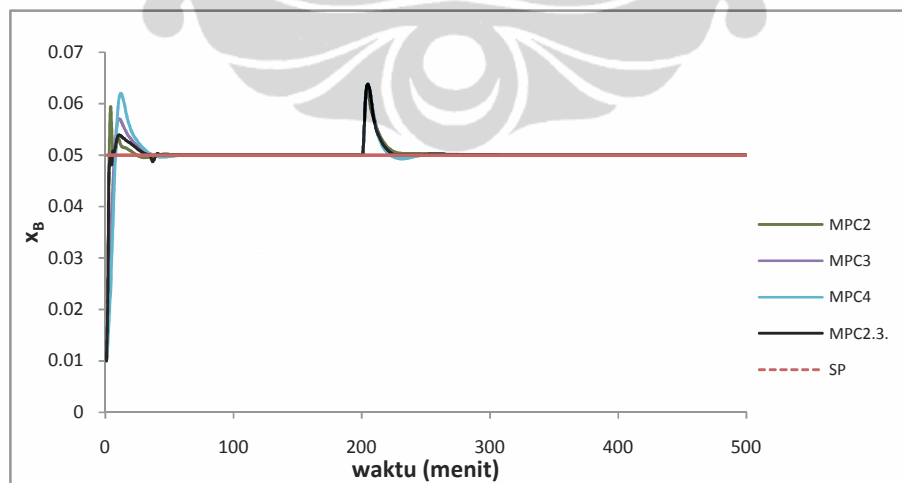
Penjelasan di atas membuktikan bahwa single MPC efektif dalam menangani gangguan dan mengembalikan komposisi produk bawah ke *set point* yang diminta. Hanya saja terdapat kesulitan untuk mencapai *set point* yang jauh dari kondisi awal MPC tersebut. Oleh karena itu, menggabungkan beberapa MPC menjadi MMPC merupakan pilihan yang tepat. Secara teori, kelebihan MMPC dibandingkan dengan single MPC adalah kemampuannya untuk mengendalikan sistem pada *set point* yang diminta dari berbagai kondisi awal untuk menangani perubahan gangguan. Hal ini karena MMPC yang terdiri dari beberapa model yang dijalankan berasal dari kondisi awal yang berbeda, sehingga dapat menjangkau *set point* yang diminta dengan lebih cepat. Model tersebut dijalankan bergantian sesuai dengan pengaturan *switch* yang diberikan.

Gambar 4.11 menunjukkan karakteristik MPC2.3 pada skenario 1 yang merupakan gabungan dari karakteristik single MPC2 dan MPC3 yang dijalankan sesuai dengan performa terbaiknya masing-masing. MPC2.3 dapat mencapai *set point* lebih cepat dibanding MPC2 dan MPC3. MPC2.3 juga dapat menangani gangguan dan mengembalikan komposisi produk bawah ke *set point* yang ditentukan dengan baik pada  $t=200$ .



Gambar 4.11. Perbandingan MPC2, MPC3, dan MPC2.3; skenario 1

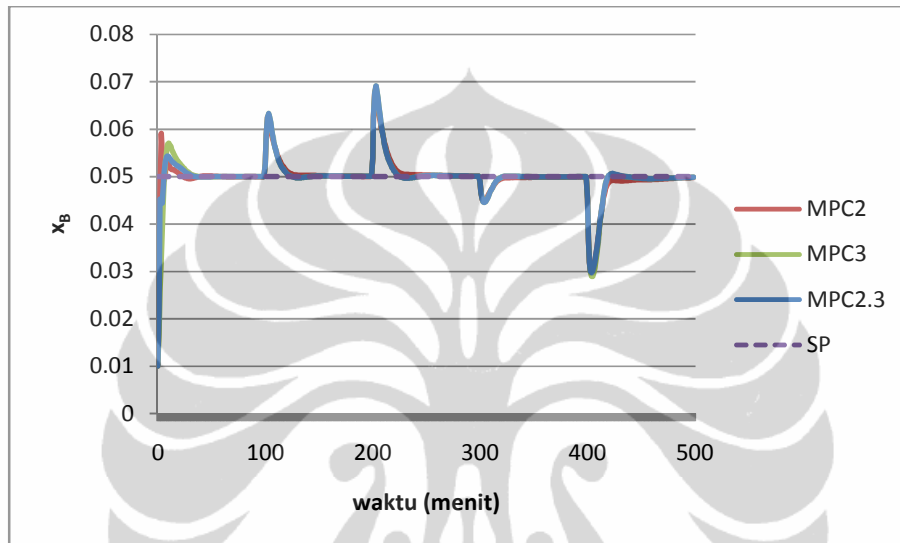
Gambar 4.12 menunjukkan karakteristik MPC2.3.4 pada skenario 1 yang merupakan gabungan dari karakteristik single MPC yang dijalankan sesuai dengan performa terbaiknya masing-masing. MPC2.3.4 dapat mencapai set point lebih cepat dibanding MPC2, MPC3 maupun MPC4. MPC2.3.4 juga dapat menangani gangguan dan mengembalikan komposisi produk bawah ke *set point* yang ditentukan dengan baik pada  $t=200$ .



Gambar 4.12. Perbandingan MPC2, MPC3, MPC4, MPC2.3.4; skenario 1



Gambar 4.13 menunjukkan karakteristik MPC2.3 pada skenario 2 yang merupakan gabungan dari karakteristik single MPC2 dan MPC3 yang dijalankan sesuai dengan performa terbaiknya masing-masing. MPC2.3 dapat mencapai set point lebih cepat dibanding MPC2 dan MPC3. MPC2.3 juga dapat menangani gangguan *multi step* dan mengembalikan komposisi produk bawah ke *set point* yang ditentukan dengan baik..



Gambar 4.13. Perbandingan MPC2, MPC3, dan MPC2.3; skenario 2

Untuk mendapatkan MMPC terbaik, masing-masing MPC lokal penyusunnya harus terdiri dari MPC lokal yang dapat menjangkau *setpoint* secara cepat dan menangani gangguan dengan baik. Kemudian, untuk menentukan spesifikasi *switch* yang tepat adalah dengan melihat respon masing-masing MPC lokal dan menentukan batasan daerah operasi dimana MPC lokal tersebut bekerja dengan baik.

Di dalam penelitian ini didapatkan bahwa untuk skenario 1, hasil IAE pengendalian MPC secara keseluruhan lebih baik dibanding pengendalian PI. Hasil IAE pengendalian MMPC secara keseluruhan lebih baik dibanding pengendalian MPC, walaupun hasil terbaik pengendalian MMPC belum bisa mengalahkan hasil terbaik pengendalian MPC.

Untuk skenario 2, hasil IAE pengendalian MPC yang bisa mengalahkan pengendalian PI hanya ada dua MPC, yaitu MPC2 dan MPC3. Oleh karena itu, hanya ada satu model MMPC yang dapat dibuat, yaitu

MPC2.3. Walaupun dalam skenario 2 hanya ada satu MMPC yang dapat dibuat, performa MPC2.3 dapat mengalahkan performa single MPC terbaik skenario 2 yaitu MPC2. Hal ini menunjukkan bahwa MMPC dapat mengatasi gangguan yang terjadi pada kolom distilasi dengan efektif, baik pada skenario 1 maupun skenario 2.

Untuk mendapatkan MMPC yang terbaik dibandingkan MMPC lainnya diperlukan percobaan secara teliti. Namun, secara keseluruhan MMPC akan memberikan hasil yang lebih baik dibandingkan pengendalian dengan single MPC. Tabel 4.13 menunjukkan IAE dan ISE untuk kontroler PI, single MPC terbaik, dan MMPC pada skenario 1 dan skenario 2.

Tabel 4.13. Perbandingan IAE dan ISE Kontroler PI, MPC, dan MMPC

Skenario 1			Skenario 2		
Kontroler	IAE	ISE	Kontroler	IAE	ISE
PI	0.7494	0.0160	PI	0.9808	0.0168
MPC2	0.2564	0.0036	MPC2	0.8016	0.0094
MPC10	0.7427	0.0185	MPC3	0.8568	0.0115
MPC2.3	0.2582	0.0036	MPC2.3	0.7730	0.0096
MPC5.6	0.5433	0.0102			
MPC2.3.4	0.2605	0.0037			
MPC2.3.5	0.2624	0.0038			
MPC2.3.6	0.2639	0.0038			
MPC2.4.5	0.2911	0.0040			
MPC2.4.6	0.2927	0.0040			
MPC2.5.6	0.2639	0.0038			
MPC3.4.5	0.3735	0.0056			
MPC3.4.6	0.3751	0.0057			
MPC3.5.6	0.4019	0.0062			
MPC4.5.6	0.4797	0.0083			

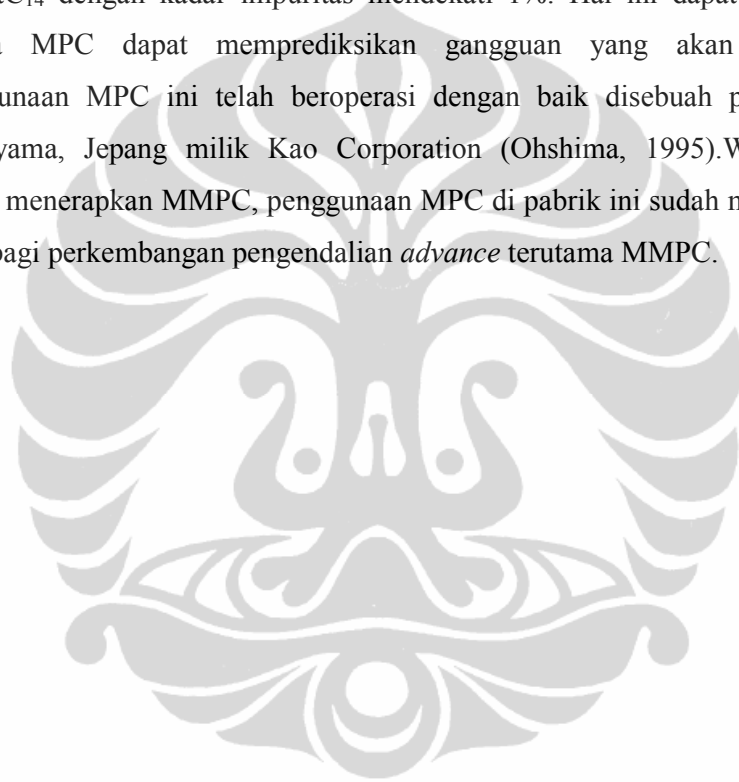
Dari table 4.13 dapat dilihat bahwa untuk skenario 1, hasil IAE dari MMPC dua model yang paling besar, yaitu MPC5.6 dengan nilai 0,5433 berhasil mengalahkan hasil IAE *single* MPC yang paling besar yaitu MPC10 dengan nilai 0,427. Hal ini menunjukkan bahwa performa MMPC dua model lebih baik secara keseluruhan dibanding performa *single* MPC, walaupun performa MMPC dua model terbaik yaitu MPC 2.3 belum bisa mengalahkan performa *single* MPC terbaik yaitu MPC2. MMPC tiga model

yang paling besar, yaitu MPC4.5.6 dengan nilai 0,4797 berhasil mengalahkan hasil IAE MMPC dua model yang paling besar yaitu MPC5.6 dengan nilai 0,5433. Hal ini membuktikan bahwa MMPC tiga model lebih baik performanya dibanding MMPC dua model, walaupun MMPC tiga model terbaik yaitu MPC2.3.4 belum bisa mengalahkan MMPC dua model terbaik yaitu MPC2.3. Namun, untuk skenario 1 secara keseluruhan MMPC tiga model lebih efektif menangani gangguan pada pengendalian kolom distilasi dibanding MMPC dua model. Sedangkan untuk skenario 2 dapat dilihat bahwa MMPC yang dibuat sudah memenuhi kualifikasi untuk menangani gangguan pada pengendalian kolom distilasi dengan lebih baik dibanding *single* MPC.

Dari penelitian ini dapat dilihat bahwa *single* MPC terbukti cukup dan efektif dalam menangani gangguan yang kecil dan sedikit pada pengendalian kolom distilasi. Namun, performa MMPC tetap lebih baik secara keseluruhan dibanding *single* MPC. Sedangkan untuk gangguan yang lebih banyak dan variatif, MMPC terbukti lebih efektif untuk menangani gangguan.

Salah satu contoh penerapan MPC untuk penanganan gangguan di industri adalah pada pabrik pemisahan asam lemak jenuh. Pabrik ini terdiri dari evaporator dan tiga kolom distilasi jenis packed-bed. Ketiga kolom dioperasikan dengan tekanan kurang dari tekanan atmosfer. Campuran asam lemak jenuh, dengan komponen mulai dari asam lemak  $C_6$  hingga  $C_{18}$  diumpankan ke kolom distilasi pertama.  $C_6$ ,  $C_8$ , dan  $C_{10}$  didistilasi pada kolom pertama. Aliran bawah dimasukkan ke evaporator, di mana asam lemak diuapkan dan dipisahkan dari pengotor seperti ester. Asam lemak menguap diumpankan ke kolom distilasi kedua, di mana  $C_{12}$  didistilasi. Pada kolom kedua, konsentrasi  $C_{12}$  pada produk atas tidak diperbolehkan di bawah 98%. Produk bawah dari kolom kedua diumpankan ke kolom yang ketiga, yang memisahkan komponen  $C_{14}$  dari  $C_{16}$  dan  $C_{18}$ . Distilasi kolom ketiga harus dioperasikan untuk menjaga konsentrasi  $C_{14}$  dalam distilat lebih tinggi dari 98%.

Umpan asam lemak yang diproses berasal dari reaktor dimana ada dua komponen dalam asam lemak yang berbeda komposisinya. Hal ini merupakan faktor yang mengganggu proses distilasi. Salah satu cara menangani gangguan tersebut adalah dengan memanipulasi laju alir pemanas reboiler dan laju refluks. Laju alir umpan dan komposisi umpan adalah gangguan ke kontroler yang terukur. Penggunaan single MPC pada sistem ini menunjukkan hasil yang cukup memuaskan untuk respon output distilat $C_{14}$  dengan kadar impuritas mendekati 1%. Hal ini dapat tercapai karena MPC dapat memprediksikan gangguan yang akan terjadi. Penggunaan MPC ini telah beroperasi dengan baik di sebuah pabrik di Wakayama, Jepang milik Kao Corporation (Ohshima, 1995). Walaupun belum menerapkan MMPC, penggunaan MPC di pabrik ini sudah membuka jalan bagi perkembangan pengendalian *advance* terutama MMPC.



## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Berdasarkan hasil dan pembahasan penelitian ini, dapat disimpulkan:

1. MMPC yang didasarkan pada pemilihan model dengan metode RMPC secara keseluruhan memberikan hasil yang lebih baik dibandingkan pengendalian dengan single MPC.
2. Pada skenario 1, hasil terbaik diberikan oleh kontroler MPC2 dengan IAE 0,2564 dan ISE 0,0036.
3. Pada skenario 2, hasil terbaik diberikan oleh kontroler MPC2.3 dengan IAE 0,7730 dan ISE 0,0096.
4. Yang perlu diperhatikan dalam pemilihan MPC lokal yang digunakan pada MMPC adalah kemampuan MPC lokal tersebut menjangkau *setpoint* operasi dan menangani gangguan dengan cepat serta pengaturan *switch* antarmodel.
5. MPC lokal penyusun MMPC harus terdiri dari MPC lokal yang dapat menangani pemberian gangguan dengan baik dan mengembalikan komposisi produk ke *set point* yang diminta.

#### 5.2. Saran

Saran untuk penelitian selanjutnya adalah dilakukannya simulasi pengendalian dengan MMPC untuk sistem multivariabel lainnya dengan pengujian untuk beberapa aspek berikut ini sehingga apa yang telah disimpulkan oleh penelitian ini dapat divalidasi untuk plant lainnya:

- Plant nonlinear lain (selain Kolom A)
- Skenario perubahan variabel yang lain, misalnya dengan penambahan variabel *holdup* cairan

## DAFTAR REFERENSI

- Bequette, B. W. (1998). *Process Dynamics: Modeling, Analysis, and Simulation*. New Jersey: Prentice Hall, Inc.
- Camacho, E. F., Bordons, C. (2007). Nonlinear Model Predictive Control: An Introductory Review: University of Seville, Spain.
- Cynthia, Vania. (2012). *Distillation Column Control by Using MMPC Based on RMPC Method*. Faculty of Engineering, University of Indonesia, Depok, Indonesia.
- Dougherty, D., Cooper, D. (2003a). A practical multiple model adaptive strategy for multivariable model predictive control. *Control Engineering Practice*, 11, 649-664.
- Dougherty, D., Cooper, D. (2003b). A practical multiple model adaptive strategy for single-loop MPC. *Control Engineering Practice*, 11, 141-159.
- ESDM, K. (2010). *Handbook of Energy and Economic Statistics of Indonesia*.
- Huang, H. P., Jeng, J.C., Chiang, C.H., Pan, W. (2003). A direct method for multi-loop PI/PID controller design. *Journal of Process Control*, 13, 769-786.
- Humphrey, J. L., Seibert, A.F., Koort, R.A. (1991). Separation Technologies - Advances and Priorities.
- Johnson, M. A., Moradi, M.H. (2005). *PID Control: New Identification and Design Methods*. London: Springer.
- Khaisongkram, W., Banjerpongchai, D (2006). Linear controller design and performance limits of binary distillation column subject to disturbances with bounds on magnitudes and rates of change. *Journal of Process Control*, 16, 845–854.
- Luyben, M. L., Luyben, W.L. (1997). *Essentials of Process Control*. Singapore: McGraw-Hill.
- Marlin, T. (2000). *Process Control: Designing Processes and Control Systems for Dynamic Performance 2nd Edition*. New York: McGraw-Hill.
- Ohshima, M., Ohno, H., Hashimoto, I., Sasajima, M., Maejima, M., Tsuto, K., Ogawa, T. (1995). *Model predictive control with adaptive disturbance prediction and its application to fatty acid distillation column control*. Department of Chemical Engineering, Kyoto University.
- Palma, F. D., Magni, L. (2004). A multi-model structure for model predictive control. *Annual Reviews in Control*, 28, 47-52.
- Porfirio, C. R., Neto, E. Almeida., Odloak, D. (2003). Multi-model predictive control of an industrial C3/C4 splitter. *Control Engineering Practice*, 11, 765-779.
- Qin, S. J., Badgwell, T.A. (2002). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11, 733-764.
- Riggs, J. B. (2000). Comparison of Advanced Distillation Control Method. Lubbock, Texas: Texas Tech University.
- Skogestad, S. MATLAB Distillation column model ("Column A"), from [http://www.nt.ntnu.no/users/skoge/book/matlab\\_m/cola/cola.html](http://www.nt.ntnu.no/users/skoge/book/matlab_m/cola/cola.html)
- Skogestad, S. (1997). Dynamics and Control of Distillation Columns: A Tutorial Introduction. *Trans IChemE*, 75.

- Wahid, A., Ahmad, A. (2009). *Representative Model Predictive Control*. Paper presented at the Proceeding of the 11th International Conference on QiR (Quality in Research), Faculty of Engineering, University of Indonesia, Depok, Indonesia.
- Zhao, Futao., Gupta, Yash P. (2004). *A simplified predictive control algorithm for disturbance rejection*. Department of Chemical Engineering, Dalhousie University, Halifax, Canada



## Lampiran 1: Program Identifikasi Sistem dengan Perubahan Feed Rate (F)

```

% Identifikasi Sistem dengan Perubahan F
open_system ('colas_lv_nonlin_F');
N=500;
t=1:N;
L=1*(1+0.2);
sim('colas_lv_nonlin_F',t);
FT=F*ones(1,500)';
for i=1:N
    Kp_y1_1=(y1(N)-y1(1))/(F-1);
    Kp_y2_1=(y2(N)-y2(1))/(F-1);
    y1_28=(0.28*(y1(N)-y1(1)))+y1(1);
    y1_63=(0.63*(y1(N)-y1(1)))+y1(1);
    y2_28=(0.28*(y2(N)-y2(1)))+y2(1);
    y2_63=(0.63*(y2(N)-y2(1)))+y2(1);
    if y1_28>y1(i) & y1_28<y1(i+1)
        t_28y1=i+((y1_28-y1(i))/(y1(i+1)-y1(i)));
    end
    if y1_63>y1(i) & y1_63<y1(i+1)
        t_63y1=i+((y1_63-y1(i))/(y1(i+1)-y1(i)));
    end
    if y2_28>y2(i) & y2_28<y2(i+1)
        t_28y2=i+((y2_28-y2(i))/(y2(i+1)-y2(i)));
    end
    if y2_63>y2(i) & y2_63<y2(i+1)
        t_63y2=i+((y2_63-y2(i))/(y2(i+1)-y2(i)));
    end
    end
    tau_y1_1=1.5*(t_63y1-t_28y1);
    tetta_y1_1=t_63y1-tau_y1_1;
    tau_y2_1=1.5*(t_63y2-t_28y2);
    tetta_y2_1=t_63y2-tau_y2_1;
    GsLy1_1=[Kp_y1_1 tetta_y1_1 tau_y1_1]
    GsLy2_1=[Kp_y2_1 tetta_y2_1 tau_y2_1]
%%
N=500;
t=1:N;
F=1*(1+0.3);
sim('colas_lv_nonlin_F',t);
FT=F*ones(1,500)';
for i=1:N
    Kp_y1_2=(y1(N)-y1(1))/(F-1);
    Kp_y2_2=(y2(N)-y2(1))/(F-1);
    y1_28=(0.28*(y1(N)-y1(1)))+y1(1);
    y1_63=(0.63*(y1(N)-y1(1)))+y1(1);
    y2_28=(0.28*(y2(N)-y2(1)))+y2(1);
    y2_63=(0.63*(y2(N)-y2(1)))+y2(1);
    if y1_28>y1(i) & y1_28<y1(i+1)
        t_28y1=i+((y1_28-y1(i))/(y1(i+1)-y1(i)));
    end
    if y1_63>y1(i) & y1_63<y1(i+1)
        t_63y1=i+((y1_63-y1(i))/(y1(i+1)-y1(i)));
    end
    if y2_28>y2(i) & y2_28<y2(i+1)
        t_28y2=i+((y2_28-y2(i))/(y2(i+1)-y2(i)));
    end
    if y2_63>y2(i) & y2_63<y2(i+1)
        t_63y2=i+((y2_63-y2(i))/(y2(i+1)-y2(i)));
    end
end

```



```

end
end
tau_y1_2=1.5*(t_63y1-t_28y1);
tetta_y1_2=t_63y1-tau_y1_2;
tau_y2_2=1.5*(t_63y2-t_28y2);
tetta_y2_2=t_63y2-tau_y2_2;
GsLy1_2=[Kp_y1_2 tetta_y1_2 tau_y1_2]
GsLy2_2=[Kp_y2_2 tetta_y2_2 tau_y2_2]
%%
N=500;
t=1:N;
F=1*(1+0.5);
sim('colas_lv_nonlin_F',t);
FT=F*ones(1,500)';
for i=1:N
    Kp_y1_3=(y1(N)-y1(1))/(F-1);
    Kp_y2_3=(y2(N)-y2(1))/(F-1);
    y1_28=(0.28*(y1(N)-y1(1)))+y1(1);
    y1_63=(0.63*(y1(N)-y1(1)))+y1(1);
    y2_28=(0.28*(y2(N)-y2(1)))+y2(1);
    y2_63=(0.63*(y2(N)-y2(1)))+y2(1);
    if y1_28>y1(i) & y1_28<y1(i+1)
        t_28y1=i+((y1_28-y1(i))/(y1(i+1)-y1(i)));
    end
    if y1_63>y1(i) & y1_63<y1(i+1)
        t_63y1=i+((y1_63-y1(i))/(y1(i+1)-y1(i)));
    end
    if y2_28>y2(i) & y2_28<y2(i+1)
        t_28y2=i+((y2_28-y2(i))/(y2(i+1)-y2(i)));
    end
    if y2_63>y2(i) & y2_63<y2(i+1)
        t_63y2=i+((y2_63-y2(i))/(y2(i+1)-y2(i)));
    end
end
tau_y1_3=1.5*(t_63y1-t_28y1);
tetta_y1_3=t_63y1-tau_y1_3;
tau_y2_3=1.5*(t_63y2-t_28y2);
tetta_y2_3=t_63y2-tau_y2_3;
GsLy1_3=[Kp_y1_3 tetta_y1_3 tau_y1_3]
GsLy2_3=[Kp_y2_3 tetta_y2_3 tau_y2_3]
%%
N=500;
t=1:N;
F=1*(1+0.7);
sim('colas_lv_nonlin_F',t);
FT=F*ones(1,500)';
for i=1:N
    Kp_y1_4=(y1(N)-y1(1))/(F-1);
    Kp_y2_4=(y2(N)-y2(1))/(F-1);
    y1_28=(0.28*(y1(N)-y1(1)))+y1(1);
    y1_63=(0.63*(y1(N)-y1(1)))+y1(1);
    y2_28=(0.28*(y2(N)-y2(1)))+y2(1);
    y2_63=(0.63*(y2(N)-y2(1)))+y2(1);
    if y1_28>y1(i) & y1_28<y1(i+1)
        t_28y1=i+((y1_28-y1(i))/(y1(i+1)-y1(i)));
    end
    if y1_63>y1(i) & y1_63<y1(i+1)
        t_63y1=i+((y1_63-y1(i))/(y1(i+1)-y1(i)));
    end
end

```

```
if y2_28>y2(i) & y2_28<y2(i+1)
    t_28y2=i+((y2_28-y2(i))/(y2(i+1)-y2(i)));
end
if y2_63>y2(i) & y2_63<y2(i+1)
    t_63y2=i+((y2_63-y2(i))/(y2(i+1)-y2(i)));
end
end
tau_y1_4=1.5*(t_63y1-t_28y1);
tetta_y1_4=t_63y1-tau_y1_4;
tau_y2_4=1.5*(t_63y2-t_28y2);
tetta_y2_4=t_63y2-tau_y2_4;
GsLy1_4=[Kp_y1_4 tetta_y1_4 tau_y1_4]
GsLy2_4=[Kp_y2_4 tetta_y2_4 tau_y2_4]
```



## Lampiran 2: Program Identifikasi Sistem dengan Perubahan Komposisi Feed (zF)

```

% Identifikasi Sistem dengan Perubahan zF
open_system ('colas_lv_nonlin_zF');
N=500;
t=1:N;
zF=0.5*(1+0.2);
sim('colas_lv_nonlin_zF',t);
zFT=zF*ones(1,500)';
for i=1:N
    Kp_y1_1=(y1(N)-y1(1))/(zF-0.5);
    Kp_y2_1=(y2(N)-y2(1))/(zF-0.5);
    y1_28=(0.28*(y1(N)-y1(1)))+y1(1);
    y1_63=(0.63*(y1(N)-y1(1)))+y1(1);
    y2_28=(0.28*(y2(N)-y2(1)))+y2(1);
    y2_63=(0.63*(y2(N)-y2(1)))+y2(1);
    if y1_28<y1(i) & y1_28>y1(i+1)
        t_28y1=i+((y1_28-y1(i))/(y1(i+1)-y1(i)));
    end
    if y1_63<y1(i) & y1_63>y1(i+1)
        t_63y1=i+((y1_63-y1(i))/(y1(i+1)-y1(i)));
    end
    if y2_28<y2(i) & y2_28>y2(i+1)
        t_28y2=i+((y2_28-y2(i))/(y2(i+1)-y2(i)));
    end
    if y2_63<y2(i) & y2_63>y2(i+1)
        t_63y2=i+((y2_63-y2(i))/(y2(i+1)-y2(i)));
    end
    end
    tau_y1_1=1.5*(t_63y1-t_28y1);
    tetta_y1_1=t_63y1-tau_y1_1;
    tau_y2_1=1.5*(t_63y2-t_28y2);
    tetta_y2_1=t_63y2-tau_y2_1;
    GsVy1_1=[Kp_y1_1 tetta_y1_1 tau_y1_1]
    GsVy2_1=[Kp_y2_1 tetta_y2_1 tau_y2_1]
    %%
N=500;
t=1:N;
zF=0.5*(1+0.3);
sim('colas_lv_nonlin_zF',t);
zFT=zF*ones(1,500)';
for i=1:N
    Kp_y1_2=(y1(N)-y1(1))/(zF-0.5);
    Kp_y2_2=(y2(N)-y2(1))/(zF-0.5);
    y1_28=(0.28*(y1(N)-y1(1)))+y1(1);
    y1_63=(0.63*(y1(N)-y1(1)))+y1(1);
    y2_28=(0.28*(y2(N)-y2(1)))+y2(1);
    y2_63=(0.63*(y2(N)-y2(1)))+y2(1);
    if y1_28<y1(i) & y1_28>y1(i+1)
        t_28y1=i+((y1_28-y1(i))/(y1(i+1)-y1(i)));
    end
    if y1_63<y1(i) & y1_63>y1(i+1)
        t_63y1=i+((y1_63-y1(i))/(y1(i+1)-y1(i)));
    end
    if y2_28<y2(i) & y2_28>y2(i+1)
        t_28y2=i+((y2_28-y2(i))/(y2(i+1)-y2(i)));
    end
    if y2_63<y2(i) & y2_63>y2(i+1)
        t_63y2=i+((y2_63-y2(i))/(y2(i+1)-y2(i)));
    end

```

```

end
end
tau_y1_2=1.5*(t_63y1-t_28y1);
tetta_y1_2=t_63y1-tau_y1_2;
tau_y2_2=1.5*(t_63y2-t_28y2);
tetta_y2_2=t_63y2-tau_y2_2;
GsVy1_2=[Kp_y1_2 tetta_y1_2 tau_y1_2]
GsVy2_2=[Kp_y2_2 tetta_y2_2 tau_y2_2]
%%
N=500;
t=1:N;
zF=0.5*(1+0.5);
sim('colas_lv_nonlin_zF',t);
zFT=zF*ones(1,500)';
for i=1:N
    Kp_y1_3=(y1(N)-y1(1))/(zF-0.5);
    Kp_y2_3=(y2(N)-y2(1))/(zF-0.5);
    y1_28=(0.28*(y1(N)-y1(1)))+y1(1);
    y1_63=(0.63*(y1(N)-y1(1)))+y1(1);
    y2_28=(0.28*(y2(N)-y2(1)))+y2(1);
    y2_63=(0.63*(y2(N)-y2(1)))+y2(1);
    if y1_28<y1(i) & y1_28>y1(i+1)
        t_28y1=i+((y1_28-y1(i))/(y1(i+1)-y1(i)));
    end
    if y1_63<y1(i) & y1_63>y1(i+1)
        t_63y1=i+((y1_63-y1(i))/(y1(i+1)-y1(i)));
    end
    if y2_28<y2(i) & y2_28>y2(i+1)
        t_28y2=i+((y2_28-y2(i))/(y2(i+1)-y2(i)));
    end
    if y2_63<y2(i) & y2_63>y2(i+1)
        t_63y2=i+((y2_63-y2(i))/(y2(i+1)-y2(i)));
    end
end
tau_y1_3=1.5*(t_63y1-t_28y1);
tetta_y1_3=t_63y1-tau_y1_3;
tau_y2_3=1.5*(t_63y2-t_28y2);
tetta_y2_3=t_63y2-tau_y2_3;
GsVy1_3=[Kp_y1_3 tetta_y1_3 tau_y1_3]
GsVy2_3=[Kp_y2_3 tetta_y2_3 tau_y2_3]
%%
N=500;
t=1:N;
zF=0.5*(1+0.7);
sim('colas_lv_nonlin_zF',t);
zFT=zF*ones(1,500)';
for i=1:N
    Kp_y1_4=(y1(N)-y1(1))/(zF-0.5);
    Kp_y2_4=(y2(N)-y2(1))/(zF-0.5);
    y1_28=(0.28*(y1(N)-y1(1)))+y1(1);
    y1_63=(0.63*(y1(N)-y1(1)))+y1(1);
    y2_28=(0.28*(y2(N)-y2(1)))+y2(1);
    y2_63=(0.63*(y2(N)-y2(1)))+y2(1);
    if y1_28<y1(i) & y1_28>y1(i+1)
        t_28y1=i+((y1_28-y1(i))/(y1(i+1)-y1(i)));
    end
    if y1_63<y1(i) & y1_63>y1(i+1)
        t_63y1=i+((y1_63-y1(i))/(y1(i+1)-y1(i)));
    end
end

```

```
if y2_28<y2(i) & y2_28>y2(i+1)
    t_28y2=i+((y2_28-y2(i))/(y2(i+1)-y2(i)));
end
if y2_63<y2(i) & y2_63>y2(i+1)
    t_63y2=i+((y2_63-y2(i))/(y2(i+1)-y2(i)));
end
end
tau_y1_4=1.5*(t_63y1-t_28y1);
tetta_y1_4=t_63y1-tau_y1_4;
tau_y2_4=1.5*(t_63y2-t_28y2);
tetta_y2_4=t_63y2-tau_y2_4;
GsVy1_4=[Kp_y1_4 tetta_y1_4 tau_y1_4]
GsVy2_4=[Kp_y2_4 tetta_y2_4 tau_y2_4]
```



## Lampiran 3: Program Simulasi Kontroler PI

```

open_system('Colas_PI')
%% PI_Setpoints
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=0.05;
T(t1:N,:)=0.05;
T(t2:N,:)=0.05;
T(t3:N,:)=0.05;
T(t4:N,:)=0.05;
Cola_Setpoints.time = [1:N]';
Cola_Setpoints.signals.values = [T(1:N,1) 0.99*ones(N,1)];
%%
% Simulate model.
sim('Colas_PI',[1:N])
%figure(12)
subplot 211%plot grafik yD
plot(t,y1,'b')
subplot 212%plot grafik yB
plot(t,y2,'k')
%%
E=abs (T(:,1) - y2(:,1));
IAE_PI = ones (1,N) *abs (T(:,1) - y2(:,1))
ISE_PI = (T(:,1) - y2(:,1))'*(T(:,1) - y2(:,1))
%s = xlswrite('SingleMPC.xlsx', y2(:,1), 'PI', 'C2:C201')
%s = xlswrite('SingleMPC.xlsx', y1(:,1), 'PI', 'D2:D201')

```

## Lampiran 4: Program Simulasi Single MPC pada Skenario 1

```

if ~mpcchecktoolboxinstalled('simulink')
    disp('Simulink(R) is required to run this demo.')
return
end
if ~mpcchecktoolboxinstalled('scd')
    disp('Simulink Control Design(R) is required to run this
demo.')
return
end
%%
%% Set initial states and inputs
yp = [0.99; 0.02];
up = [2.7063; 3.2063; 1; 0.5; 1];
%%
%% Compute initial condition
[x0, u0, y0, dx] = trim('cola_inout', [], up, yp, [], [1 2], 2);
%%
%% Obtain linear model
[a, b, c, d] = linmod('cola_inout', x0, u0);
%%
%% Verify that the linear model is open-loop stable at this
condition
eig(a)
%%
%% All eigenvalues are negative so the plant is open-loop stable.
%
% Configure a state-space model for use in MPC
Cola = ss(a, b, c, d);
Cola.InputName = {'LT', 'VB', 'F', 'zF', 'qF'};
Cola.OutputName = {'y_D', 'y_B'};
%%
% Designate the reflux and boil up flow as the manipulated
variable.
Cola.InputGroup.MV = [1,2];
%%
% The other three inputs are unmeasured disturbances.
Cola.InputGroup.UD = [3,4,5];
%%
% Create the MPC model format and set the nominal conditions.
Model.Plant = Cola;
Model.Nominal.X = x0;
Model.Nominal.Y = y0;
Model.Nominal.U = [u0(1);u0(2);0;0;0]; % NOTE: nominal values for
unmeasured disturbance must be zero.

%% Define a MPC Controller
Ts = 0.1; % Sampling time (minutes)
P = 12;
M = 2;
MPC = mpc(Model, Ts,P,M);
%%
% The goal will be to track a specified transition in the bottom
% concentration. The top concentration will be measured and used
in
% state estimation but the controller will not attempt to regulate
it
% directly. It will vary as needed to regulate the bottom

```

```

concentration. Thus,
% set its MPC weight to zero.
MPC.Weights.OV = [1 1];
% Controller blocks in a feedback configuration.
open_system('Cola_MMPC_1')

%%
% Set plot scales and simulate the response.
set_param('Cola_MMPC_1/Measurements','Open','on')
set_param('Cola_MMPC_1/Manipulated Var','Open','on')

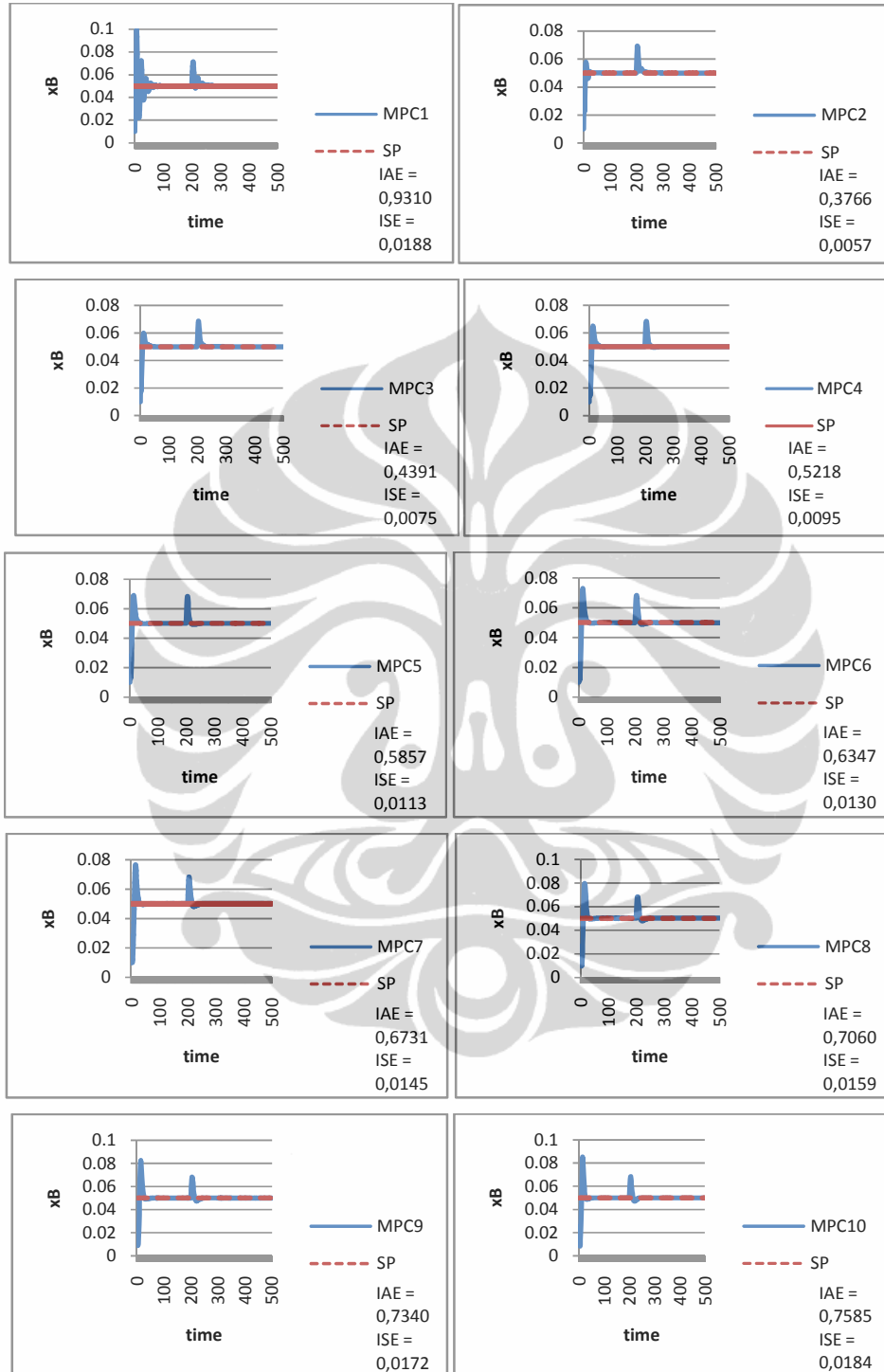
%% Modify Controller Design: Increase Estimator Signal-to-Noise
% The controller is stable but its disturbance rejection is
sluggish. Try
% increasing the state estimator signal-to-noise by a factor of 10
and test
% the design again.
%D = ss(getindist(MPC));
%D.b = eye(2)*10;
%set(D,'InputName',[],'OutputName',[],'InputGroup',[],'OutputGroup',
[],[]);
%setindist(MPC,'model',D);
%% Cola_Setpoints
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=0.05;
T(t1:N,:)=0.05;
T(t2:N,:)=0.05;
T(t3:N,:)=0.05;
T(t4:N,:)=0.05;
Cola_Setpoints.time = [1:N]';
Cola_Setpoints.signals.values = [0.99*ones(N,1) T(1:N,1)];
%% Cola_Disturbance
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=1.0;
T(t1:N,:)=1.2;
T(t2:N,:)=1.5;
T(t3:N,:)=1.4;
T(t4:N,:)=1.0;
Cola_Disturbance.time = [1:N]';
Cola_Disturbance.signals.values = T(1:N,1);
%%
open_system('Cola_MMPC_1/MPC Controller')
%%
% Simulate model.
sim('Cola_MMPC_1',1:N)
t=1:N;
%figure(2)
subplot 211
plot(t,yD,'r--')
subplot 212

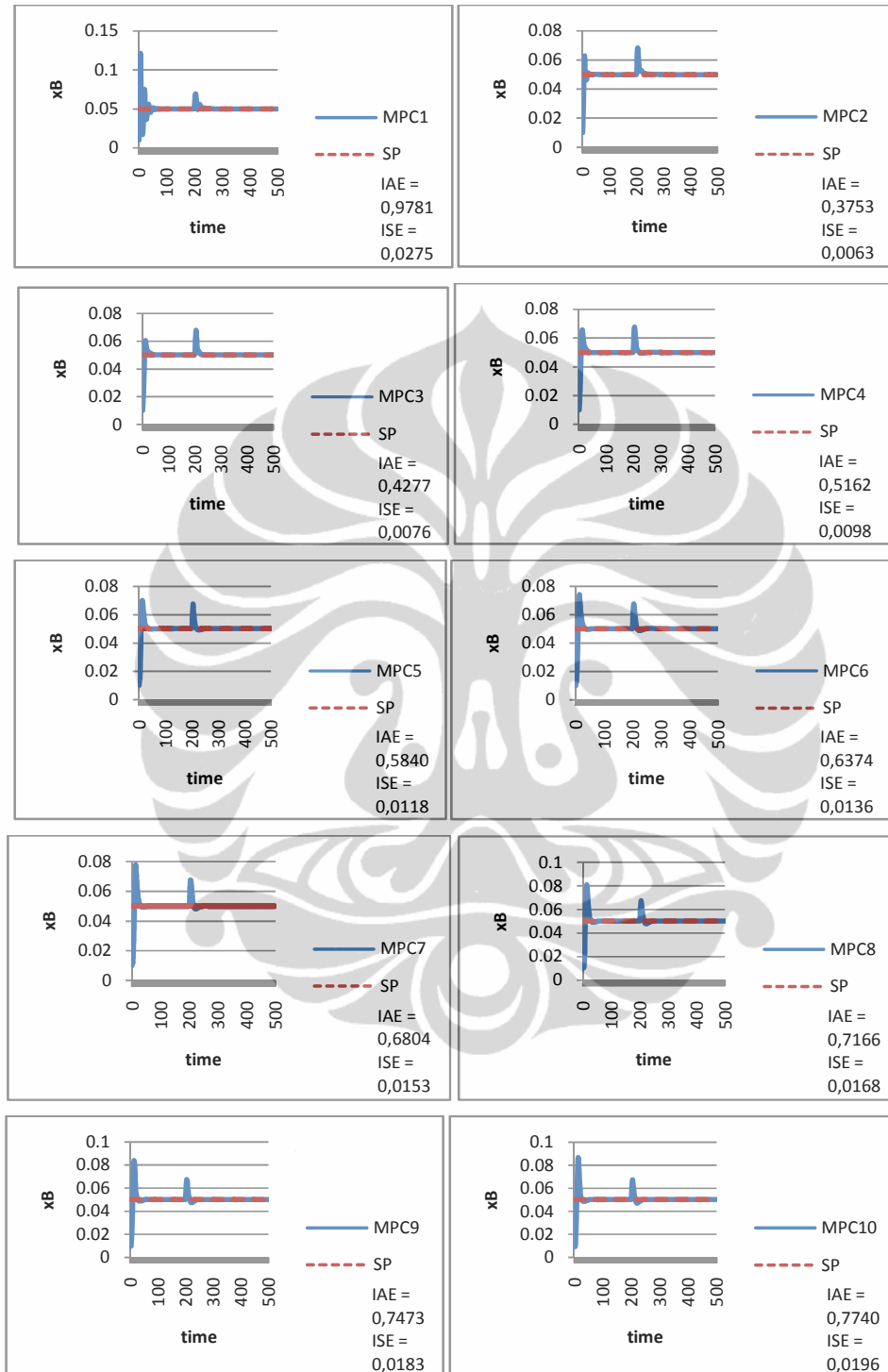
```

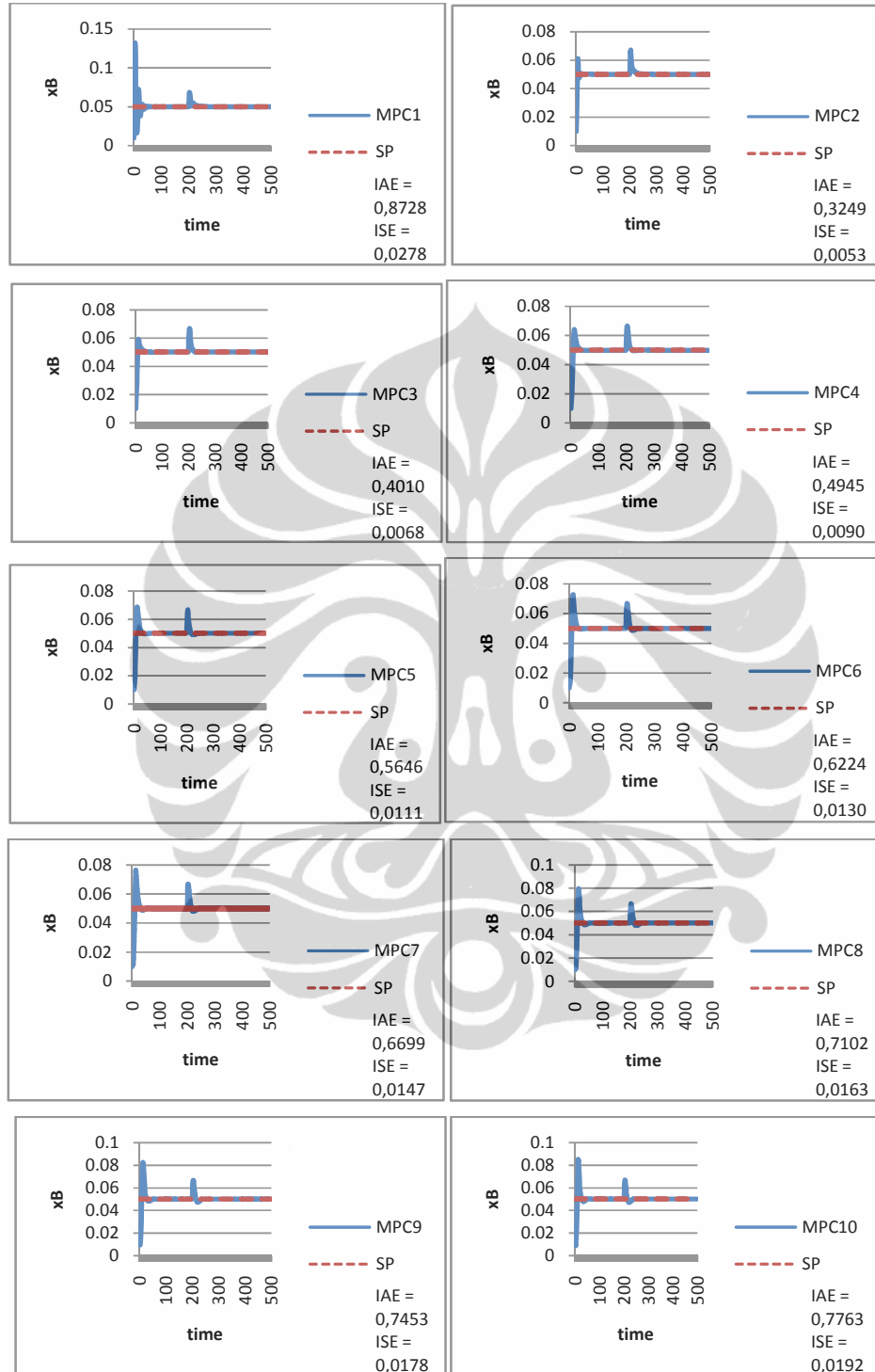


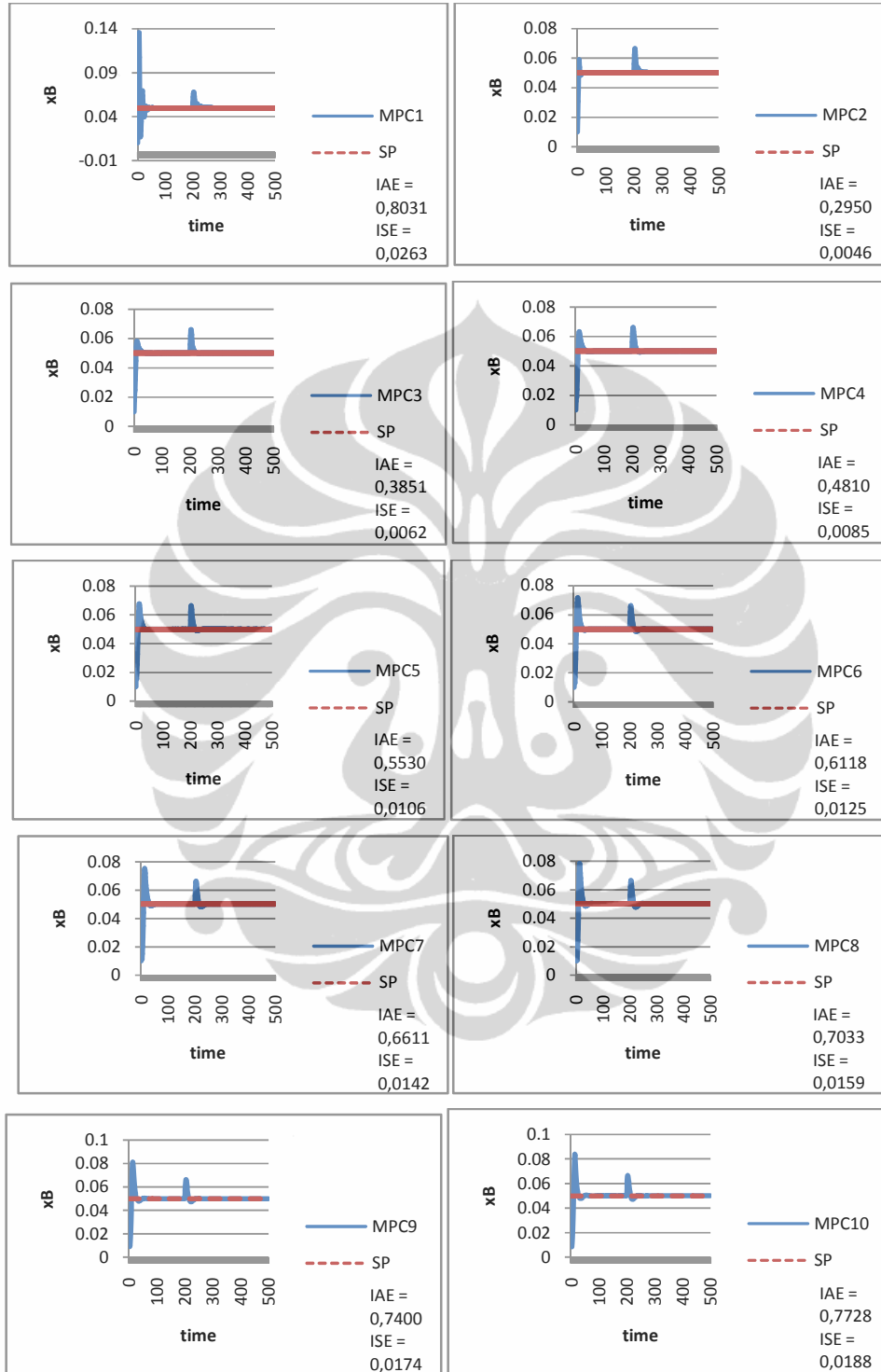
```
plot(t,xB,'g--')
%%
s = xlswrite('SP.xlsx', xB(:,1), 'SP1', 'D2:D301')
s = xlswrite('SP.xlsx', yD(:,1), 'SP1', 'O2:O301')
%%
IAE_MPC2 = ones (1,N)*abs (xB(:,2) - xB(:,1))
ISE_MPC2 = (xB(:,2) - xB(:,1))' * (xB(:,2) - xB(:,1))
```

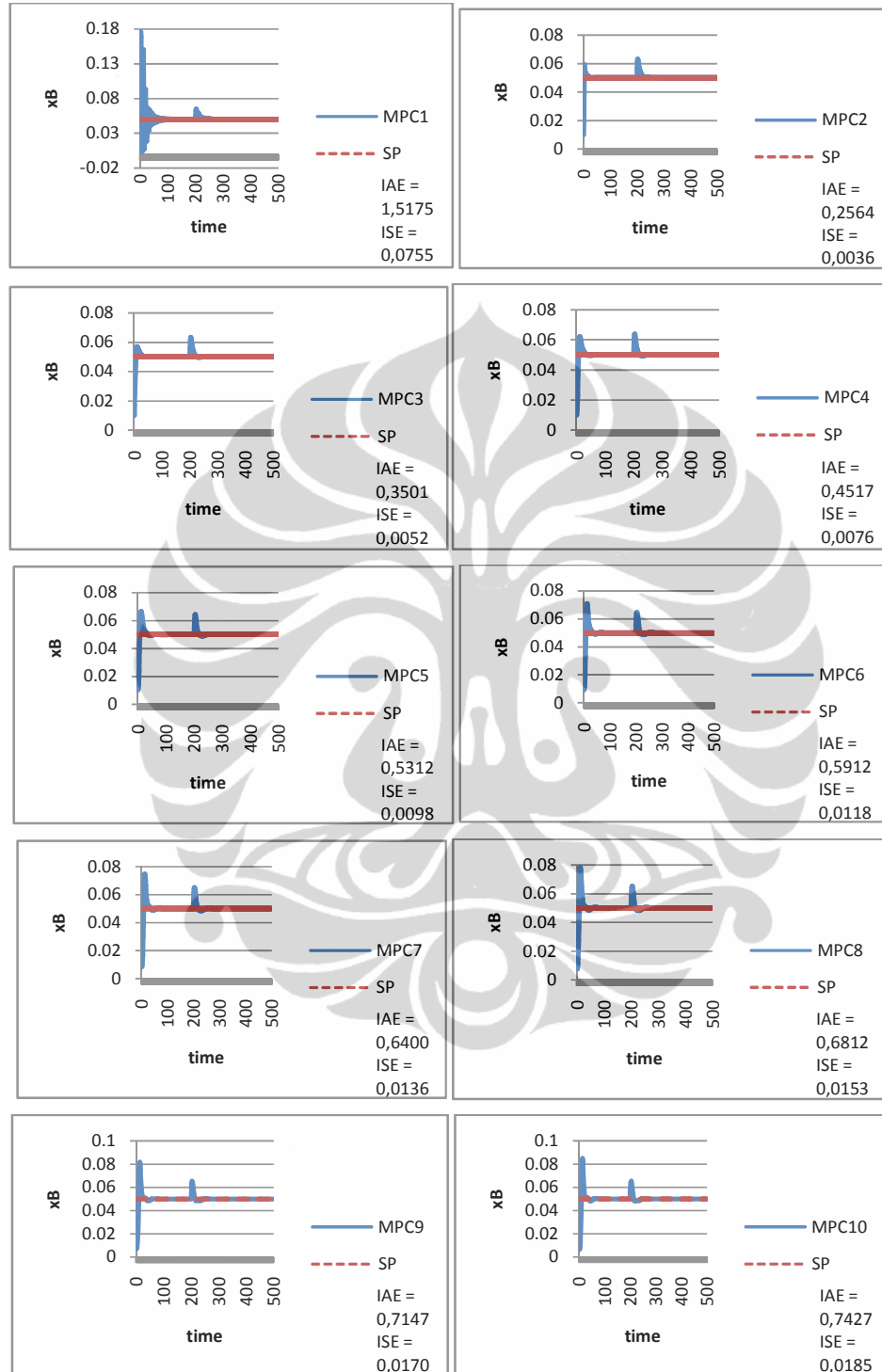


Lampiran 5: Respon  $x_B$  pada  $T_s = 0,72$ ;  $P = 66$ ;  $M = 26$ ; Skenario 1

Lampiran 6: Respon  $x_B$  pada  $T_s = 0,53$ ;  $P = 67$ ;  $M = 27$ ; Skenario 1

Lampiran 7: Respon  $x_B$  pada  $T_s = 0,36$ ;  $P = 69$ ;  $M = 29$ ; Skenario 1

Lampiran 8: Respon  $x_B$  pada  $T_s = 0,29$ ;  $P = 70$ ;  $M = 30$ ; Skenario 1

Lampiran 9: Respon  $x_B$  pada  $T_s = 0,1; P = 12; M = 2; \text{Skenario 1}$ 

## Lampiran 10: Program Simulasi Single MPC pada Skenario 2

```

if ~mpcchecktoolboxinstalled('simulink')
    disp('Simulink(R) is required to run this demo.')
return
end
if ~mpcchecktoolboxinstalled('scd')
    disp('Simulink Control Design(R) is required to run this
demo.')
return
end
%%
% Set initial states and inputs
yp = [0.99; 0.02];
up = [2.7063; 3.2063; 1; 0.5; 1];
%%
% Compute initial condition
[x0, u0, y0, dx] = trim('cola_inout', [], up, yp, [], [1 2], 2);
%%
% Obtain linear model
[a, b, c, d] = linmod('cola_inout', x0, u0);
%%
% Verify that the linear model is open-loop stable at this
condition
eig(a)
%%
% All eigenvalues are negative so the plant is open-loop stable.
%
% Configure a state-space model for use in MPC
Cola = ss(a, b, c, d);
Cola.InputName = {'LT', 'VB', 'F', 'zF', 'qF'};
Cola.OutputName = {'y_D', 'y_B'};
%%
% Designate the reflux and boil up flow as the manipulated
variable.
Cola.InputGroup.MV = [1,2];
%%
% The other three inputs are unmeasured disturbances.
Cola.InputGroup.UD = [3,4,5];
%%
% Create the MPC model format and set the nominal conditions.
Model.Plant = Cola;
Model.Nominal.X = x0;
Model.Nominal.Y = y0;
Model.Nominal.U = [u0(1);u0(2);0;0;0]; % NOTE: nominal values for
unmeasured disturbance must be zero.

%% Define a MPC Controller
Ts = 0.1; % Sampling time (minutes)
P = 12;
M = 2;
MPC = mpc(Model, Ts,P,M);
%%
% The goal will be to track a specified transition in the bottom
% concentration. The top concentration will be measured and used
in
% state estimation but the controller will not attempt to regulate
it
% directly. It will vary as needed to regulate the bottom

```

```

concentration. Thus,
% set its MPC weight to zero.
MPC.Weights.OV = [1 1];
% Controller blocks in a feedback configuration.
open_system('Cola_MMPC_1')

%%
% Set plot scales and simulate the response.
set_param('Cola_MMPC_1/Measurements','Open','on')
set_param('Cola_MMPC_1/Manipulated Var','Open','on')

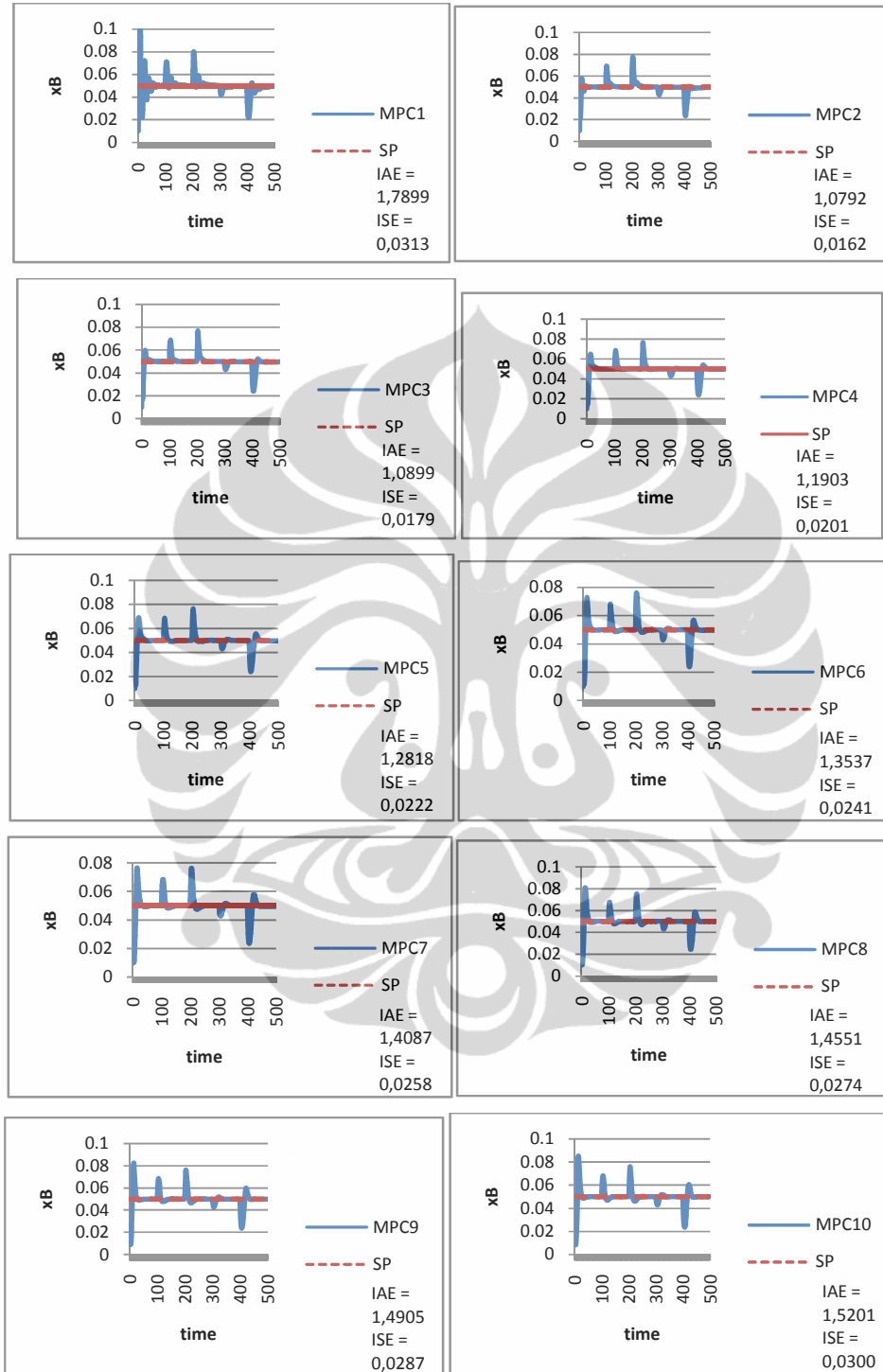
%% Modify Controller Design: Increase Estimator Signal-to-Noise
% The controller is stable but its disturbance rejection is
sluggish. Try
% increasing the state estimator signal-to-noise by a factor of 10
and test
% the design again.
%D = ss(getindist(MPC));
%D.b = eye(2)*10;
%set(D,'InputName',[],'OutputName',[],'InputGroup',[],'OutputGroup',
[],[]);
%setindist(MPC,'model',D);
%% Cola_Setpoints
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=0.05;
T(t1:N,:)=0.05;
T(t2:N,:)=0.05;
T(t3:N,:)=0.05;
T(t4:N,:)=0.05;
Cola_Setpoints.time = [1:N]';
Cola_Setpoints.signals.values = [0.99*ones(N,1) T(1:N,1)];
%% Cola_Disturbance
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=1.0;
T(t1:N,:)=1.2;
T(t2:N,:)=1.5;
T(t3:N,:)=1.4;
T(t4:N,:)=1.0;
Cola_Disturbance.time = [1:N]';
Cola_Disturbance.signals.values = T(1:N,1);
%%
open_system('Cola_MMPC_1/MPC Controller')
%%
% Simulate model.
sim('Cola_MMPC_1',1:N)
t=1:N;
%figure(2)
subplot 211
plot(t,yD,'r--')
subplot 212

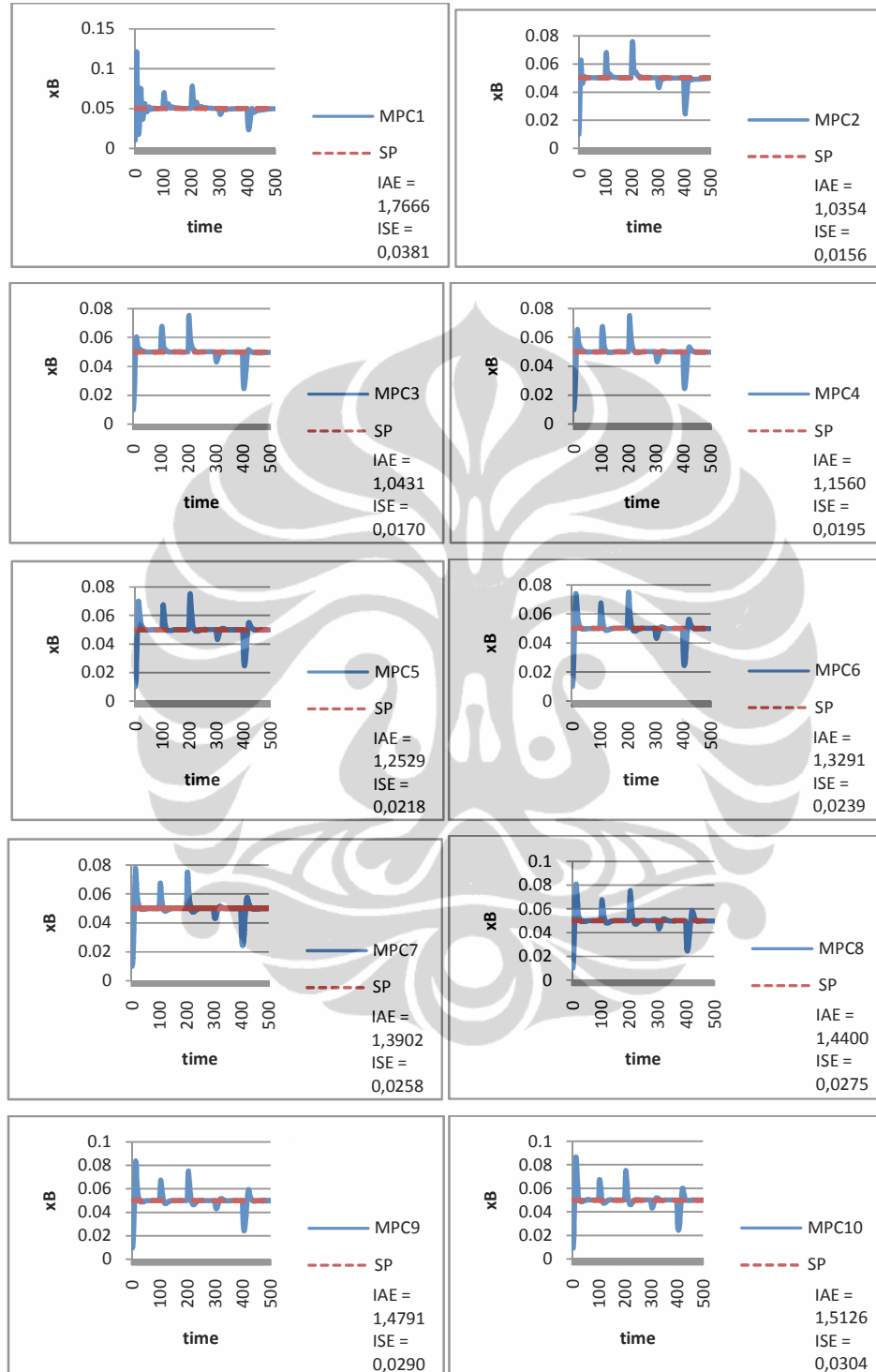
```

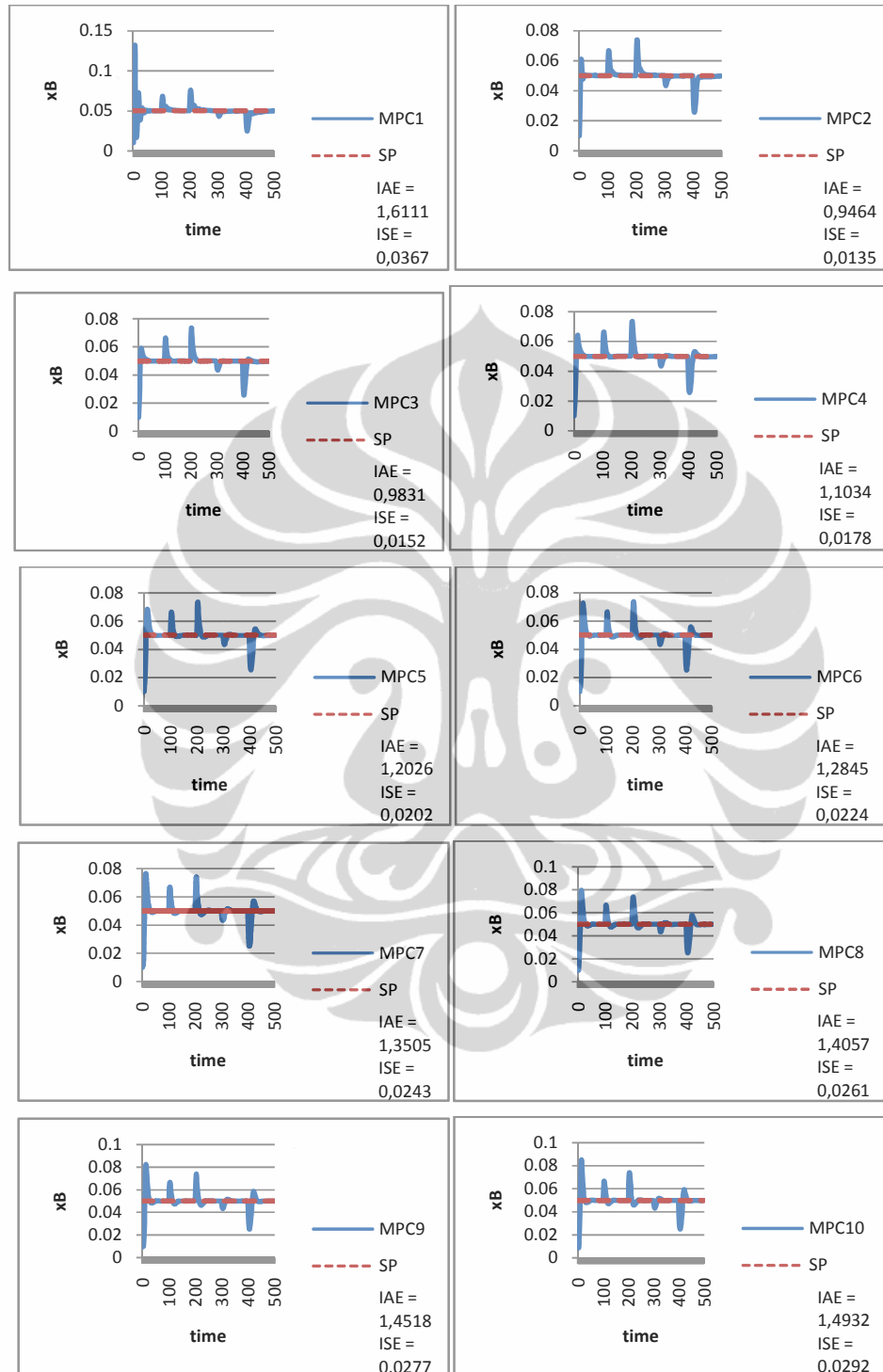


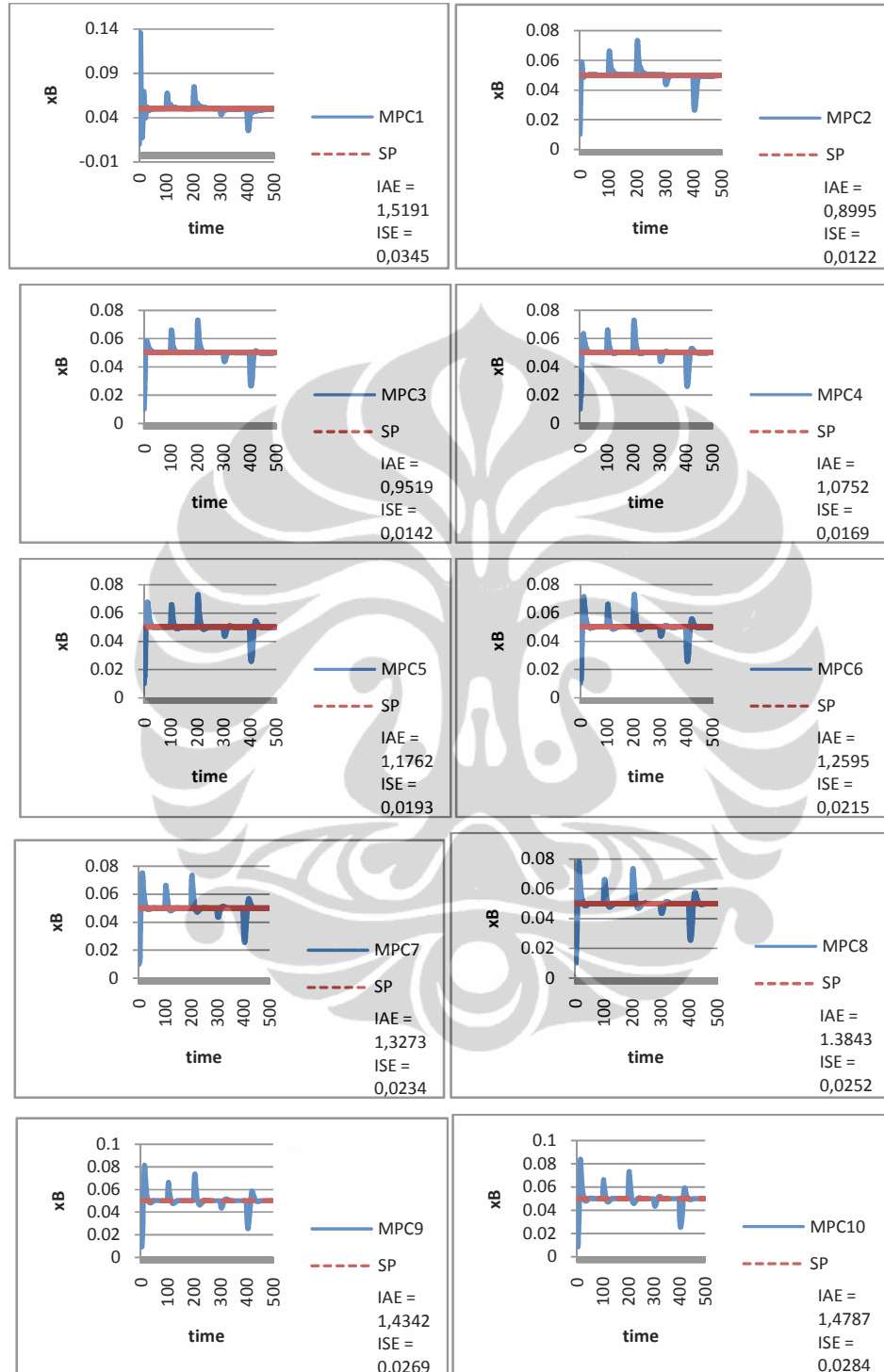
```
plot(t,xB,'g--')
%%
s = xlswrite('SP.xlsx', xB(:,1), 'SP1', 'D2:D301')
s = xlswrite('SP.xlsx', yD(:,1), 'SP1', 'O2:O301')
%%
IAE_MPC2 = ones (1,N)*abs (xB(:,2) - xB(:,1))
ISE_MPC2 = (xB(:,2) - xB(:,1))' * (xB(:,2) - xB(:,1))
```

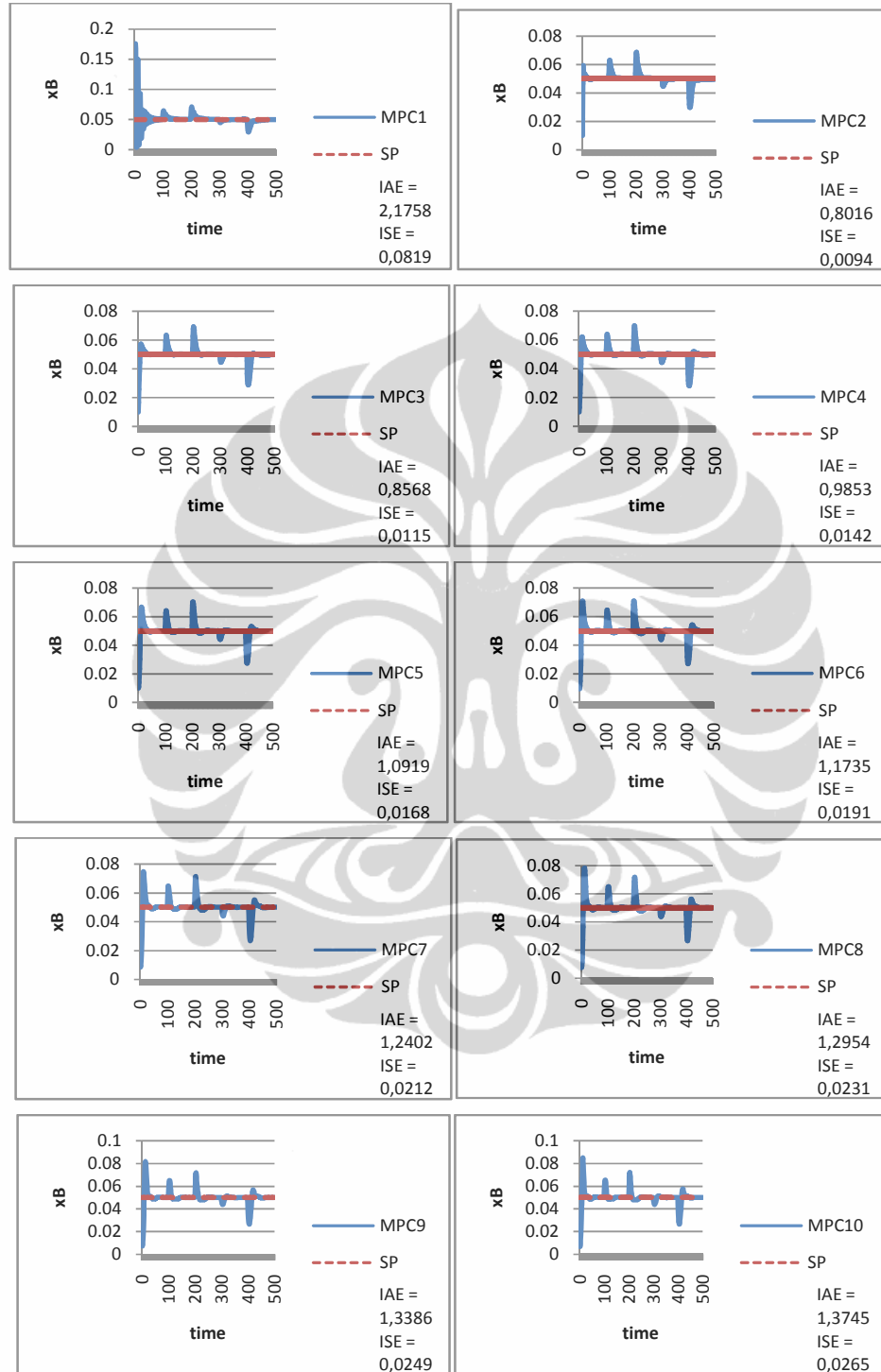


Lampiran 11: Respon  $x_B$  pada  $T_s = 0,72$ ;  $P = 66$ ;  $M = 26$ ; Skenario 2

Lampiran 12: Respon  $x_B$  pada  $T_s = 0,53$ ;  $P = 67$ ;  $M = 27$ ; Skenario 2

Lampiran 13: Respon  $x_B$  pada  $T_s = 0,36$ ;  $P = 69$ ;  $M = 29$ ; Skenario 2

Lampiran 14: Respon  $x_B$  pada  $T_s = 0,29$ ;  $P = 70$ ;  $M = 30$ ; Skenario 2

Lampiran 15: Respon  $x_B$  pada  $T_s = 0,1$ ;  $P = 12$ ;  $M = 2$ ; Skenario 2

## Lampiran 16: Program Simulasi MMPC Dua Model pada Skenario 1

```

if ~mpcchecktoolboxinstalled('simulink')
    disp('Simulink(R) is required to run this demo.')
return
end
if ~mpcchecktoolboxinstalled('scd')
    disp('Simulink Control Design(R) is required to run this
demo.')
return
end
%%
% Set initial states and inputs
yp = [0.99; 0.02];
up = [2.7063; 3.2063; 1; 0.5; 1];
%%
% Compute initial condition
[x0, u0, y0, dx] = trim('cola_inout', [], up, yp, [], [1 2], 2);
%%
% Obtain linear model
[a, b, c, d] = linmod('cola_inout', x0, u0);
%%
% Verify that the linear model is open-loop stable at this
condition
eig(a)
%%
% All eigenvalues are negative so the plant is open-loop stable.
%
% Configure a state-space model for use in MPC
Cola = ss(a, b, c, d);
Cola.InputName = {'LT', 'VB', 'F', 'zF', 'qF'};
Cola.OutputName = {'y_D', 'y_B'};
%%
% Designate the reflux and boil up flow as the manipulated
variable.
Cola.InputGroup.MV = [1,2];
%%
% The other three inputs are unmeasured disturbances.
Cola.InputGroup.UD = [3,4,5];
%%
% Create the MPC model format and set the nominal conditions.
Model.Plant = Cola;
Model.Nominal.X = x0;
Model.Nominal.Y = y0;
Model.Nominal.U = [u0(1);u0(2);0;0;0]; % NOTE: nominal values for
unmeasured disturbance must be zero.

%% Define a MPC Controller
Ts = 0.1; % Sampling time (minutes)
P = 12;
M = 2;
MPC = mpc(Model, Ts,P,M);
%%
% The goal will be to track a specified transition in the bottom
% concentration. The top concentration will be measured and used
in
% state estimation but the controller will not attempt to regulate
it
% directly. It will vary as needed to regulate the bottom

```

```

concentration. Thus,
% set its MPC weight to zero.
MPC.Weights.OV = [1 1];
% Controller blocks in a feedback configuration.
open_system('Cola_MMPC_1')

%%
% Set plot scales and simulate the response.
set_param('Cola_MMPC_1/Measurements','Open','on')
set_param('Cola_MMPC_1/Manipulated Var','Open','on')

%% Modify Controller Design: Increase Estimator Signal-to-Noise
% The controller is stable but its disturbance rejection is
sluggish. Try
% increasing the state estimator signal-to-noise by a factor of 10
and test
% the design again.
%D = ss(getindist(MPC));
%D.b = eye(2)*10;
%set(D,'InputName',[],'OutputName',[],'InputGroup',[],'OutputGroup',
[],[]);
%setindist(MPC,'model',D);
%% Cola_Setpoints
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=0.05;
T(t1:N,:)=0.05;
T(t2:N,:)=0.05;
T(t3:N,:)=0.05;
T(t4:N,:)=0.05;
Cola_Setpoints.time = [1:N]';
Cola_Setpoints.signals.values = [.99*ones(N,1) T(1:N,1)];
%% Cola_Disturbance
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=1.0;
T(t1:N,:)=1.2;
T(t2:N,:)=1.5;
T(t3:N,:)=1.4;
T(t4:N,:)=1.0;
Cola_Disturbance.time = [1:N]';
Cola_Disturbance.signals.values = T(1:N,1);
%%
open_system('Cola_MMPC_1/MPC Controller')
%%
% Simulate model.
sim('Cola_MMPC_1',1:N)
%% Design Another Controller for the Final Operating Condition
yp = [0.99; 0.03];
%%
% Compute initial condition
[x0, u0, y0, dx] = trim('cola_inout', [], up, yp, [], [1 2], 2);

```



```

%%
% Obtain linear model
[a, b, c, d] = linmod('cola_inout', x0, u0);
%%
% Verify that the linear model is open-loop stable at this
condition
eig(a)
%%
% All eigenvalues are negative so the plant is open-loop stable.
%
% Configure a state-space model for use in MPC
Cola = ss(a, b, c, d);
Cola.InputName = {'LT', 'VB', 'F', 'zF', 'qF'};
Cola.OutputName = {'y_D', 'y_B'};
%%
% Designate the coolant temperature as the manipulated variable.
Cola.InputGroup.MV = [1,2];
%%
% The other two inputs are unmeasured disturbances.
Cola.InputGroup.UD = [3,4,5];
%%
% Create the MPC model format and set the nominal conditions.
MPC2 = MPC;
MPC2.Model.Plant = Cola;
Model.Nominal.X = x0;
Model.Nominal.Y = y0;
Model.Nominal.U = [u0(1);u0(2);0;0;0]; % NOTE: nominal values for
unmeasured disturbance must be zero.
%set_param('Cola_MMPC_1/MPC Controller', 'mpcobj', 'MPC2')

% Control the Plant With the Multiple MPC Controllers Block
% The following model uses the Multiple MPC Controllers block
instead of
% the MPC Controller block to implement three MPC controllers
across the
% operating range.
open_system('Cola_MMPC_2')
%%
% Note that it has been configured to use the three controllers in a
% sequence: MPC/MPC5/MPC2.
open_system('Cola_MMPC_2/Multiple MPC Controllers')
%%
% Note also that the two switches specify when to switch from one
% controller to another. The rules are:
% If CSTR concentration >= 8, use MPC
% If 3 <= CSTR concentration < 8, use MPC5
% If CSTR concentration < 3, use MPC2
%%
% Simulate with the Multiple MPC Controllers block
set_param('Cola_MMPC_2/Measurements', 'Open', 'on');
set_param('Cola_MMPC_2/Manipulated Var', 'Open', 'on');
sim('Cola_MMPC_2', 1:N)
IAE_MMPC=ones (1,N)*abs (xB(:,2) - xB(:,1))
ISE_MMPC=(xB(:,2) - xB(:,1))'*(xB(:,2) - xB(:,1))
%%
%s = xlswrite('MMPC.xlsx', xB(:,1), 'SP2', 'L2:L201')
%s = xlswrite('MMPC.xlsx', yD(:,1), 'SP2', 'W2:W201')

```

## Lampiran 17: Program Simulasi MMPC Tiga Model pada Skenario 1

```

if ~mpcchecktoolboxinstalled('simulink')
    disp('Simulink(R) is required to run this demo.')
return
end
if ~mpcchecktoolboxinstalled('scd')
    disp('Simulink Control Design(R) is required to run this
demo.')
return
end
%%
% Set initial states and inputs
yp = [0.99; 0.04];
up = [2.7063; 3.2063; 1; 0.5; 1];
%%
% Compute initial condition
[x0, u0, y0, dx] = trim('cola_inout', [], up, yp, [], [1 2], 2);
%%
% Obtain linear model
[a, b, c, d] = linmod('cola_inout', x0, u0);
%%
% Verify that the linear model is open-loop stable at this
condition
eig(a)
%%
% All eigenvalues are negative so the plant is open-loop stable.
%
% Configure a state-space model for use in MPC
Cola = ss(a, b, c, d);
Cola.InputName = {'LT', 'VB', 'F', 'zF', 'qF'};
Cola.OutputName = {'y_D', 'y_B'};
%%
% Designate the reflux and boil up flow as the manipulated
variable.
Cola.InputGroup.MV = [1,2];
%%
% The other three inputs are unmeasured disturbances.
Cola.InputGroup.UD = [3,4,5];
%%
% Create the MPC model format and set the nominal conditions.
Model.Plant = Cola;
Model.Nominal.X = x0;
Model.Nominal.Y = y0;
Model.Nominal.U = [u0(1);u0(2);0;0;0]; % NOTE: nominal values for
unmeasured disturbance must be zero.

%% Define a MPC Controller
Ts = 0.1; % Sampling time (minutes)
P = 12;
M = 2;
MPC = mpc(Model, Ts,P,M);
%%
% The goal will be to track a specified transition in the bottom
% concentration. The top concentration will be measured and used
in
% state estimation but the controller will not attempt to regulate
it
% directly. It will vary as needed to regulate the bottom

```

```

concentration. Thus,
% set its MPC weight to zero.
MPC.Weights.OV = [1 1];
% Controller blocks in a feedback configuration.
open_system('Cola_MMPC_1')

%%
% Set plot scales and simulate the response.
set_param('Cola_MMPC_1/Measurements','Open','on')
set_param('Cola_MMPC_1/Manipulated Var','Open','on')

%% Modify Controller Design: Increase Estimator Signal-to-Noise
% The controller is stable but its disturbance rejection is
sluggish. Try
% increasing the state estimator signal-to-noise by a factor of 10
and test
% the design again.
%D = ss(getindist(MPC));
%D.b = eye(2)*10;
%set(D,'InputName',[],'OutputName',[],'InputGroup',[],'OutputGroup',
[],[]);
%setindist(MPC,'model',D);
%% Cola_Setpoints
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=0.05;
T(t1:N,:)=0.05;
T(t2:N,:)=0.05;
T(t3:N,:)=0.05;
T(t4:N,:)=0.05;
Cola_Setpoints.time = [1:N]';
Cola_Setpoints.signals.values = [.99*ones(N,1) T(1:N,1)];
%% Cola_Disturbance
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=1.0;
T(t1:N,:)=1.2;
T(t2:N,:)=1.5;
T(t3:N,:)=1.4;
T(t4:N,:)=1.0;
Cola_Disturbance.time = [1:N]';
Cola_Disturbance.signals.values = T(1:N,1);
%%
open_system('Cola_MMPC_1/MPC Controller')
%%
% Simulate model.
sim('Cola_MMPC_1',1:N)
%% Design Another Controller for the Final Operating Condition
yp = [0.99; 0.05];
%%
% Compute initial condition
[x0, u0, y0, dx] = trim('cola_inout', [], up, yp, [], [1 2], 2);

```

```

%%
% Obtain linear model
[a, b, c, d] = linmod('cola_inout', x0, u0);
%%
% Verify that the linear model is open-loop stable at this
condition
eig(a)
%%
% All eigenvalues are negative so the plant is open-loop stable.
%
% Configure a state-space model for use in MPC
Cola = ss(a, b, c, d);
Cola.InputName = {'LT', 'VB', 'F', 'zF', 'qF'};
Cola.OutputName = {'y_D', 'y_B'};
%%
% Designate the coolant temperature as the manipulated variable.
Cola.InputGroup.MV = [1,2];
%%
% The other two inputs are unmeasured disturbances.
Cola.InputGroup.UD = [3,4,5];
%%
% Create the MPC model format and set the nominal conditions.
MPC2 = MPC;
MPC2.Model.Plant = Cola;
Model.Nominal.X = x0;
Model.Nominal.Y = y0;
Model.Nominal.U = [u0(1);u0(2);0;0;0]; % NOTE: nominal values for
unmeasured disturbance must be zero.
%set_param('Cola_MMPC_1/MPC Controller', 'mpcobj', 'MPC2')
% Design Another Controller for the Final Operating Condition
yp = [0.99; 0.06];
%%
% Compute initial condition
[x0, u0, y0, dx] = trim('cola_inout', [], up, yp, [], [1 2], 2);
%%
% Obtain linear model
[a, b, c, d] = linmod('cola_inout', x0, u0);
%%
% Verify that the linear model is open-loop stable at this
condition
eig(a)
%%
% All eigenvalues are negative so the plant is open-loop stable.
%
% Configure a state-space model for use in MPC
Cola = ss(a, b, c, d);
Cola.InputName = {'LT', 'VB', 'F', 'zF', 'qF'};
Cola.OutputName = {'y_D', 'y_B'};
%%
% Designate the coolant temperature as the manipulated variable.
Cola.InputGroup.MV = [1,2];
%%
% The other two inputs are unmeasured disturbances.
Cola.InputGroup.UD = [3,4,5];
%%
% Create the MPC model format and set the nominal conditions.
MPC3 = MPC;
MPC3.Model.Plant = Cola;
Model.Nominal.X = x0;

```

```

Model.Nominal.Y = y0;
Model.Nominal.U = [u0(1);u0(2);0;0;0]; % NOTE: nominal values for
unmeasured disturbance must be zero.
%% Control the Plant With the Multiple MPC Controllers Block
% The following model uses the Multiple MPC Controllers block
instead of
% the MPC Controller block to implement three MPC controllers
across the
% operating range.
open_system('Cola_MMPC_3')
%%
% Note that it has been configured to use the three controllers in
a
% sequence: MPC/MPC5/MPC2.
open_system('Cola_MMPC_3/Multiple MPC Controllers')
%%
% Note also that the two switches specify when to switch from one
% controller to another. The rules are:
% If CSTR concentration >= 8, use MPC
% If 3 <= CSTR concentration < 8, use MPC5
% If CSTR concentration < 3, use MPC2
%%
% Simulate with the Multiple MPC Controllers block
set_param('Cola_MMPC_3/Measurements','Open','on');
set_param('Cola_MMPC_3/Manipulated Var','Open','on');
sim('Cola_MMPC_3',1:N)
t=1:N;
figure(4)
subplot 211
plot(t,yD,'r')
subplot 212
plot(t,xB)
IAE_MMPC=ones(1,N)*abs(xB(:,2) - xB(:,1))
ISE_MMPC=(xB(:,2) - xB(:,1))'*(xB(:,2) - xB(:,1))
%%
%s = xlswrite('SingleMPC.xlsx', xB(:,2), 'MMPC', 'B2:B201')
%s = xlswrite('SingleMPC.xlsx', xB(:,1), 'MMPC', 'C2:C201')
%s = xlswrite('SingleMPC.xlsx', yD(:,1), 'MMPC', 'N2:N201')

```

Lampiran 18: Respon  $x_B$  pada MMPC Dua Model; Skenario 1

MPC2.3			MPC2.4		
switch	IAE	ISE	switch	IAE	ISE
0.01	0.2893	0.0041	0.01	0.3498	0.0050
0.02	0.2854	0.0039	0.02	0.3409	0.0046
0.03	0.2804	0.0038	0.03	0.3332	0.0044
0.04	0.2733	0.0037	0.04	0.3205	0.0041
0.05	0.2582	0.0036	0.05	0.2890	0.0039
0.06	0.2584	0.0036	0.06	0.2606	0.0037
MPC2.5			MPC2.6		
switch	IAE	ISE	switch	IAE	ISE
0.01	0.3983	0.0058	0.01	0.4357	0.0065
0.02	0.3851	0.0053	0.02	0.4190	0.0058
0.03	0.3743	0.0049	0.03	0.4062	0.0054
0.04	0.3549	0.0045	0.04	0.3845	0.0048
0.05	0.3191	0.0043	0.05	0.3474	0.0047
0.06	0.2624	0.0038	0.06	0.2639	0.0038
MPC3.4			MPC3.5		
switch	IAE	ISE	switch	IAE	ISE
0.01	0.4024	0.0062	0.01	0.4433	0.0070
0.02	0.3973	0.0059	0.02	0.4342	0.0065
0.03	0.3908	0.0057	0.03	0.4238	0.0061
0.04	0.3857	0.0056	0.04	0.4168	0.0060
0.05	0.3714	0.0056	0.05	0.3921	0.0059
0.06	0.3524	0.0053	0.06	0.3543	0.0053
MPC3.6			MPC4.5		
switch	IAE	ISE	switch	IAE	ISE
0.01	0.4765	0.0077	0.01	0.4879	0.0083
0.02	0.4634	0.0071	0.02	0.4832	0.0081
0.03	0.4505	0.0065	0.03	0.4805	0.0080
0.04	0.4432	0.0064	0.04	0.4790	0.0080
0.05	0.4093	0.0063	0.05	0.4681	0.0080
0.06	0.3559	0.0054	0.06	0.4624	0.0078
MPC4.6			MPC5.6		
switch	IAE	ISE	switch	IAE	ISE
0.01	0.5171	0.0089	0.01	0.5553	0.0103
0.02	0.5086	0.0086	0.02	0.5535	0.0102
0.03	0.5042	0.0084	0.03	0.5524	0.0102
0.04	0.5024	0.0084	0.04	0.5520	0.0102
0.05	0.4825	0.0083	0.05	0.5433	0.0102
0.06	0.4742	0.0082	0.06	0.5434	0.0101

Lampiran 19: Respon  $x_B$  pada MMPC Tiga Model; Skenario 1

MPC2.3.4				MPC2.3.5			
switch1	switch2	IAE	ISE	switch1	switch2	IAE	ISE
0.01	0.02	0.3448	0.0048	0.01	0.02	0.3889	0.0054
	0.03	0.3389	0.0046		0.03	0.3788	0.0051
	0.04	0.3309	0.0044		0.04	0.3660	0.0048
	0.05	0.3140	0.0043		0.05	0.3397	0.0047
	0.06	0.2915	0.0041		0.06	0.2934	0.0042
0.02	0.03	0.3365	0.0045	0.02	0.03	0.3770	0.0050
	0.04	0.3279	0.0043		0.04	0.3625	0.0046
	0.05	0.3104	0.0042		0.05	0.3361	0.0045
	0.06	0.2876	0.0040		0.06	0.2895	0.0041
0.03	0.04	0.3239	0.0042	0.03	0.04	0.3586	0.0045
	0.05	0.3057	0.0041		0.05	0.3315	0.0044
	0.06	0.2827	0.0039		0.06	0.2846	0.0039
0.04	0.05	0.2991	0.0040	0.04	0.05	0.3248	0.0043
	0.06	0.2756	0.0038		0.06	0.2775	0.0038
0.05	0.06	0.2605	0.0037	0.05	0.06	0.2624	0.0038
MPC2.3.6				MPC2.4.5			
switch1	switch2	IAE	ISE	switch1	switch2	IAE	ISE
0.01	0.02	0.4228	0.0059	0.01	0.02	0.3931	0.0056
	0.03	0.4096	0.0055		0.03	0.3877	0.0054
	0.04	0.3947	0.0051		0.04	0.3835	0.0053
	0.05	0.3622	0.0050		0.05	0.3712	0.0053
	0.06	0.2950	0.0042		0.06	0.3518	0.0050
0.02	0.03	0.4084	0.0054	0.02	0.03	0.3796	0.0051
	0.04	0.3914	0.0050		0.04	0.3746	0.0050
	0.05	0.3586	0.0049		0.05	0.3625	0.0049
	0.06	0.2911	0.0041		0.06	0.3429	0.0047
0.03	0.04	0.3873	0.0049	0.03	0.04	0.3664	0.0047
	0.05	0.3538	0.0048		0.05	0.3550	0.0047
	0.06	0.2862	0.0040		0.06	0.3352	0.0044
0.04	0.05	0.3468	0.0047	0.04	0.05	0.3428	0.0045
	0.06	0.2791	0.0039		0.06	0.3225	0.0042
0.05	0.06	0.2639	0.0038	0.05	0.06	0.2911	0.0040

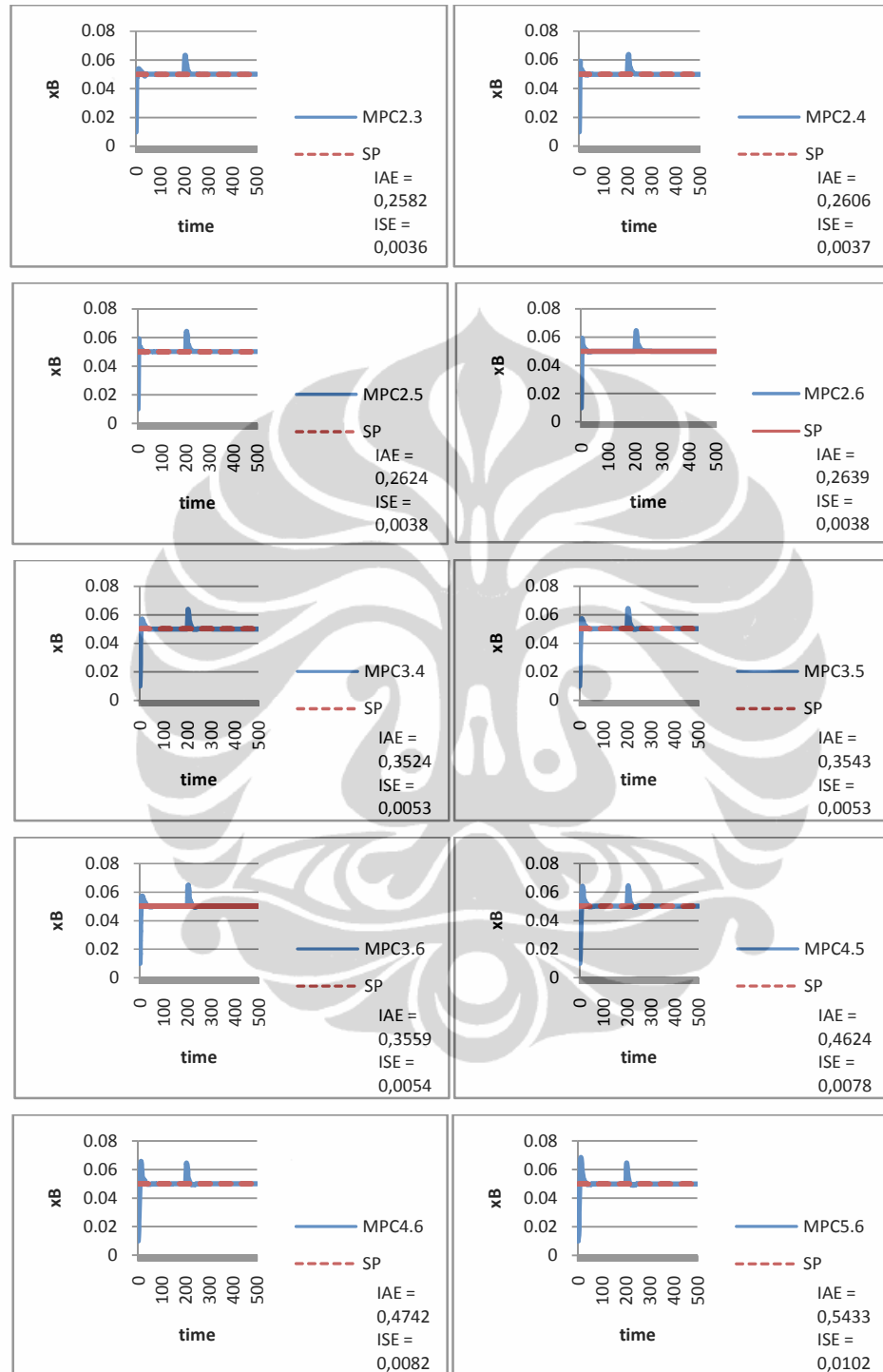
Lampiran 19 (lanjutan): Respon  $x_B$  pada MMPC Tiga Model; Skenario 1

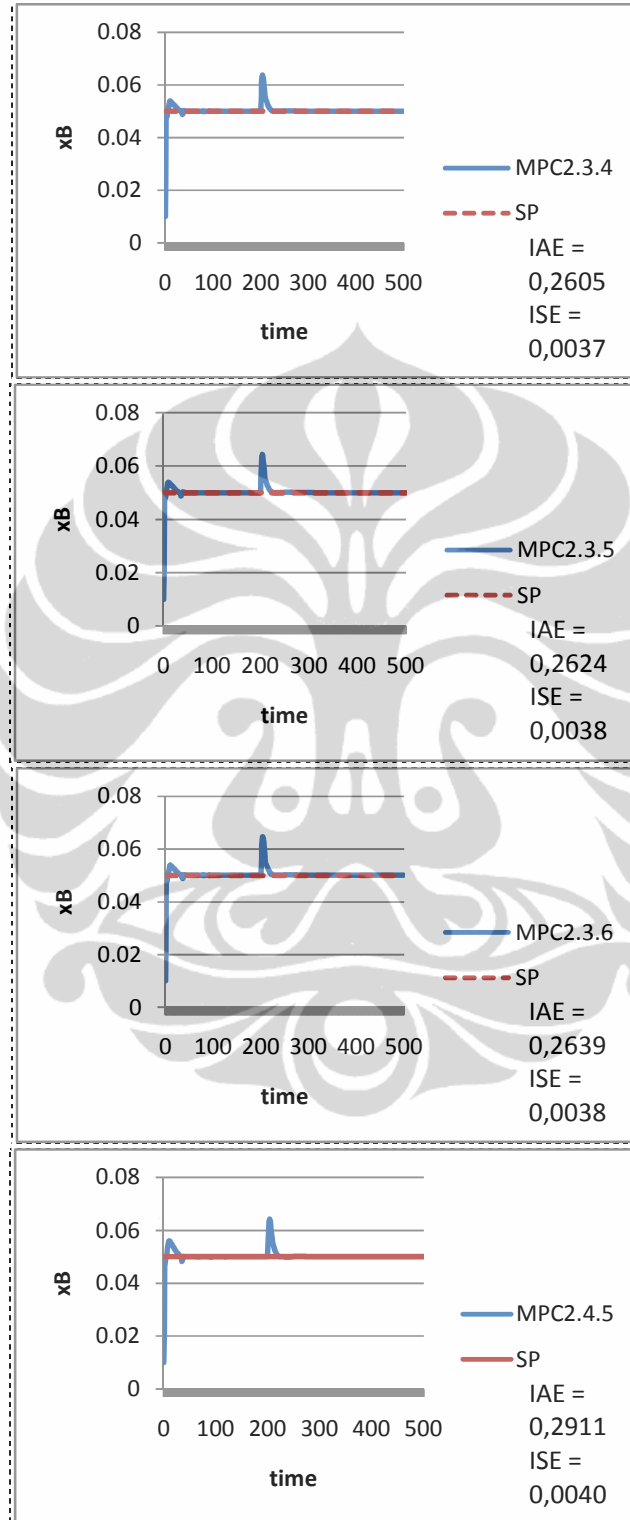
MPC2.4.6				MPC2.5.6			
switch1	switch2	IAE	ISE	switch1	switch2	IAE	ISE
0.01	0.02	0.4264	0.0061	0.01	0.02	0.4311	0.0063
	0.03	0.4175	0.0058		0.03	0.4272	0.0062
	0.04	0.4113	0.0056		0.04	0.4257	0.0061
	0.05	0.3908	0.0056		0.05	0.4159	0.0061
	0.06	0.3534	0.0051		0.06	0.3999	0.0058
0.02	0.03	0.4096	0.0055	0.02	0.03	0.4140	0.0056
	0.04	0.4023	0.0053		0.04	0.4126	0.0056
	0.05	0.3821	0.0053		0.05	0.4029	0.0056
	0.06	0.3446	0.0047		0.06	0.3867	0.0053
0.03	0.04	0.3951	0.0051	0.03	0.04	0.4017	0.0053
	0.05	0.3747	0.0050		0.05	0.3924	0.0052
	0.06	0.3368	0.0045		0.06	0.3759	0.0050
0.04	0.05	0.3628	0.0048	0.04	0.05	0.3738	0.0048
	0.06	0.3241	0.0042		0.06	0.3566	0.0045
0.05	0.06	0.2927	0.0040	0.05	0.06	0.2639	0.0038
MPC3.4.5				MPC3.4.6			
switch1	switch2	IAE	ISE	switch1	switch2	IAE	ISE
0.01	0.02	0.4384	0.0067	0.01	0.02	0.4670	0.0072
	0.03	0.4345	0.0066		0.03	0.4609	0.0070
	0.04	0.4325	0.0065		0.04	0.4585	0.0069
	0.05	0.4213	0.0065		0.05	0.4379	0.0069
	0.06	0.4044	0.0062		0.06	0.4060	0.0063
0.02	0.03	0.4295	0.0064	0.02	0.03	0.4559	0.0067
	0.04	0.4274	0.0063		0.04	0.4534	0.0067
	0.05	0.4163	0.0063		0.05	0.4328	0.0067
	0.06	0.3993	0.0060		0.06	0.4009	0.0060
0.03	0.04	0.4212	0.0061	0.03	0.04	0.4474	0.0065
	0.05	0.4101	0.0061		0.05	0.4268	0.0064
	0.06	0.3929	0.0058		0.06	0.3945	0.0058
0.04	0.05	0.4053	0.0060	0.04	0.05	0.4224	0.0063
	0.06	0.3877	0.0056		0.06	0.3893	0.0057
0.05	0.06	0.3735	0.0056	0.05	0.06	0.3751	0.0057

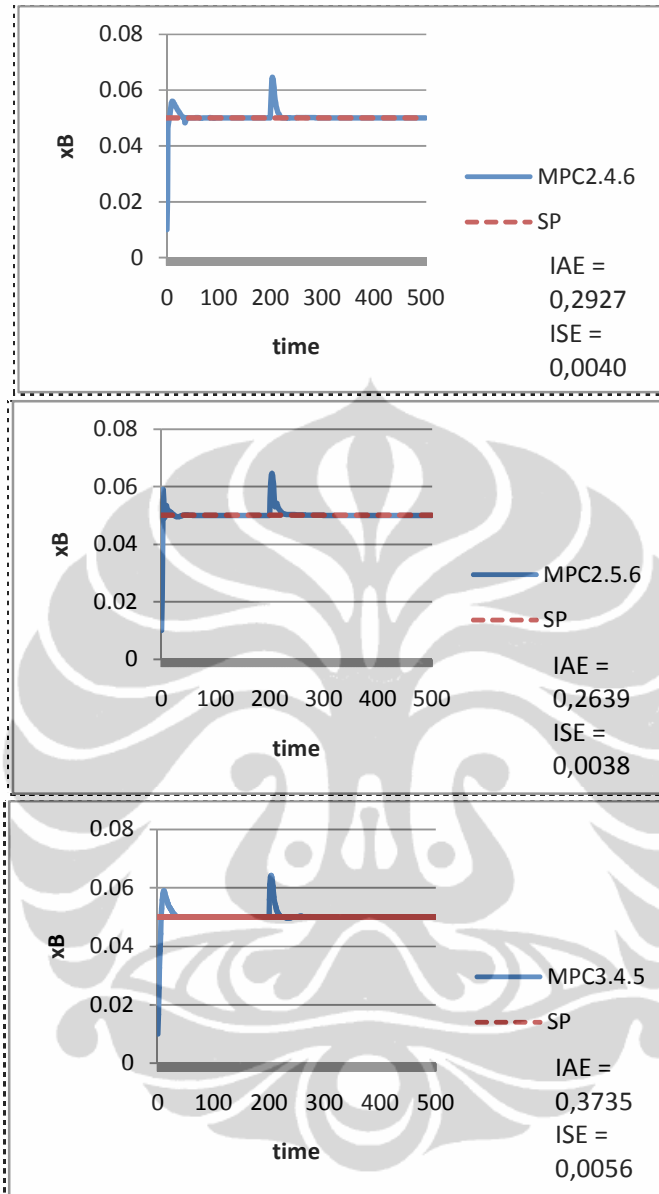


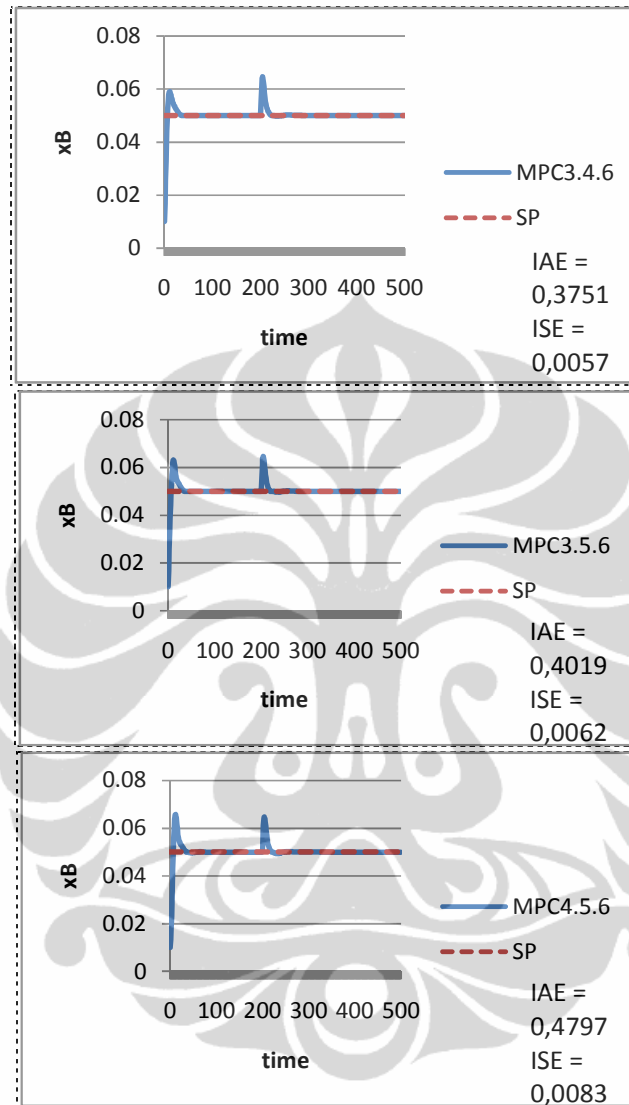
Lampiran 19 (lanjutan): Respon  $x_B$  pada MMPC Tiga Model; Skenario 1

MPC3.5.6				MPC4.5.6			
switch1	switch2	IAE	ISE	switch1	switch2	IAE	ISE
0.01	0.02	0.4715	0.0074	0.01	0.02	0.5133	0.0088
	0.03	0.4693	0.0074		0.03	0.5116	0.0087
	0.04	0.4687	0.0073		0.04	0.5110	0.0087
	0.05	0.4591	0.0073		0.05	0.5019	0.0087
	0.06	0.4529	0.0072		0.06	0.4994	0.0086
0.02	0.03	0.4603	0.0069	0.02	0.03	0.5069	0.0085
	0.04	0.4598	0.0069		0.04	0.5063	0.0085
	0.05	0.4501	0.0069		0.05	0.4972	0.0085
	0.06	0.4438	0.0068		0.06	0.4949	0.0084
0.03	0.04	0.4497	0.0065	0.03	0.04	0.5037	0.0084
	0.05	0.4400	0.0065		0.05	0.4946	0.0084
	0.06	0.4336	0.0064		0.06	0.4921	0.0083
0.04	0.05	0.4336	0.0063	0.04	0.05	0.4934	0.0084
	0.06	0.4268	0.0062		0.06	0.4908	0.0083
0.05	0.06	0.4019	0.0062	0.05	0.06	0.4797	0.0083

Lampiran 20: Respon  $x_B$  pada MMPC Dua Model; Skenario 1

Lampiran 21: Respon  $x_B$  pada MMPC Tiga Model; Skenario 1Lampiran 21(lanjutan): Respon  $x_B$  pada MMPC Tiga Model; Skenario 1



Lampiran 21(lanjutan): Respon  $x_B$  pada MMPC Tiga Model; Skenario 1

## Lampiran 22: Program Simulasi MMPC Dua Model pada Skenario 2

```

if ~mpcchecktoolboxinstalled('simulink')
    disp('Simulink(R) is required to run this demo.')
return
end
if ~mpcchecktoolboxinstalled('scd')
    disp('Simulink Control Design(R) is required to run this
demo.')
return
end
%%
% Set initial states and inputs
yp = [0.99; 0.02];
up = [2.7063; 3.2063; 1; 0.5; 1];
%%
% Compute initial condition
[x0, u0, y0, dx] = trim('cola_inout', [], up, yp, [], [1 2], 2);
%%
% Obtain linear model
[a, b, c, d] = linmod('cola_inout', x0, u0);
%%
% Verify that the linear model is open-loop stable at this
condition
eig(a)
%%
% All eigenvalues are negative so the plant is open-loop stable.
%
% Configure a state-space model for use in MPC
Cola = ss(a, b, c, d);
Cola.InputName = {'LT', 'VB', 'F', 'zF', 'qF'};
Cola.OutputName = {'y_D', 'y_B'};
%%
% Designate the reflux and boil up flow as the manipulated
variable.
Cola.InputGroup.MV = [1,2];
%%
% The other three inputs are unmeasured disturbances.
Cola.InputGroup.UD = [3,4,5];
%%
% Create the MPC model format and set the nominal conditions.
Model.Plant = Cola;
Model.Nominal.X = x0;
Model.Nominal.Y = y0;
Model.Nominal.U = [u0(1);u0(2);0;0;0]; % NOTE: nominal values for
unmeasured disturbance must be zero.

%% Define a MPC Controller
Ts = 0.1; % Sampling time (minutes)
P = 12;
M = 2;
MPC = mpc(Model, Ts,P,M);
%%
% The goal will be to track a specified transition in the bottom
% concentration. The top concentration will be measured and used
in
% state estimation but the controller will not attempt to regulate
it
% directly. It will vary as needed to regulate the bottom

```

```

concentration. Thus,
% set its MPC weight to zero.
MPC.Weights.OV = [1 1];
% Controller blocks in a feedback configuration.
open_system('Cola_MMPC_1')

%%
% Set plot scales and simulate the response.
set_param('Cola_MMPC_1/Measurements','Open','on')
set_param('Cola_MMPC_1/Manipulated Var','Open','on')

%% Modify Controller Design: Increase Estimator Signal-to-Noise
% The controller is stable but its disturbance rejection is
sluggish. Try
% increasing the state estimator signal-to-noise by a factor of 10
and test
% the design again.
%D = ss(getindist(MPC));
%D.b = eye(2)*10;
%set(D,'InputName',[],'OutputName',[],'InputGroup',[],'OutputGroup',
[],[]);
%setindist(MPC,'model',D);
%% Cola_Setpoints
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=0.05;
T(t1:N,:)=0.05;
T(t2:N,:)=0.05;
T(t3:N,:)=0.05;
T(t4:N,:)=0.05;
Cola_Setpoints.time = [1:N]';
Cola_Setpoints.signals.values = [.99*ones(N,1) T(1:N,1)];
%% Cola_Disturbance
N=500;
t1=100;
t2=200;
t3=300;
t4=400;
T(1:N,:)=1.0;
T(t1:N,:)=1.2;
T(t2:N,:)=1.5;
T(t3:N,:)=1.4;
T(t4:N,:)=1.0;
Cola_Disturbance.time = [1:N]';
Cola_Disturbance.signals.values = T(1:N,1);
%%
open_system('Cola_MMPC_1/MPC Controller')
%%
% Simulate model.
sim('Cola_MMPC_1',1:N)
%% Design Another Controller for the Final Operating Condition
yp = [0.99; 0.03];
%%
% Compute initial condition
[x0, u0, y0, dx] = trim('cola_inout', [], up, yp, [], [1 2], 2);

```

```

%%
% Obtain linear model
[a, b, c, d] = linmod('cola_inout', x0, u0);
%%
% Verify that the linear model is open-loop stable at this
condition
eig(a)
%%
% All eigenvalues are negative so the plant is open-loop stable.
%
% Configure a state-space model for use in MPC
Cola = ss(a, b, c, d);
Cola.InputName = {'LT', 'VB', 'F', 'zF', 'qF'};
Cola.OutputName = {'y_D', 'y_B'};
%%
% Designate the coolant temperature as the manipulated variable.
Cola.InputGroup.MV = [1,2];
%%
% The other two inputs are unmeasured disturbances.
Cola.InputGroup.UD = [3,4,5];
%%
% Create the MPC model format and set the nominal conditions.
MPC2 = MPC;
MPC2.Model.Plant = Cola;
Model.Nominal.X = x0;
Model.Nominal.Y = y0;
Model.Nominal.U = [u0(1);u0(2);0;0;0]; % NOTE: nominal values for
unmeasured disturbance must be zero.
%set_param('Cola_MMPC_1/MPC Controller', 'mpcobj', 'MPC2')

% Control the Plant With the Multiple MPC Controllers Block
% The following model uses the Multiple MPC Controllers block
instead of
% the MPC Controller block to implement three MPC controllers
across the
% operating range.
open_system('Cola_MMPC_2')
%%
% Note that it has been configured to use the three controllers in a
% sequence: MPC/MPC5/MPC2.
open_system('Cola_MMPC_2/Multiple MPC Controllers')
%%
% Note also that the two switches specify when to switch from one
% controller to another. The rules are:
% If CSTR concentration >= 8, use MPC
% If 3 <= CSTR concentration < 8, use MPC5
% If CSTR concentration < 3, use MPC2
%%
% Simulate with the Multiple MPC Controllers block
set_param('Cola_MMPC_2/Measurements', 'Open', 'on');
set_param('Cola_MMPC_2/Manipulated Var', 'Open', 'on');
sim('Cola_MMPC_2', 1:N)
IAE_MMPC=ones (1,N)*abs (xB(:,2) - xB(:,1))
ISE_MMPC=(xB(:,2) - xB(:,1))'*(xB(:,2) - xB(:,1))
%%
%s = xlswrite('MMPC.xlsx', xB(:,1), 'SP2', 'L2:L201')
%s = xlswrite('MMPC.xlsx', yD(:,1), 'SP2', 'W2:W201')

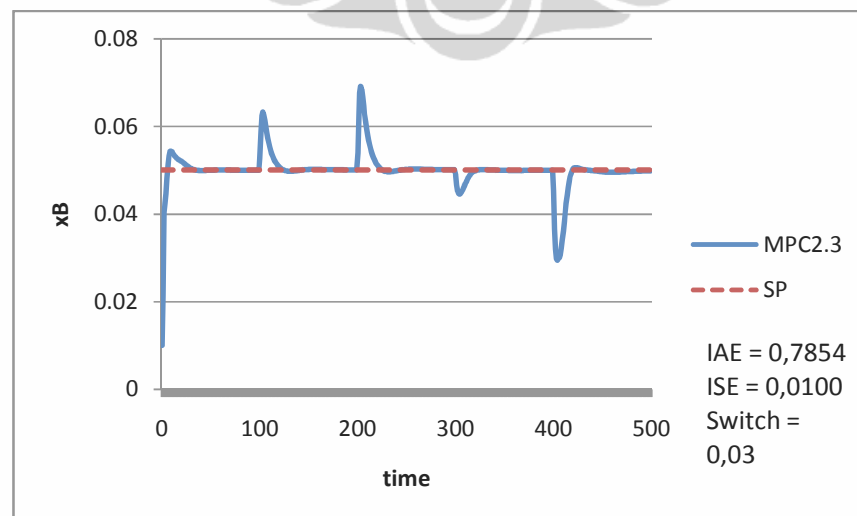
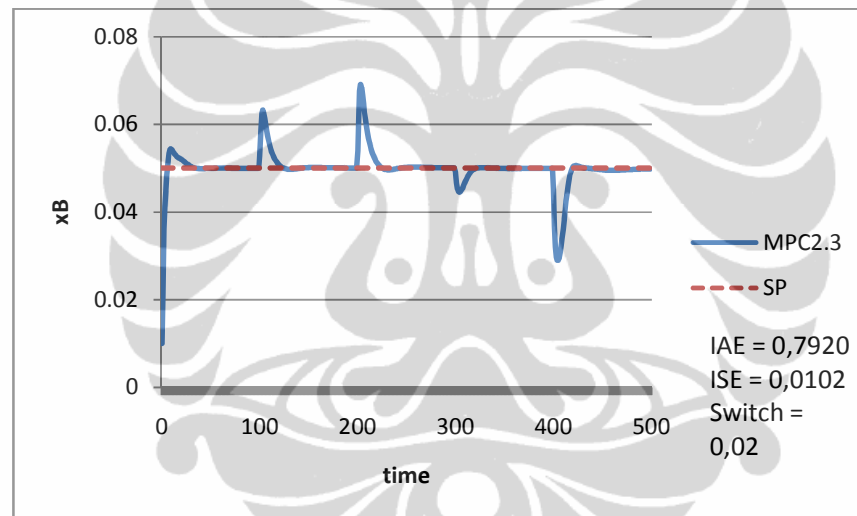
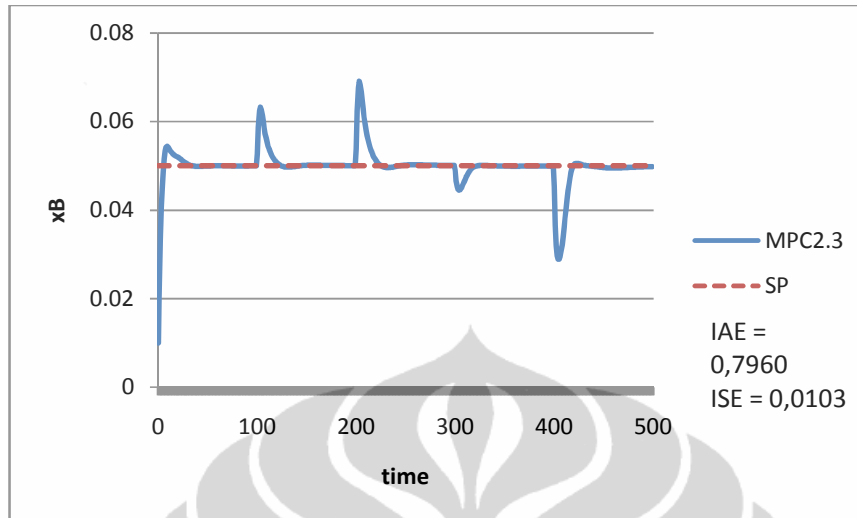
```



Lampiran 23: Respon  $x_B$  pada MMPC Dua Model; Skenario 2

MPC2.3		
switch	IAE	ISE
0.01	0.7960	0.0103
0.02	0.7920	0.0102
0.03	0.7854	0.0100
0.04	0.7730	0.0096
0.05	0.7864	0.0096
0.06	0.8082	0.0096



Lampiran 24: Respon  $x_B$  pada MMPC Dua Model; Skenario 2

Lampiran 24(lanjutan): Respon  $x_B$  pada MMPC Dua Model; Skenario 2