PENGENDALIAN DERAU SECARA AKTIF MENGGUNAKAN ALGORITMA FUZI ADAPTIF

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh :
PRADANA WAHYU ILLAHI
NIM. 0210630097

DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN ELEKTRO
MALANG
2009

repository.ub.ac
UNIVERSITAS BRAWIJAYA

PENGENDALIAN DERAU SECARA AKTIF MENGGUNAKAN ALGORITMA FUZI ADAPTIF

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
PRADANA WAHYU ILLAHI
NIM. 0210630097

Telah diperiksa dan disetujui oleh Dosen Pembimbing:

Pembimbing I	Pembimbing II
Goegoes Dwi N., ST, MT	**Ir. Retnowati, MT**
NIP. 132 318 316	**NIP. 131 124 656**

repository.ub.ac
UNIVERSITAS BRAWIJAYA

PENGENDALIAN DERAU SECARA AKTIF MENGGUNAKAN ALGORITMA FUZI ADAPTIF

Disusun oleh:

PRADANA WAHYU ILLAHI

NIM. 0210630097

DOSEN PENGUJI

Ir. Bambang Siswojo, MT

NIP. 131 759 588

Ir. Erni Yudaningtyas, MT

NIP. 131 879 035

Rusmi Ambarwati, ST., MT

NIP. 132 258 188

Mengetahui,

Ketua Jurusan Teknik Elektro

Ir. Heru Nurwasito, M.Kom.

NIP. 131 879 033

KATA PENGANTAR

Segala puji Allah SWT yang telah memberikan hidayah sehingga penyusun berhasil menyelesaikan skripsi ini. Karena hanya dengan limpahan rahmat dan karunia-Nya semata maka skripsi ini dapat terselesaikan. Skripsi dengan judul "Pengendalian derau secara aktif menggunakan algoritma fuzi adaptif" ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik di Fakultas Teknik Jurusan Teknik Elektro Universitas Brawijaya.

Teriring ucapan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu terlaksananya penyusunan skripsi ini hingga dapat terselesaikan dengan baik, antara lain :

1. Bapak Ir. Heru Nurwarsito., M.Kom, selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.
2. Bapak Rudy Yuwono., ST. MSc. selaku Sekretaris Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.
3. Bapak Ir. Purwanto M.T, selaku Ketua KKDK Teknik Kontrol Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang sebagai dosen pembimbing yang telah banyak memberikan masukan dalam penyusunan skripsi ini.
4. Bapak Goegoes Dwi Nusantoro ST., MT, selaku dosen pembimbing yang telah banyak memberikan memberikan saran yang bermanfaat bagi kelancaran penyusunan skripsi ini.
5. Ibu Ir. Retnowati MT selaku dosen pembimbing yang telah banyak memberikan masukan dalam penyusunan skripsi ini.
6. Segenap karyawan pengajaran dan *recording* Teknik Elektro Universitas Brawijaya.
7. Bapak Mochtar Bashori dan Ibu Sapti Sutris Nani beserta Adikku Musa Wahyu Pangeran yang telah memberikan segalanya tanpa pamrih, semoga Allah SWT. membalasnya.
8. istriku tersayang Yuni Astutik dan putraku Amjad Jundi Arrahman, terimakasih atas segala semangat dan kasih sayangnya.

9. Rekan angkatan Teknik Elektro 2002, terima kasih atas waktu dan perjuangan yang telah kalian torehkan.
10. Dan semua pihak yang tidak bisa penyusun sebutkan satu persatu.

Akhirnya dengan segala kerendahan hati, penyusun amat menyadari bahwa skripsi ini masih teramat jauh untuk dikatakan sempurna. Oleh karena itu, saran maupun kritik sangat penyusun harapkan untuk memperbaiki skripsi ini. Namun, penyusun tetap berharap agar skripsi ini akan dapat bermanfaat untuk kita semua.

Malang, 05 Agustus 2009

Penyusun

DAFTAR ISI

PENGANTAR i
DAFTAR ISI iii
DAFTAR TABEL v
DAFTAR GAMBAR vi
ABSTRAK vii
BAB I. PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Ruang Lingkup Pembahasan 3
1.4 Tujuan Penulisan 3
1.5 Sistematika Penulisan 4

BAB II. DASAR TEORI 5
2.1 Konsep dasar Actine Noise Control (ANC) 5
2.2 Pengaruh Lintasan Sekunder 8
2.3 Sinyal-sinyal Noise 9
2.3.1 Derau Pita Lebar 9
2.3.2 Derau Pita Sempit 10
2.4 Perambatan Gelombang Akustik 10
2.5 Sistem ANC menggunakan algoritma filtered-x 11
2.6 Himpunan Fuzi 13
2.7 Sistem Inferensi Fuzi 16
2.8 Fuzi adaptif filtered-x dalam sistem ANC 17
2.9 Bahasa Komputasi Teknik MATLAB 23
2.10 Sound Card 23
2.11 Metode *s-functions* 24

BAB III. METODE PENELITIAN …….. 26

3.1 Pengambilan Data *Sample* Noise……..……... 26

3.2 Pemodelan Matematik Sistem……. 27

3.3 Pengujian Kinerja Sistem……..…… 28

3.4 Pengambilan Keputusan…… ……. 28

BAB IV. PEMODELAN SISTEM ANC…… 29

4.1 Penentuan Model Sistem ANC…… 29

4.2 Algoritma FxLMS untuk sistem ANC…… 31

4.3 Algoritma Fuzi Adaptif untuk sistem ANC …… 43

BAB V. HASIL PENGUJIAN……. 40

5.1 Pengujian Fuzi Adaptif……... 40

5.2 Pengujian Fuzi Adaptif filtered-x untuk sistem ANC..... 42

BAB VI. KESIMPULAN DAN SARAN …….. 44

5.1 Kesimpulan……..….... 44

5.2 Saran……..... 44

DAFTAR PUSTAKA ……..……. 45

LAMPIRAN

DAFTAR TABEL

No.	Judul	Halaman
Tabel 1	Aturan fuzzy pada sistem ANC ……	36

DAFTAR GAMBAR

No.	Judul	Halaman
Gambar 2.1	Sistem ANC secara umum ……	6
Gambar 2.2	Sistem ANC kanal tunggal dalam saluran udara ……	7
Gambar 2.3	Sistem Identifikasi ANC ……	8
Gambar 2.4	Pengaruh Lintasan Sekunder……..	8
Gambar 2.5	Jalur Sekunder sistem ANC ……..	9
Gambar 2.6	Blok diagram ANC berbasis filtered-x LMS……..	12
Gambar 2.7	Contoh himpunan fuzi ……..……..	14
Gambar 2.8	Operasi pada himpunan fuzi ……..……..	15
Gambar 2.9	Sistem inferensi fuzi……..……..	16
Gambar 2.10	Blok diagram ANC berbasis fuzi adaptif…..……..	19
Gambar 2.11	Hubungan blok *s-functions*, dialog box, dan source...	25
Gambar 3.1	Desain alat pengujian……..……..	26
Gambar 3.2	Sistem ANC pada suatu saluran (duct)……..……..	27
Gambar 4.1	Tanggapan Magnitude dan fasa dari model speaker...	29
Gambar 4.2	Realisasi model tapis digital orde 2……..……..	31
Gambar 4.3	Diagram blok system ANC tapis FIR FxLMS……..	31
Gambar 4.4	Diagram blok system ANC dengan Simulink……..	33
Gambar 4.5	Penambahan bagian koreksi pada blok LMS……..	33
Gambar 4.6	Skema umum masukan-keluaran pada fuzi adaptif..	35
Gambar 4.7	Blok diagram penggunaan fuzi adaptif……..……..	39
Gambar 5.1	Hasil pengujian variasi frekuensi 50Hz……..……..	40
Gambar 5.2	Hasil pengujian variasi frekuensi 100Hz…..…….	40
Gambar 5.3	Hasil pengujian variasi frekuensi 150Hz…..……	41
Gambar 5.4	Hasil pengujian variasi frekuensi 250Hz…….	41

Gambar 5.5	Hasil pengujian variasi frekuensi 500Hz…….	41
Gambar 5.6	Blok diagram fuzi adaptif sebuah duct ANC……..	42
Gambar 5.7	Kotak dialog saat eksekusi fuzi adaptif ANC …….	43

repository.ub.ac
UNIVERSITAS BRAWIJAYA

ABSTRAK

PRADANA WAHYU ILLAHI, Jurusan Elektro, Fakultas Teknik Universitas Brawijaya, Agustus 2009, " Pengendali Derau Secara Aktif menggunakan Algoritma *Fuzzy Adaptif*", Dosen Pembimbing : Goegoes Dwi Nusantoro, ST., MT dan Ir. Retnowati, MT.

Abstraksi

Teknik pengolahan sinyal audio digital saat ini telah mengalami perkembangan yang sangat pesat. Pengolahan sinyal seperti transmisi, perekaman, penapisan, kompresi dan ekspansi dapat dikerjakan dengan menghasilkan distorsi yang kecil dan dengan SNR (*signal to noise ratio*) yang lebih baik daripada pemrosesan sinyal analog. Salah satu teknik tersebut adalah sistem ANC (*Active Noise Control*) yang menggunakan sinyal buatan (*artificial signal*) dengan besar yang sama serta fasanya berlawanan untuk menghapuskan derau atau bunyi yang tidak diinginkan.

Beberapa sistem ANC menggunakan logika fuzi dengan membandingkan unjuk kerjanya dengan tapis optimal FIR dan control PD dalam mereduksi derau. Hasilnya dengan menggunakan algoritma yang diusulkan kendali fuzi jauh lebih tinggi tingkat pelemahan derau dibandingkan dengan unjuk kerja kendali tapis optimal linear. Namun, teknik logika fuzi menemui kesulitan dalam menemukan banyaknya aturan fuzi yang digunakan yang disebabkan perubahan masukan. ANC pada umumnya di rancang berdasarkan model matematis yang bersifat *linear*. Model seperti ini sulit digunakan untuk mengontrol *noise* dalam lingkungan *plant* yang tak-*linear*. Hal inilah yang menyebabkan performansi peredam *noise* pada *broadband* frekuensi menjadi turun dibanding peredam *noise* pada *narrowband* frekuensi. Untuk mengatasi hal ini pada sistem ANC digunakan model matematis yang rumit untuk merancang bagian pengontrol. Pada sistem ANC dengan fuzi adaptif *filtered-x*, dapat dilakukan pemrosesan data numeris dan informasi linguistik untuk mengadaptasikan sistem ANC. Keuntungan dari sistem ini antara lain, dapat mengurangi kompleksitas sistem dan dapat mengkompensasi sifat ketak-linearan pada *plant*.

Kata kunci : Derau, *s-functions, Active Noise Control* (ANC), *filtered-x* LMS, fuzi adaptif.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Teknik pengolahan sinyal audio digital saat ini telah mengalami perkembangan yang sangat pesat. Pengolahan sinyal seperti transmisi, perekaman, penapisan, kompresi dan ekspansi dapat dikerjakan dengan menghasilkan distorsi yang kecil dan dengan SNR (*signal to noise ratio*) yang lebih baik daripada pemrosesan sinyal analog (Suwadi., et al, 2000).

Salah satu teknik tersebut adalah sistem ANC (*Active Noise Control*) yang menggunakan sinyal buatan (*artificial signal*) dengan besar yang sama serta fasanya berlawanan untuk menghapuskan derau atau bunyi yang tidak diinginkan (Chang YC., et al, 2000). Gelombang akustik (*acoustic wave*) atau suara bising yang tidak diinginkan memberikan pengaruh negatif pada manusia. Ia memberikan keadaan yang tidak nyaman pada seorang operator dalam menjalankan tugasnya, bahkan dapat mengalami kecelakaan kerja akibat pendengaran yang terganggu karena suara bising tersebut.

Metode konvensional yang biasa digunakan dalam mengendalikan atau melemahkan derau disebut kendali derau secara pasif (*Passive Noise Control, PNC*) memiliki kesanggupan untuk menekan derau akustik pada frekuensi tinggi, namun untuk derau akustik frekuensi rendah, PNC tidak dapat bekerja secara baik. Misalkan untuk frekuensi yang rendah dibawah 500 Hz, yaitu gelombang bunyi yang memiliki frekuensi 100 Hz akan memiliki panjang gelombang kira-kira 3,4 m pada kondisi normal (Elliott, S.J., et al, 1993). Hal ini menyebabkan material penyerap derau (*noise absorber*) ketebalannya harus ditambah untuk cukup mereduksi derau dengan baik, yang menjadikannya tidak praktis (Park, H.K., et al, 1999). Sementara itu derau (suara bising) yang ditimbulkan oleh mesin-mesin industri, mobil, kipas angin, pengkondisi udara dan pesawat terbang memiliki karakteristik-karakteristik frekuensi rendah yang memiliki panjang gelombang suara yang sangat panjang, sehingga dibutuhkan bahan yang besar dan tebal.

Berbeda dengan metode pasif, ANC tidak hanya dapat menghapuskan derau frekuensi rendah, tetapi juga mengurangi bobot, volume, dan biaya sistem kendali derau secara keseluruhan (Kuo S. M. dan Morgan D. R., et al, 1996). Dengan menempatkan sistem ANC pada suatu tempat, ia akan mengidentifikasi fungsi alih dari plant akustik, dan tranduser seperti *speaker* dan *mikrophone* serta *plant* saluran (*duct plant*) untuk membangkitkan isyarat anti derau yang tepat.

Banyak peneliti telah menerapkan algoritma *filtered-x* sebagai pengendali derau secara adaptif karena kesederhanaannya. Algoritma *filtered-x* adalah suatu tapis adaptif yang parameter-parameter bobotnya dapat secara otomatis diperbaharui menggunakan algoritma LMS (*least mean square*). Pendekatan ini cukup efektif untuk mereduksi derau frekuensi rendah, seperti gangguan dari fan atau mesin elektrik dalam sistem akustik. Namun kekurangan dari metode ini terdapat pada kecepatan konvergensinya yang lambat. Hal ini disebabkan algoritma *filtered-x* membutuhkan ukuran langkah (*step size*) yang kecil untuk memperbaharui parameter-parameter bobotnya agar dapat menjaga kestabilan unjuk kerja sistem. Kesulitan lain yang muncul dari lambatnya perubahan bobot ini adalah tidak dapat mengikuti perubahan sinyal gangguan pada waktu yang bersamaan (*in time*). Inilah yang menjadi penyebab utama kekurangan algoritma *filtered-x*. Oleh sebab itu kecepatan menjejak (*tracking*) sinyal gangguan menjadi sangat rendah sehingga tidak sempat dibangkitkan sinyal anti-*noise* yang akurat. Hal ini mengurangi performansi peredam *noise* yang bekerja pada daerah frekuensi yang lebar (*broad band*).

Beberapa sistem ANC menggunakan logika fuzi dengan membandingkan unjuk kerjanya dengan tapis optimal FIR dan kontrol PD dalam mereduksi derau. Hasilnya dengan menggunakan algoritma yang diusulkan kendali fuzi jauh lebih tinggi tingkat pelemahan derau dibandingkan dengan unjuk kerja kendali tapis optimal linear. Namun, teknik logika fuzi menemui kesulitan dalam menemukan banyaknya aturan fuzi yang digunakan yang disebabkan perubahan masukan (Vijila S.K.C.,et al,2003).

Salah satu cara untuk meningkatkan kinerja sistem ANC adalah menggunakan algoritma fuzi adaptif *filtered-x*. Selayaknya sistem fuzi, maka fuzi adaptif

filtered-x terdiri dari informasi bahasa yang mewakili kepakaran manusia dan hanya sedikit berisi informasi numeris. Metode ini dapat mengurangi sisa derau dari sistem ANC yang menggunakan algoritma *filtered-x* karena dapat mengatur parameter-parameter bobotnya secara otomatis. Oleh karena itu, permasalahan atas kelemahan pada kecepatan konvergensi dapat diatasi.

1.2 Rumusan Masalah

Permasalahan yang terdapat pada latar belakang pendahuluan disusun dalamrumusan masalah sebagai berikut :

1. Bagaimana mereduksi derau (*noise*) pada frekuensi rendah (<500Hz) dalam suatu saluran udara (*duct*) dengan ukuran 300 x 50 x60 cm?
2. Bagaimana cara menerapkan algoritma fuzi adaptif *filtered-x* pada sistem ANC dengan simulasi program MATLAB 6.5?

1.3 Ruang Lingkup

Berdasarkan permasalahan yang ada, maka bahasan yang akan dibuat dibatasi pada hal-hal sebagai berikut:

1. Pembahasan lebih dititikberatkan pada perancangan ANC pada *plant* saluran udara (*duct plant*) kanal tunggal menggunakan algoritma fuzi adaptif *filtered-x* secara *offline*.
2. Pengumpulan data derau (*noise*) dilakukan dengan membangkitkan suara derau dari program komputer.
3. Simulasi dilakukan dengan paket program MATLAB.

1.4 Tujuan

Tujuan skripsi ini adalah meneliti seberapa besar kemampuan dan kinerja sistem algoritma fuzi adaptif *filtered-x* dalam proses pengendalian derau secara aktif.

Skripsi ini juga bertujuan merancang dan membuat suatu perangkat lunak pengendali derau secara aktif menggunakan paket program Matlab.

1.5 Sistematika Penulisan

Laporan skripsi ini akan disusun dalam enam bab, yaitu :

BAB I Pendahuluan

Membahas latar belakang, rumusan masalah, ruang lingkup, tujuan, dan sistematika penulisan.

BAB II Dasar Teori

Membahas teori tentangkonsep dasar ANC, teori dasar algoritma fuzi adaptif *filtered-x* dan teori-teori lain yang mendukung dalam perencanaan *Active Noise Control* (ANC).

BAB III Metodologi Penulisan

Membahas mengenai langkah-langkah yang digunakan dalam menyelesaikan permasalahan dalam penyusunan skripsi.

BAB IV Pemodelan sistem ANC

Membahas perancangan dan realisasi sistem pengendalian derau secara aktif pada plant saluran udara (*duct plant*) menggunakan algoritma fuzi adaptif *filtered-x*.

BAB V Hasil Pengujian

Berisi hasil simulasi dari sistem ANC dengan berbagai parameter serta melakukan analisis sistem.

BAB VI Kesimpulan dan Saran

Membahas tentang kesimpulan hasil perancangan dan simulasi yang telah dilakukan dan saran untuk pengembangan aplikasi selanjutnya.

BAB II

DASAR TEORI

Masalah-masalah derau akustik semakin menjadi jelas dengan semakin bertambahnya pemakaian peralatan industri seperti mesin-mesin listrik, blower, kipasangin, *transformer*, kompresor dan lain-lain.Penggunaan tradisional kendali derau yang sering disebut kendali derau secara pasif menjadi tidak praktis berkaitan dengan besar volume, harga dan beratnya serta hanya mampu melemahkan derau frekuensi tinggi saja.

Kendali derau secara aktif (ANC, *active noise control*) meliputi sistem *electroacustic* atau *electomechanic* yang menghapus derau primer didasarkan pada prinsip superposisi,secara khusus suatu anti derau (*antinoise*) memiliki amplitude yang sama dan fasa yangberlawanan kemudian digabungkan dengan derau primer yang hasilnya adalahmenghapus kedua derau tersebut sehingga menciptakan daerah tenang (*silence zone*).

Dalam bab ini, secara singkat akan diuraikan konsep dasar ANC, algoritma *filtered-x,* logika fuzi dan algoritma fuzi adaptif yang akan menunjang dalam skripsi ini.

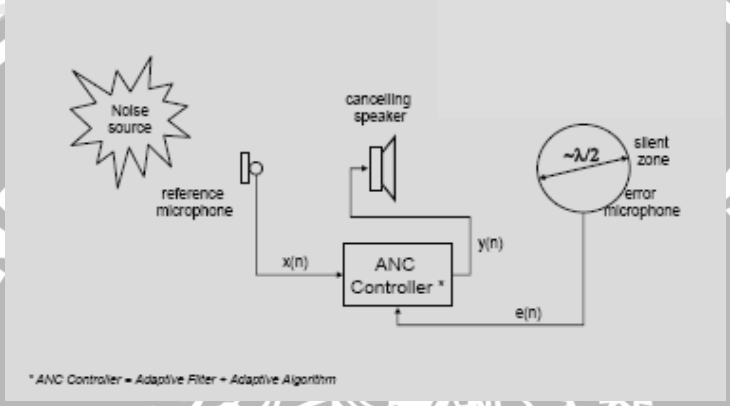
2.1 Konsep Dasar *Active Noise Control* (ANC)

Keberhasilan sistem ANC untuk meredam suara sangat bergantung pada akurasi dari amplitudo dan sudut fase dari gelombang anti-*noise* yang dihasilkan oleh sistem.Dalam penerapannya, jangkauan frekuensi *noise* yang dapat diredam tidaklah sebesar pada metode peredaman secara pasif.Di samping itu, sistem ANC dapat berfungsi secara optimal pada frekuensi rendah.*Active Noise Control* dalam penerapannya dibedakan menjadi dua jenis, yaitu *single-channel* dan *multi-channel* (Sen M.Kuo,1996).

Contoh dari penerapan dalam *single-channel* adalah peredaman suara pada saluran udara atau ventilasi, baik untuk pendingin maupun penghangat ruangan. Saluran ini dikenal dengan nama saluran HVAC (*Heating, Ventilating, and*

Conditioning) atau bisa juga disebut ruang akustik atau *duct.*Untuk penerapan pada *multiple-channel* contohnya adalah peredaman suara pada suatu ruangan terbuka. Dalam penelitian ini yang akan dicoba penerapannya ialah sistem *single-channel*.

Konfigurasi sistem ANC secara umum digambarkan dalam Gambar 2-1 di bawah ini.



Gambar 2-1. Sistem ANC secara umum

Sumber: Kuo, S.M. dan Morgan, D.R. 1999

Keterangan:

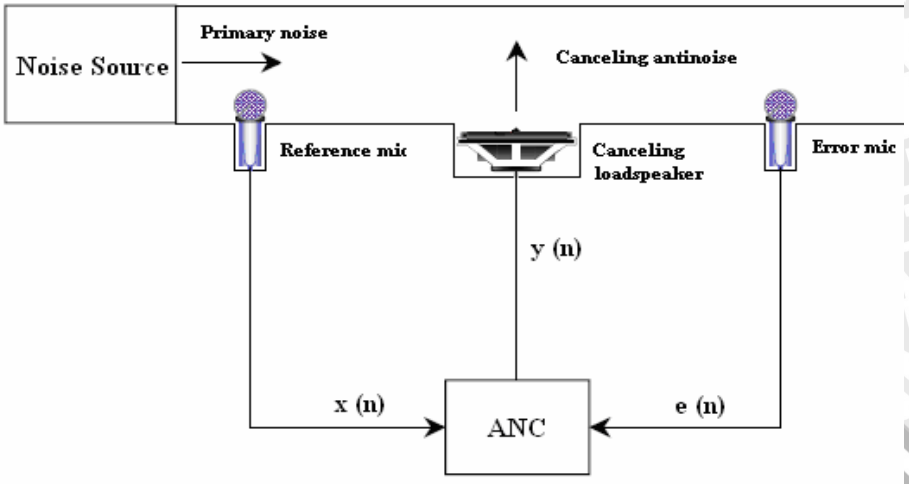
x(n) = sinyal referensi (*input*)

e(n) = sinyal *error*

y(n) = sinyal *output*

tanda * pada blok *ANC Controller* menunjukkan bahwa *ANC Controller* merupakan penjumlahan dari filter adaptif dan algoritma adaptif.

Peredaman derau secara aktif mengacu pada prinsip superposisi destruktif. Suatu sinyal anti derau (dengan amplitudo yang sama dan fasa yang berlawanan dengan sinyal derau primer) dibangkitkan dan dikombinasikan dengan sinyal derau primer, yang kemudian menghasilkan efek saling menghilangkan pada kedua sinyal tersebut. Tingkat penghilangan sinyal derau primer tergantung pada akurasi amplitudo dan fasa dari sinyal anti derau yang dibangkitkan.

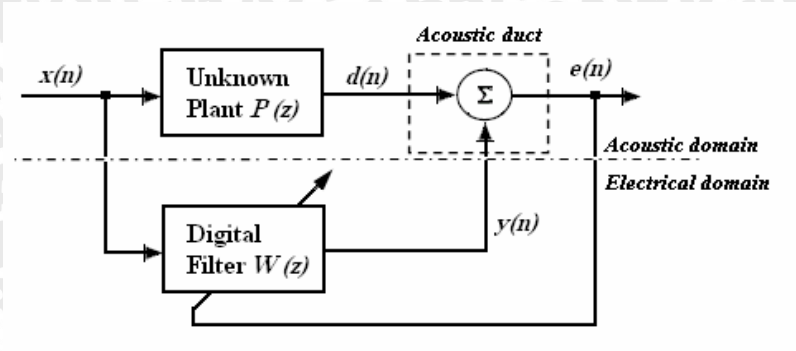


Gambar 2-2. Sistem ANC kanal tunggal dalam saluran udara (*duct*)

Sumber : Bharath. M Siravara, 2002:5

Sistem dasar ANC bidang lebar (*broadband*) ditunjukkan dalam Gambar 2-2. Dalam gambar tersebut terdapat noise primer, *microphone* refferensi untuk menangkap sinyal *noise* primer yang kemudian diubah menjadi sinyal refferensi *x(n)*. Terdapat juga *microphone* error untuk menangkap sinyal interferensi *e(n)* dan modul ANC yang berfungsi mengatur sinyal refferensi *x(n)* dan sinyal *error e(n)* sehingga menghasilkan sinyal anti-*noise y(n)*.

Kerangka sistem identifikasi adaptif ditunjukkan dalam Gambar 2-3.Dalam Gambar 2-3, memperlihatkan suatu filter adaptif *W* (*z*) yang digunakan untuk mengestimasi plant yang tidak diketahui P (z). Lintasan primer P (z) terdiri dari tanggapan akustik dari referensi sensor ke sensor *galat* dimana pelemahan derau dilakukan. Jika plant adalah dinamis, algoritma adaptif kemudian bertugas secara kontinyu untuk mengestimasi plant dinamis tersebut selama waktu berjalan.



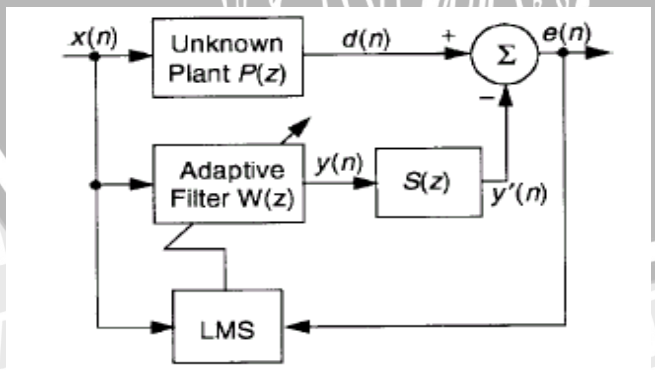
Gambar 2-3. Sistem identifikasi ANC

Sumber : Kuo, S.S., 1999:95

Tujuan tapis adaptif $W(z)$ adalah meminimalisasi sinyal error sisa $e(n)$. Dari Gambar 2-2, E (z) = 0 setelah tapis adaptif$W(z)$ konvergen. Kita kemudian mendapatkan W (z) = P (z) untuk X (z)≠ 0, yang memperlihatkan bahwa $y(n) = d(n)$. Sehingga ketika $d(n)$ dan $y(n)$ digabungkan secara akustik maka galat sisa $e(n)$ adalah $e(n) = d(n) - y(n) = 0$, yang merupakan hasil dari penghapusan sempurna kedua suara didasarkan prinsip superposisi.

2.2 Pengaruh Lintasan Sekunder

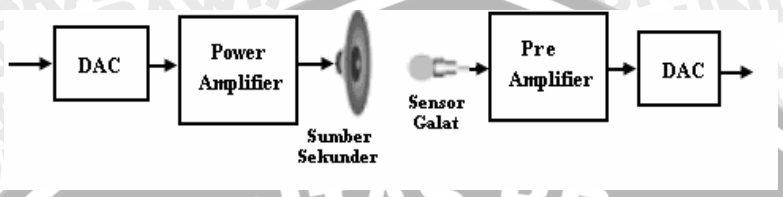
Penggunaan filter adaptif untuk penerapan ANC yang ditunjukkan dalam Gambar 2.3 adalah rumit, dengan suatu kenyataan bahwa simpul penjumlah (*summing junction*) dalam Gambar 2.4 mewakili superposisi akustik dalam suatu ruang dari *loudspeaker* penghapus ke *microphone error*, dimana derau primer digabungkan dengan keluaran filter adaptif.



Gambar 2-4. Penyederhanaan blok diagram ANC

Sumber : Kuo, S.M. dan Morgan, D.R. 1999

Sehingga diperlukan kompensasi untuk fungsi alih lintasan sekunder $S(z)$ dari $y(n)$ ke $e(n)$, yang mana biasanya berupa ADC, tapis rekonstruksi, penguat daya, *loudspeaker,microphone* dan lain-lain. Seperti yang ditunjukkan dalam Gambar 2-5.



Gambar 2-5. Jalur sekunder system ANC
Sumber : Kuo, S.M. dan Morgan, D.R. 1999

Untuk tujuan analisis, akan disajikan sistem aktual dalam Gambar 2-3 dengan blok diagram Gambar 2-4. Dari Gambar 2-4, transformasi z dari isyarat *error*adalah :

$$E(z) = [P(z) - S(z)W(z)]X(z) \qquad (2.1)$$

2.3 Sinyal-sinyal Noise

2.3.1 Derau Pita Lebar (*broadband noise*)

Derau pita lebar ditandai dengan tersebarnya energi akustik di banyak tempat dalam daerah frekuensi. Salah satu sebab derau jenis ini adalah gejala turbulensi yang bersifat acak, misalnya bunyi frekuensi rendah pesawat jet dan impuls ledakan. Untuk peredaman derau pita lebar, sinyal masukan referensi harus memberikan petunjuk pendahulu yang cukup tentang derau. Ini berarti sinyal acuan harus menyerupai sinyal derau utama. Salah satu caranya adalah dengan menempatkan *sensor* acuan sedekat mungkin dengan sumber derau. Dengan demikian sinyal acuan yang berkorelasi dengan derau utama dapat diperoleh, dan digunakan untuk membangkitkan sinyal anti derau yang dapat meredam derau di sekitar sumber sekunder.

2.3.2 Derau Pita Sempit (*narrowband noise*)

Derau pita sempit hanya memperlihatkan konsentrasi energi di tempat tertentu saja dalam daerah frekuensi.Derau jenis ini dapat berasal dari gejala mekanik yang berulang, seperti halnya putaran *as* dari mesin yang frekuensinya hampir periodik.Mesin *internal combustion*, turbin dan *compressor* adalah contoh sumber derau pita sempit.

2.4 Perambatan Gelombang Akustik

Perambatan gelombang suara pada suatu saluran (*path*) dipengaruhi oleh kondisi yang ada di sekitarnya. Jenis perambatan gelombang suara pada ruang tertutup, seperti pada saluran akustik / *duct*, dibagi menjadi dua macam, yaitu gelombang datar (*plane waves*) dan *cross mode wave* (*higher order acoustic mode propagation*) [Norton, M.P].

Pada gelombang datar, perambatan gelombang adalah pararel atau searah dengan saluran *(duct)*.Tekanan akustik yang terjadi pada penampang melintang dari saluran (*duct cross-section*) memiliki nilai yang konstan di sepanjang *duct*. Untuk *cross modewave*, arah perambatannya membentuk sudut terhadap dinding *duct*. Kemampuan gelombang akustik untuk merambat dalam *duct* dipengaruhi oleh kecepatan perambatan gelombang pada medium serta frekwensi gelombang akustik tersebut. Kecepatan perambatan gelombang suara pada medium ditentukan oleh [Conrad J., Hemond, Jr.]:

$$c = \sqrt{\frac{\gamma . G . T^{o}}{M}} \qquad (2.2)$$

c = kecepatan suara (m/s)

 = perbandingan dari suhu spesifik pada tekanan konstan dengan suhu spesifik pada volume konstan.

= 1,4 (untuk medium udara)

G = konstanta gas

= 8317 m^2 / s^2K

T° = suhu dalam Kelvin

= 27 °C + 273

= 300 K

M = berat molekul dari medium

= 29 (udara)

Jadi, dengan demikian kecepatan perambatan gelombang akustik pada udara

$$c = \sqrt{\frac{1{,}4.8317.300}{29}} = 347 m/s$$

2.5 Sistem ANC menggunakan algoritma filtered-x

Sistem dasar ANC yang menggunakan algoritma *filtered-x* ditunjukkan dalam Gambar 2-6.Sedangkan dalam kerangka *plant*, $H_p(z)$ merupakan speaker primer dan $P(z)$ merupakan silsilah matematis dari *transfer function* ruang akustik.Sinyal masukan x_k menyatakan jumlah contoh *noise* yang tidak diinginkan.Sedangkan sisa derau e_k dapat diukur dari *microphone* galat. Sinyal keluaran dari algoritma *filtered-x*, u_k, diarahkan menuju speaker anti-derau dengan *transfer function* $H_s(z)$ yang berfungsi untuk menghilangkan derau yang tidak diinginkan dalam sebuah ruang (*duct*). *Transfer function* dari lokasi speaker anti-derau menuju *microphone* galat $M_p(z)$ yang disebut sebagai alur galat dan disebut $H_e(z)$. Order (N+1) dari adaptif *active noise control* dengan pengubahan parameter bobot (W_k) diperbarui oleh algoritma *filtered-x* LMS. Oleh sebab itu, derau yang tidak diinginkan e_k dapat dicari dari persamaan (2.3)

$$e_k = x_k H_p P H_e M_p + W_k^t X_k H_s H_e M_p \qquad (2.3)$$

Urutan parameter bobot dan masukan derau ditunjukkan dalam persamaan (2.4)

$$W_k = \left[w_{ok}, w_{1k}, ..., w_{Nk}\right]^t \qquad (2.4)$$

$$X_k = \left[x_k, x_{k-1}, ..., x_{k-N}\right]^t \quad (2.5)$$

Estimasi gradien dari sistem adaptif dalam waktu ke-k ditunjukkan dalam persamaan (2.6) sebagai berikut :

$$\hat{\nabla}_k = \frac{\partial e_k^2}{\partial W_k} = 2e_k (X_k H_s H_e M_p) \quad (2.6)$$

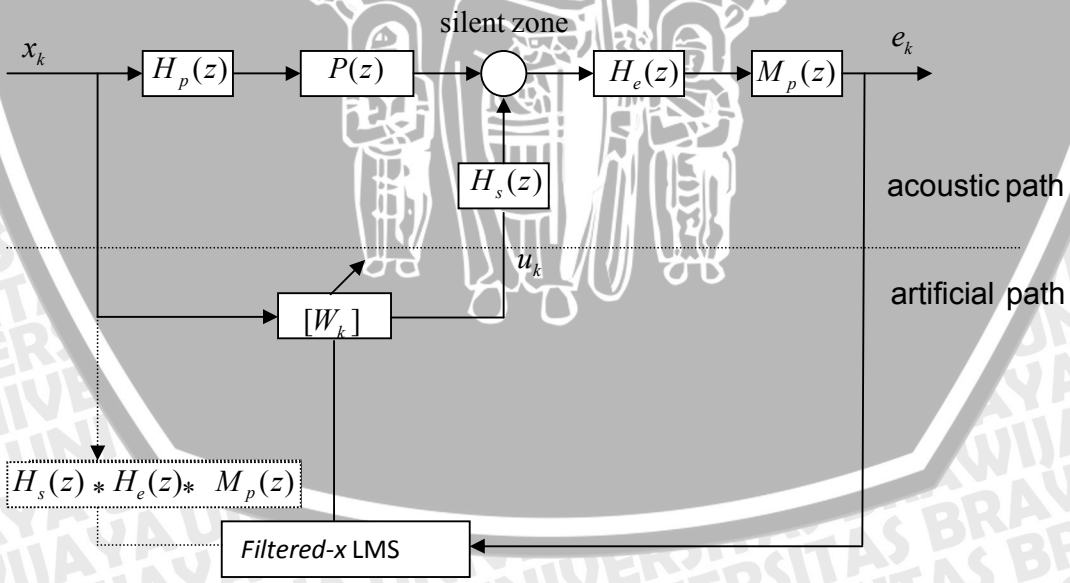
Sedangkan parameter bobot diperbarui oleh algoritma LMSditunjukkan dalam persamaan (2.7) sebagai berikut :

$$W_{k+1} = W_k - \mu\hat{\nabla}_k = W_k - 2\mu e_k (X_k H_s H_e M_p) \quad (2.7)$$

Persamaan tersebut berbeda dengan algoritma LMS yang standar yaitu :

$$W_{k+1} = W_k - 2\mu e_k X_k \quad (2.8)$$

Berdasarkan persamaan(2.7) dan (2.8), maka terlihat perbedaan pada bagian akhir. Pada bagian akhir (2.7) serupa dengan urutan keluaran derau Xk yang merupakan gabungan filter HsHeMp. Oleh karena itu (2.8) disebut sebagai algoritma *filtered-x* LMS.



Gambar 2-6. Blok diagram ANC berbasis *filtered-x* LMS

Sumber : perancangan

2.6 Himpunan Fuzi dan Inferensi Fuzi

Bagian ini memberikan suatu pengantar singkat dan ringkas dari konsep-konsep dasar himpunan fuzi, serta notasi-notasi himpunan fuzi yang akan digunakan dalam penelitian ini.

2.6.1 Himpunan-Himpunan Fuzi

Suatu himpunan klasik adalah himpunan dengan batas *crisp*. Untuk contoh, suatuhimpunan klasik dapat dinyatakan dengan

$$A = \{x | x > 6\},$$

adalah jelas dan batasnya tidak membingungkan yaitu 6, sedemikian hingga jika x lebihbesar dari bilangan ini, maka x adalah anggota himpunan A, jika sebaliknya maka $x$bukan anggota bilangan himpunan ini. Berbeda dengan himpunan klasik, suatuhimpunan fuzi, seperti namanya, adalah suatu himpunan yang tidak memiliki batas *crisp*.Yaitu, transisi dari "anggota dari suatu himpunan (*belonging to a set*)" ke "bukan anggotadari suatu himpunan (*not belonging to a set*)" sedikit demi sedikit, dan transisi yang halus initerkarakterisasi oleh fungsi keanggotaan yang memberikan fleksibelitas pada himpunanfuzi dalam pemodelan pernyataan-pernyataan bahasa yang biasa digunakan, seperti *"the water is hot"* atau *"the temperature is high"*. Sebagai catatan bahwa kefuzian (*fuzziness*) bukanlah datang dari keacakan (*randomness*) suatu bagian anggota dari suatu himpunan, tapi dari ketidakpastian (*uncertain*) dan ketidakjelasan (*imprecise*) alami konsep-konsep danpikiran-pikiran yang abstrak.

Definisi 1: Himpunan-himpunan fuzi dan fungsi-fungsi keanggotaan.

Jika X adalah kumpulan obyek yang dilambangkan secara umum oleh x, maka himpunan fuzi Adidalam X yang ditentukan sebagai himpunan urutan pasangan adalah:

$$A = \{(x,\mu(a)) | x \in X\} \qquad (2.9)$$

$\mu A(x)$ adalah fungsi keanggotaan (*membership function*, atau disingkat MF) x di A. MF memetakan setiap elemen X ke suatu derajad keanggotaan kontinyu (atau *membershipgrade*) antara 0 dan 1.

Biasanya X disebut sebagai semesta wacana (*universe of discourse*), yang dapat terdiridari obyek-obyek diskrit atau nilai-nilai kontinyu. Contohnya adalah sebagai berikut:

Contoh 1: Himpunan fuzi dengan nilai *X* diskrit.

Misalkan *X* = {1, 2, 3, 4, 5, 6, 7, 8} adalah himpunan jumlah mata kuliah yangmungkin diambil oleh seorang mahasiswa dalam suatu semester. Makahimpunan fuzi *A* = "Jumlah matakuliah yang diambil" dapat digambarkansebagai berikut:

A = {(1, 0.1), (2, 0.3), (3, 0.8), (4, 1), (5, 0.9), (6, 0.5), (7, 0.2), (8, 0.1)}.

Himpunan fuzi ini ditunjukkan dalam Gambar 2-7(a).

Contoh 2: Himpunan fuzi dengan X kontinyu.

Misal $X = R^+$ adalah himpunan umur manusia. Maka himpunan fuzi B ="umur kira-kira 50 tahun" dapat digambarkan sebagai berikut:

$$\mu_B(x) \quad \frac{1}{1+\left(\frac{x-50}{5}\right)^4}$$

Ilustrasi ini diperlihatkan dalam Gambar 2-7(b)

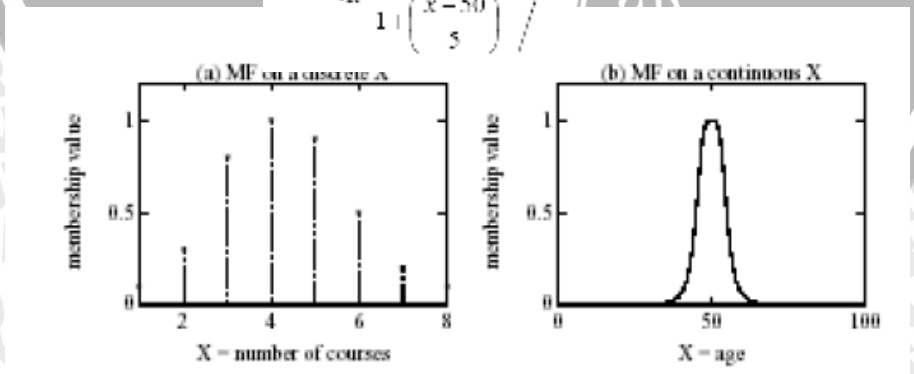
Cara lain untuk melambangkan himpunan fuzi adalah:

$$A = \begin{cases} \sum_{x_i \in X} \mu_A(x_i)/x_i, & \text{jika } X \text{ adalah diskrit} \\ \int_X \mu_A(x_i)/x_i, & \text{jika } X \text{ adalah kontinyu} \end{cases}$$

Tanda penjumlahan dan integrasi dalam himpunan fuzi menunjukkan pasangan gabungan(*union*) ((*x x*,μ*A*)), ia tidak menunjukkan penjumlahan dan integrasi. Secara sama, "/"hanyalah penanda dan bukan menunjukkan pembagian. Menggunakan notasi ini, dapatditulis ulang himpunan fuzi dalam contoh 1 dan 2 berurutan sebagai

A = 0.1/1 + 0.3/2 + 0.8/3 + 1.0/4 + 0.9/5 + 0.5/6 + 0.2/7 + 0.1/8, dan

$$B \quad \int_R \frac{1}{1+\left(\frac{x-50}{5}\right)^4} / x \,.$$



Gambar 2-7. (a) A = "jumlah mata kuliah yang diambil".

(b) B = "umur kira-kira 50 tahun".

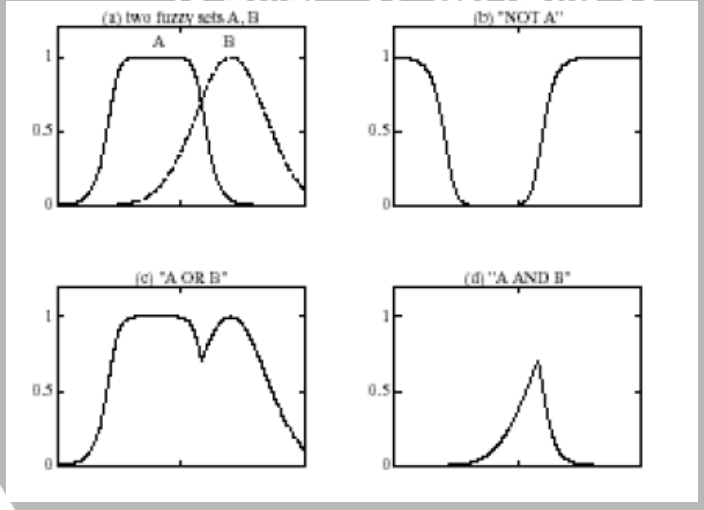
Sumber: Jang, JSR. , et al, 1995: 2.

Dari contoh 1 dan 2, dapat dilihat bahwa konstruksi suatu himpunan fuzitergantung pada dua hal yaitu identifikasi semesta wacana yang sesuai dan penentuanfungsi keanggotaan yang sepadan. Sebagai catatan, penentuan fungsi keanggotaan adalahsangat subyektif, dimana fungsi keanggotaan yang ditentukan untuk konsep yang sama(misal "*cold*") oleh orang yang berbeda hasilnya dapat tidak sama. Subyektifitas ini datangdari ketidaktentuan alami dari konsep-konsep abstrak dan tidak sama dengan konsepkeacakan. Subyektifitas dan ketidakacakan himpunan fuzi adalah perbedaan mendasardiantara studi himpunan fuzi dan teori probabilitas, yang mana berkaitan denganperlakuan obyektif dari fenomena acak.

Definisi 2: gabungan (*union or disjunction*).

Gabungan dua himpunan fuzi *A* dan *B* adalah himpunan fuzi *C*, yang ditulis sebagai$C = A \cup B$ atau $C = A$ *OR* B, dimana MF-nya dapat dinyatakan sebagai berikut

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \qquad (2.10)$$



Gambar 2-8. Operasi pada himpunan fuzi: (a) Dua himpunan fuzi A dan B, (b) A , (c) $A \cup B$, (d) $A \cap B$

Sumber: Jang, JSR. , et al, 1995: 3

Definisi 3: komplemen (*negation*).

Komplemen himpunan fuzi *A*, dilambangkan dengan $A(\neg A$, *NOT* $A)$, dan didefinisikan sebagai

$$\mu_{\neg A}(x) = 1 - \mu_A(x)$$

(2.11)

Gambar 2-8 mendemonstrasikan ketiga operasi ini: (a) mengilustrasikan dua himpunanfuzi *A* dan *B*, (b) komplemen *A*, (c) gabungan *A* dan *B*, (d) irisan *A* dan *B*. Operator-operatorAND dan OR dalam beberapa literatur disebut juga sebagai operator *T-norm*dan *T-conorm*.

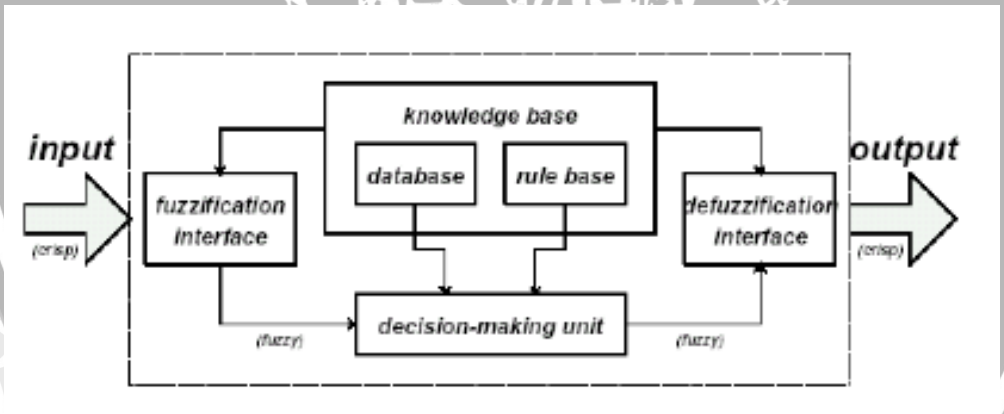
Definisi 4: Fungsi Keanggotaan (MF, *membership function*).

Ada beberapa MF yang sering kali digunakan dalam sistem fuzi misalnya segitiga, trapesium, sigmaoid, bell tergeneralisasi dan gaussian. Namun yang akan diuraikan disinihanya MF gaussian karena yang dipakai dalam penelitian ini.
MF gaussian dibentuk oleh dua parameter {σ, c} sebagai berikut:

$$gaussian(x;\sigma,c) = e^{-\left(\frac{x-c}{\sigma}\right)^2} \quad (2.12)$$

dimana c mewakili pusat MF dan σ menentukan lebar MF.

2.7 Sistem Inferensi Fuzi



Gambar 2-9. Sistem Inferensi Fuzi
Sumber: Jang, JSR, 1993: 2

Sistem inferensi fuzi yang juga dikenal sebagai sistem-sistem dasar aturan fuzi,model-model fuzi, memori asosiatif fuzi (*FAM*), atau pengendali fuzi ketika digunakansebagai pengendali. Pada dasarnya suatu sistem inferensi fuzi tersusun dari lima blok fungsi yang ditunjukkan dalam Gambar 2-9.

1.**basis aturan** (**rule base**) mengandung bilangan dari aturan-aturan fuzi *if-then*.

2.**basisdata**(**basisdata**) yang menentukan fungsi keanggotaan dari himpunan fuziyang digunakan dalam aturan fuzi.

3.**unit pembuatan-keputusan** (**decision-making**) yang membentuk operasi-operasiinferensi pada aturan-aturan.

4. **antarmuka fuzifikasi** (**fuzzification interface**) yang mengalihragam masukan*crisp* kedalam derajad kesesuaian (*degrees of match*) dengan nilai*linguistic*.

5. **antarmuka defuzifikasi (defuzzification interface)** yang mengalih-ragamkanhasil-hasil inferensi fuzi kedalam keluaran *crisp*.

Biasanya, basis aturan dan basisdata digabungkan sebagai basis pengetahuan (*knowledgebase*).

Tahap-tahap penalaran fuzi (*fuzzi reasoning*) yaitu operasi-operasi inferensi berdasaraturan-aturan fuzi adalah sebagai berikut:

1. Bandingkan variabel masukan dengan fungsi keanggotaan pada bagian premisuntuk mendapatkan nilai keanggotaan (atau ukuran-ukuran kesesuaian) darisetiap label *linguistic*. (Tahap ini sering disebut fuzifikasi)

2. Gabungkan (melalui suatu operator khusus *T-norm*, biasanya perkalian atau *min*)nilai-nilai keanggotaan pada bagian premis untuk mendapatkan kuat penyulutan(*firing strength*) dari setiap aturan.

3. Bangkitkan konsekuen yang memenuhi syarat (termasuk fuzi atau *crisp*) dari setiapaturan bergantung pada kuat penyulutan.

4. Jumlahkan konsekuen yang memenuhi syarat untuk menghasilkan keluaran *crisp*.(Tahap ini disebut defuzifikasi)

Beberapa tipe inferensi sistem fuzi telah digunakan secara luas dalam berbagaimacam penerapan.Perbedaan diantara ketiga tipe ini adalah berdasarkan konsekuen-konsekuenaturan fuzinya, agregatnya serta prosedur defuzifikasinya.

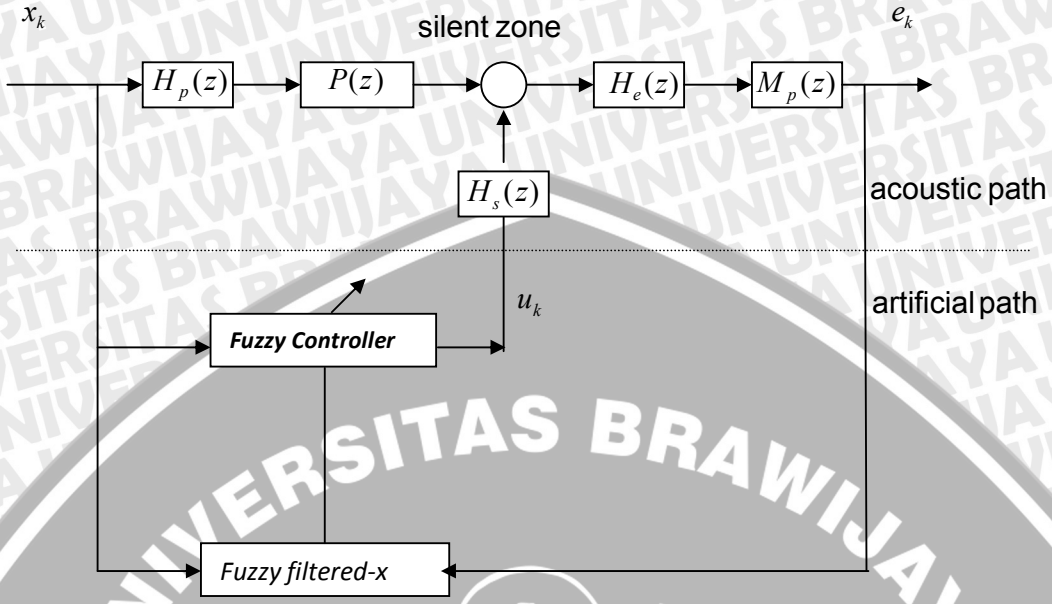
2.8Fuzi adaptif filtered-x dalam sistem ANC

Pada tahun 1965, Zadeh telah mempublikasikan teori pertama tentang himpunan fuzi. Sejak saat itu himpunan fuzi dan sistem fuzi telah dikembangkan dan diterapkan kedalam berbagai macam bidang yaitu kendali otomatis,

pengolahan isyarat, prediksi runtun waktu, identifikasi sistem non-linear, klasifikasi data, komputer *vision*, manajemen basis data, pengambilan keputusan dan lain-lain (Jang, JSR., et al, 1995).

Fungsi keanggotaan fuzi yang awalnya harus ditala secara manual oleh seorang operator, oleh Wang dikembangkan menjadi bentuk adaptif dengan menggunakan algoritma belajar tertentu, yang terkenal disebut fuzi adaptif (Wang, LX., et al, 1992).Fuzi adaptif juga telah diterapkan pada berbagai bidang misalnya identifikasi, kendali *ball and beem*, prediksi runtun waktu dan tapis-tapis adaptif (Wang, LX., et al, 1994). Chang dan Kuo (Chang, YC., et al, 2000 dan 2002) menggunakan fuzi adaptif ini sebagai pengendali derau secara aktif (ANC, *active noise control*) untuk mehapuskan isyarat-isyarat derau yang tidak dikehendaki. Dalam teori tersebut algoritma belajar yang digunakan untuk mentala fungsi keanggotaannya adalah algoritma belajar LMS dan RLS.

ANC pada umumnya di rancang berdasarkan model matematis yang bersifat *linear*.Model seperti ini sulit digunakan untuk mengontrol *noise* dalam lingkungan *plant* yang tak-*linear*.Hal inilah yang menyebabkan performansi peredam *noise* pada *broadband* frekuensi menjadi turun dibanding peredam *noise* pada *narrowband* frekuensi.Untuk mengatasi hal ini pada sistem ANC digunakan model matematis yang rumit untuk merancang bagian pengontrol.Pada sistem ANC dengan fuzi adaptif *filtered-x*, dapat dilakukan pemrosesan data numeris dan informasi linguistik untuk mengadaptasikan sistem ANC, seperti ditunjukkan pada Gambar 2-10. Keuntungan dari sistem ini antara lain, dapat mengurangi kompleksitas sistem dan dapat mengkompensasi sifat ketak-linearan pada *plant*.



Gambar 2-10. Blok diagram ANC berbasis fuzi adaptif

Sumber : perancangan

Berikut ini dijelaskan langkah-langkah untuk merancang system ANC dengan fuzi adaptif *filtered-x* :

Langkah 1 : Keluaran tapis FIR *orde* N+1 diperoleh dari rerata N+1 cuplikan masukan sinyal *noise,* dari sini dapat digunakan masukan sebanyak N+1 cuplikan yang telah lalu untuk merancang tapis fuzi. Tapis fuzi orde N+1 dalam sistem ANC dapat didefinisikan dengan fuzi setFi sebanyak M untuk setiap cuplikan data masukan *noise* x_{k-1} dengan fungsi keanggotaan tipe *Gaussian*:

$$\mu_{F_i^l}(x_{k-1}) = \exp\left[-\frac{1}{2}\left(\frac{x_{k-1}-\bar{x}_{i,k}^1}{\sigma_{x_i}^1}\right)^2\right] \qquad (2.12)$$

dengan l =1,2,…,M dan i=0,1,2,…,N, sedangkan k adalah nomor cuplikan. Pada persamaan, $\bar{x}_i^1$ adalah pusat *(center)* dan $\sigma_{x_i}^1$ adalah standar deviasi dari fungsi *Gaussian*. Pada skripsi ini menggunakan lima *membership functions* sehingga memiliki lima label linguistik yang terdistribusi pada jangkauan masukan antara [–1 1]; NB *(negative big), NS(negative small), AZ (allmost zero),* PS *(positive*

small), dan PB *(positive big).*

Langkah 2 :Mendefinisikan M aturan fuzi *IF-THEN* dengan tipe seperti berikut:

$$Rule^{l}: IF\ x_k\ is\ F_N^1\ and\ ....and$$

$$x_{k-N}\ is\ F_N^1\ THAN\ y_k\ is\ U^l \tag{2.13}$$

l =1,2,…,M. Pada persamaan, F_i^lseperti yang didefinisikan sebelumnya dan set U^l berupa fuzi *singleton* pada θ_k^l. Untuk nilai *crisp* pertama nilai θ_k^l dapat dipilih sembarang antara –1 sampai dengan 1, sedangkan aturan fuzi diberikan pada tahap berikutnya.Tujuh aturan fuzi membentuk tapis fuzi FIR, yang berlaku sebagai sebuah tapis anti-*noise* untuk meredam gangguan yang tidak diinginkan.Basis aturan dari sistem ini ditunjukkan pada Tabel 1.

Tabel 1. Aturan fuzi pada sistem ANC

	Antecedence				*Consequents*
	x_k	x_{k-1}	...	x_{k-N}	y_k
Rule 1	NB	NB	NB	NB	NB
Rule 2	NS	NS	NS	NS	NS
Rule 3	AZ	AZ	AZ	AZ	AZ
Rule 4	PS	PS	PS	PS	PS
Rule 5	PB	PB	PB	PB	PB

Langkah 3 :pada tahap ini diberikan ilustrasi mengenai proses pengambilan keputusan *(fuzzy inference)* dan proses *defuzzifikasi.* Pada *fuzzy inference,* digunakan prinsip logika fuzzy yang dikombinasikan dengan aturan fuzi yang memetakan masukan fuzi set ke keluaran fuzi set. Misal, pada tapis fuzi saat cuplikan ke k diberi masukan $X_k^* = \left[x_k^*, x_{k-1}^*, ..., x_{k-N}\right]$. Tapis fuzi ini

menggunakan mesin *inference* yang minimal dan *defuzzifikasi* tipe *centroid,* yang diperoleh dengan mengkombinasikan aturan sebanyak *M* yang didefinisikan pada tahap sebelumnya. Tapis ini di beri nama $f\left(X_k^*\right)$, dalam hal ini digunakan $u_k$untuk mengontrol sinyal yang tidak diinginkan, seperti ditunjukkan dalam Gambar 2-10.

*Langkah 4 :*Untuk meminimisasi daya sinyal gangguan, algoritma fuzi *filtered-x* LMS menggunakan formula sebagai berikut, untuk memperbarui bobot parameter θ_k^1, $\bar{x}_{i,k}^1$, dan $\sigma_{xi,k}^1$ pada setiap waktu cuplikan ke-*k*:

$$\theta_{k+1}^l = \theta_k^l + \lambda.e_k.\frac{p_k^l}{q_k} \tag{2-14}$$

$$\bar{x}_{i,k+1}^l = \bar{x}_{i,k}^l + \lambda.e_k.\gamma_k^l \frac{\left(x_{k-1} - \bar{x}_{i,k}^l\right)}{\left(\sigma_{xi,k}^i\right)^2} \tag{2.15}$$

$$\sigma_{i,k+1}^l = \sigma_{i,k}^l + \lambda.e_k.\gamma_k^l \frac{\left(x_{k-1} - \bar{x}_{i,k}^l\right)^2}{\left(\sigma_{xi,k}^i\right)^3} \tag{2.16}$$

dengan

$$p_k^l = \min\left(\exp\left[-\frac{1}{2}\left(\frac{x_{k-1} - \bar{x}_{i,k}^l}{\sigma_{xi,k}^l}\right)^2\right]\right) \tag{2.17}$$

$$q_k = \sum_{l=1}^{7} p_k^l$$

$$\gamma_k^l = \frac{\theta_k^l + u_k}{q_k}.p_k^l \tag{2.18}$$

dan λ, adalah sebuah konstanta bilangan positif kecil. Bentuk terakhir pada persamaan diatas serupa dengan masukan gangguan $x_k$yang dikirimkan pada tapis fuzi.Jadi persamaan tersebut disebut sebagai algoritma LMS fuzi *filtered-x*, yang berbeda dengan algoritma LMS *filtered-x* biasa pada bagian faktor koreksi dan

algoritma fuzi. Pada pengkajian ini digunakan nilai awal parameter sebagai berikut :

$$\bar{x}_{i,0}^{l} = x_{k-i} \; dan \; \bar{x}_{i,0}^{l} \neq x_{k-i} \; untuk \; l \neq 1 \tag{2.19}$$

dan dipilih

$$\theta_0^1 = -x_k . \frac{H_p P}{H_s} \tag{2.20}$$

selanjutnya $\theta_0^1 (l \neq 1)$ nol *(zero).* Jadi dengan persamaan (2.19) dan (2.20),*anti-noise* u_k akan sama dengan θ_0^1. Setelah speaker kedua *Hs* aktif, *noise* buatan menjadi - $x_k H_p P$, yang akan mengkompensasi *noise* yang tidak diinginkan pada daerah tenang *(silent zone).*

Selanjutnya pada persamaan terakhir dapat diterapkan fungsi alih $H_eM_p$pada bagian belakang sehingga diperoleh :

$$\theta_0^1 = -x_k . \frac{H_p P H_e M_p}{H_s H_e M_p} \tag{2.21}$$

sebagai*numerator* adalah $H_pPH_eM_p$dan *denumeratornya* adalah $H_pPH_eM_p$. Keduanya merupakan fungsi alih antara sinyal elektris masukan dengan sinyal elektris keluaran.Dalam hal ini dapat digunakan *white-noise* sabagai sinyal masukan pada *speaker* utama $H_p$dan kemudian mengumpankan pada *microphone* $M_p$pada bagian akhir sistem untuk mengukur *numerator*, melalui lintasan udara.Tanggapan keluaran terhadap masukan adalah numerator pada persamaan terakhir. Sehingga dapat digunakan masukan sinyal *white-noise* sebagai masukan bagi $H_s$dan mengukur sisa *noise* dengan *microphone* M_p. Bagian denumerator pada persamaan tersebut juga dapat ditentukan. Sehingga dapat diperoleh persamaan sebelumnya dan *plant* tidak lagi diperlukan.

2.9 Bahasa Komputasi Teknik MATLAB

MATLAB adalah bahasa pemograman dengan kinerja tinngi untuk komputasi teknikal.Didalamnya terintegrasi kemampuan komputasi, visualisasi, dan pemograman dalam suatu lingkungan yang mudah untuk digunakan karena permasalahan dan pemecahannya dinyatakan dalam notasi matematika biasa. Kegunaan MATLAB secara umum adalah untuk :

- Matematika dan komputasi
- Pengenbangan algoritma
- Pemodelan, simulasi, dan pembuatan *prototype*
- Analisis data, eksplorasi, dan visualisasi
- Pembuatan aplikasi termasuk pembuatan antarmuka grafis

Nama MATLAB singkatan dari *Matrix Laboratory*.Pada awalnya MATLAB ditulis untuk menyediakan kemudahan mengakses matriks.*Software* ini dikembangkan oleh proyek *Linpack* dan *Eispack,* yang bersama-sama menyediakan *software* untuk komputasi matriks.Dalam perkembangannya MATLAB tidak sekedar untuk mengakses matriks dengan banyaknya *toolbox* (perangkat) yang disediakan.

Toolbox MATLAB adalah perpustakaan fungsi MATLAB yang membantu MATLAB untuk memecahkan suatu kelompok permasalahan. *Toolbox* ditulis dalam bahasa MATLAB sehingga dapat dilihat algoritma sumber dan menambah yang baru bila diperlukan. Bidang yang dimungkinkan untuk diolah menggunakan MATLAB meliputi *signal processing, control system, neural networks, fuzzy logic, wavelets, simulations,* dan bidang-bidang lainnya.

2.10 *Sound Card*

Sound card merupakan peralatan tambahan pada PC yang dapat digunakan untuk berinteraksi dengan dunia luar, misalnya untuk menerima dan mengirimkan sinyal suara atau sinyal analog yang lain yang memiliki jangkauan frekuensi hingga 20 kHz. Sinyal analog DC tidak dapat masuk ke bagian *sound card* karena

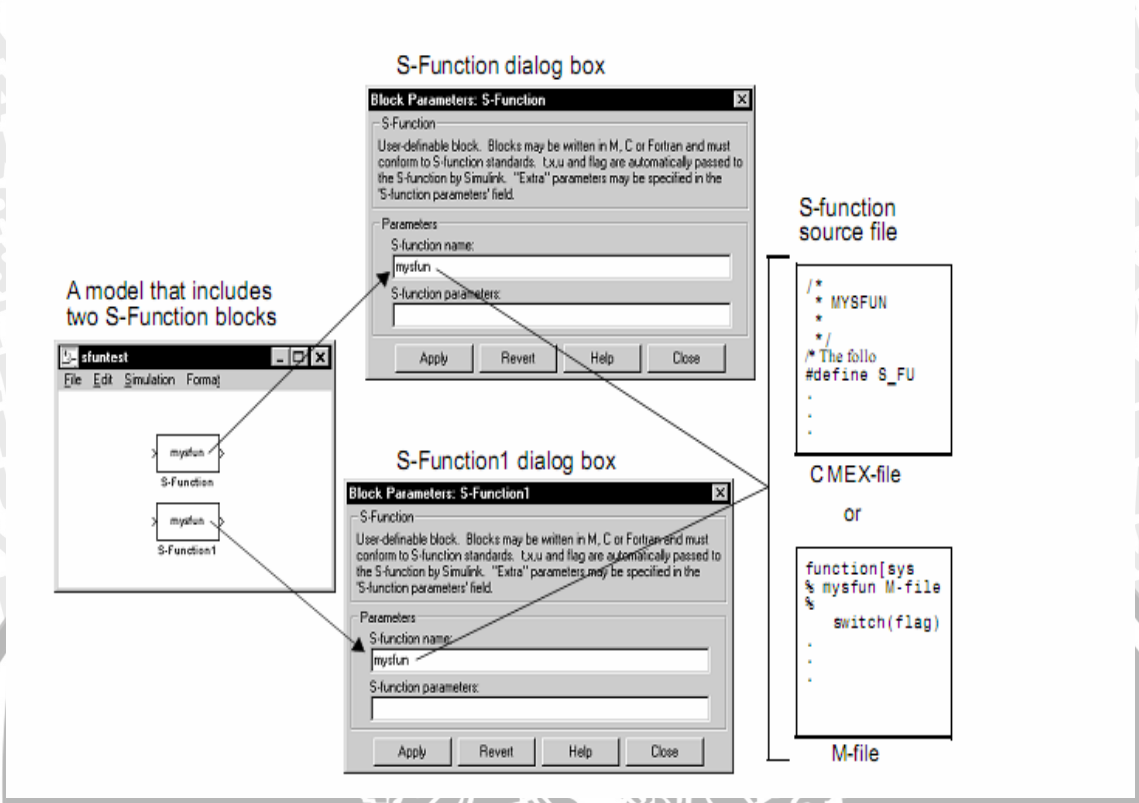
pada bagian masukan dari *sound card* terdapat sebuah kapasitor untuk meredam sinyal DC (*DC blocking / AC Coupling*).Di dalam *sound card* terdapat sebuah komponen ADC (*Analog to Digital Converter*) utnuk merubah sinyal analog menjadi sinyal digital.Juga terdapat komponen DAC (*Digitalto Analog Converter*) untuk merubah sinyal digital menjadi sinyal analog.Frekuensi cuplik dalam *sound card* yang standard dapat diatur antara 8 kHz hingga 44 kHz (Reich Heimlich dkk, 1993).

2.11 Metode *s-function*

S-funcions adalah bahasa pemograman komputer yang memiliki deskripsi sistem yang dinamis.*S-function* dapat ditulis menggunakan Matlab atau bahasa C. *S-functions* menyediakan mekanisme untuk meningkatkan kemampuan Simulink karena *S-functions* menggunakan sintaks khusus yang memungkinkan kita untuk berinteraksi dengan Simulink's equation solvers. Interaksi ini sangat mirip dengan interaksi yang berlangsung antara solvers dan built-in Simulink blok.

Bentuk s-*functions* yang sangat umum dan dapat menampung kontinyu, ciri-ciri, dan sistem campuran. Akibatnya, hampir semua model Simulink dapat digambarkan sebagai *s-functions*. *S-functions* yang dimasukkan ke dalam model Simulink dengan menggunakan *s-function*blok di *Nonlinear Blokir sublibrary*.Menggunakan s-function blok dari kotak dialog untuk menentukan nama yang *s-function*,seperti yang ditunjukkan dalam Gambar 2-11.:

Berdasarkan Gambar 2-11, terdiri dari dua model contoh sebuah blok *s-function.* Keduablok memiliki referensi sumber file yang sama (mysfun, yang dapat berupa sebuah file-CMEX atau M-file). Jika diantara CMEX-file dan M-file yang ada dengan nama yang sama, maka CMEX-file memerlukan waktu dan diutamakan adalah file yang menggunakan *s-function*. Menggunakan fasilitas Simulink untuk membuat kotak dialog dan ikon untuk blok*s-function*. Kotak dialog dapat memudahkan menetapkan parameter tambahan untuk s-*functions*.



Gambar 2-.11 Hubunganblok s-functions, Dialog Box, dan source file.

Sumber : Perancangan

Secara umum penggunaan *s-function* adalah untuk menciptakan Simulink blok.Sehinggas*-function* dapat digunakan untuk berbagai aplikasi, yaitu:

•Menambahkan tujuan umum baru ke blok Simulink

•Bekerjasama C kode yang ada ke dalam simulasi

•Menjelaskan sistem seperti kumpulan rumus matematika

•Menggunakan grafis animasi (lihat demo bandul terbalik, **penddemo**)

Sebuah keuntungan dari menggunakan *s-functions* adalah Anda dapat membangun bloktujuan umum yang dapat digunakan beberapa kali dalam sebuah model, masing-masing dengan berbagai parameter contoh dari blok.

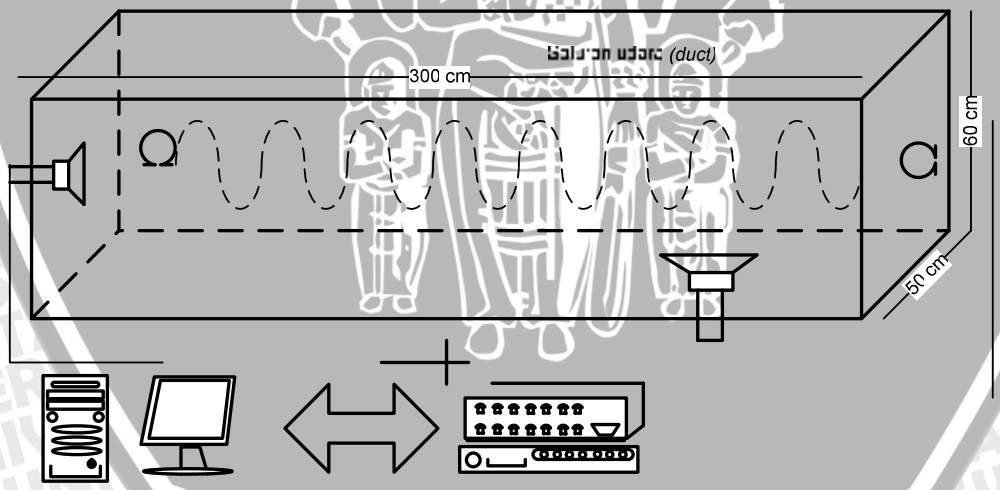
BAB III

METODE PENELITIAN

Penelitian yang dilakukan dalam skripsi ini menghasilkan sistem *Active Noise Control* yang mengendalikan derau secara aktif menggunakan algoritma fuzi *filtered-x*. Sistem ini bekerja pada frekuensi rendah. Adapun langkah-langkah yang dilakukan dalam penelitian adalah sebagai berikut :

3.1 Pengambilan Data *Sample Noise*

Pengambilan data yang dilakukan dengan cara membangkitkan *noise* berupa sinyal sinusoida 250Hz (*narrowband*), sinyal acak (*broadband*), dan gabungan dari sinus 250Hz dan sinyal acak (komposit). Software yang digunakan untuk membangkitkan noise adalah *Cool Edit Pro 2.0*. Gambar proses pengambilan data dilakukan pada perangkat keras yang ditunjukkan dalam Gambar 3-1sebagai berikut :



Gambar 3-1 Desain alat pengujian

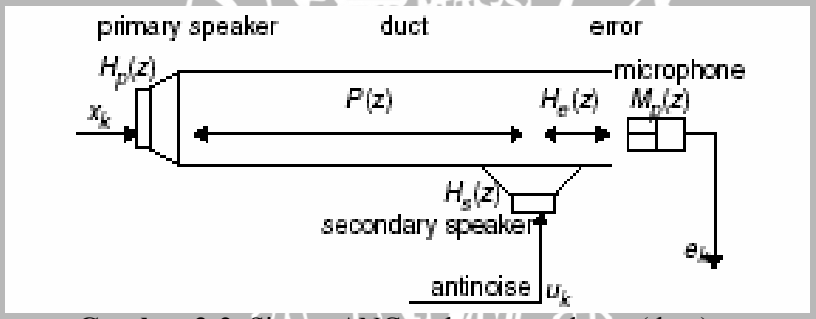
Sumber : perancangan

Berdasarkan Gambar 3-1, jika sistem diaplikasikan secara *realtime* maka diperlukan sedikitnya 2 buah PC. Pertama untuk menghasilkan sumber *noise* pada

speaker primer dan kedua untuk menghasilkan sinyal anti*noise* pada speaker sekunder. Pada saat mengaktifkan kedua speaker harus dilakukan secara bersamaan. Cara tersebut cukup rumit dengan kenyataan bahwa PC ataupun pada *software* pembangkit *noise* terdapat *delay*. Karena hal itu tidak memungkinkan, maka untuk tujuan analisis pada penelitian ini dilakukan pemodelan speaker, saluran *duct*, dan *microphone*-nya. Sedangkan untuk proses pengambilan data dilakukan secara *offline*.

3.2 Pemodelan Matematik Sistem

Berdasarkan Gambar 3-1 maka dapat dirumuskan pemodelan matematis sistem yang mengacu pada model saluran *duct* yang terdapat dalam literatur (Chang, YC.,et al, 2003) dan untuk analisis disusun diagram blok dari sistem ANC pada suatu saluran (*duct*) seperti yang ditunjukkan dalam Gambar 3-2.



Gambar 3-2. Sistem ANC pada suatu saluran (duct)

Sumber : Chang, YC., et al, 2003:417

Keterangan :

$H_p(z)$ = Speaker Primer

$H_s(z)$ = Speaker Sekunder (*antinoise*)

$P_z(z)$ = Fungsi alih saluran akustik (*acoustic duct*)

$x(k)$ = Sampel *noise* yang tidak diinginkan

e(k) = *error* sisa (*residual error*)

M_p (z)= *Microphone error*

H_e (z) = Fungsi alih lintasan *error* (jarak antara speaker sekunder dengan *microphone error*)

3.3 Pengujian Kinerja Sistem

Kinerja sistem dinyatakan dengan menghitung SNR (*signal to noise ratio*) awal dan akhirnya. SNR awal yaitu SNR sebelum dilakukan penghapusan derau dan SNR akhir yaitu SNR setelah penghapusan derau. Rumus yang digunakan untuk menghitung SNR ini yaitu:

$$SNR_{dB} = 20 * \log_{10} \frac{\text{varian(isyarat derau awal)}}{\text{varian(isyarat derau setelah reduksi)}} \quad (3.1)$$

Sedangkan untuk daya derau reratanya (*average noise power*) dapat dihitung dengan rumus berikut ini (Park H-K., dan Kong S-G., 1999) :

$$P_e = \frac{1}{m} \sum_{k=1}^{m} e^2(k) \quad (3.2)$$

dimana *m* adalah banyak data dan *e* adalah galat atau derau sisa (*residual noise*) yang terdapat pada daerah tenang (*silent zone*).

3.4 Pengambilan Keputusan

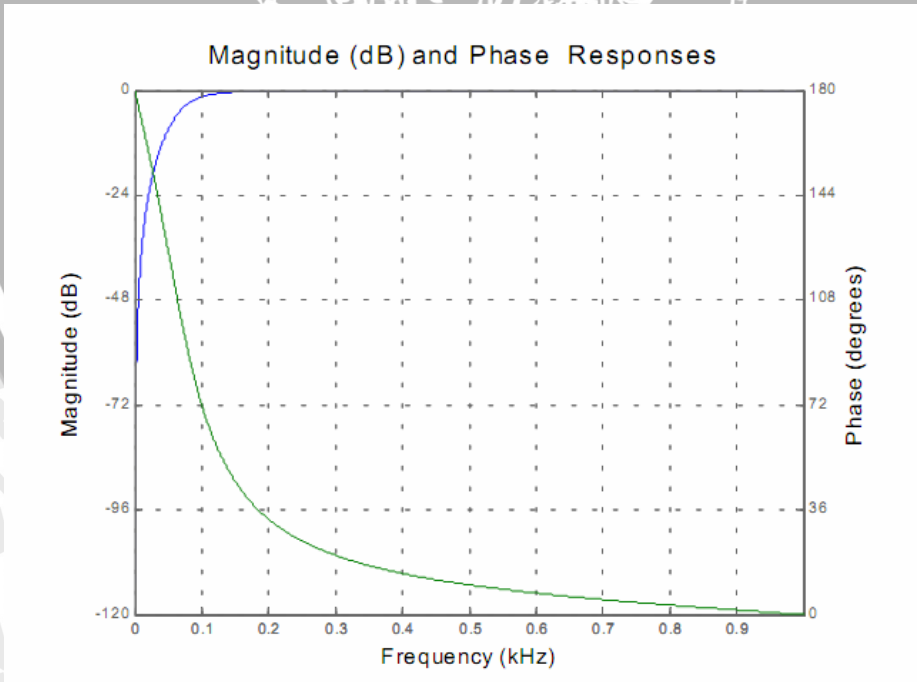
Pengambilan kesimpulan dilakukan setelah melakukan serangkaian pengujian. Jika hasil yang diperoleh sesuai dengan spesifikasi yang diinginkan maka alat tersebut telah memenuhi harapan.

BAB IV

PEMODELAN SISTEM ANC

4.1 Penentuan Model Sistem ANC

Berdasarkan Chang, YC., model saluran (*duct*) ditentukan sebagai berikut. Plant saluran akustik *P(z)* dimodelkan oleh waktu tunda murni (*pure time delay*) 25 sampel hal ini menunjukkan besarnya jarak terdekat dari speaker primer ke speaker sekunder serta isyarat yang melintasinya adalah isyarat yang terbesar artinya isyarat tersebut tidak mengalami pelemahan karena dispersi dinding saluran maupun karena jaraknya yang relatif dekat. Sedangkan lintasan galat $H_e(z)$ dimodelkan oleh waktu tunda 5 sampel. Berikut ini adalah gambar yang menunjukkan tanggapan *magnitude* dan fasa dari pemodelan speaker.



Gambar 4-1. Tanggapan Magnitude dan fasa dari model speaker

Sumber: perancangan.

Sedangkan untuk *speaker* primer, *speaker* sekunder, dan *microphone* galat dimodelkan oleh tapis lolos atas *Butterworth* orde dua dengan frekuensi *cut-off* 80 Hz, frekuensi *sampling*-nya sebesar 2000Hz sedangkan sistem menggunakan tapis FIR adaptif orde 21 dengan parameter bobot (*taps filter*) awal nol.

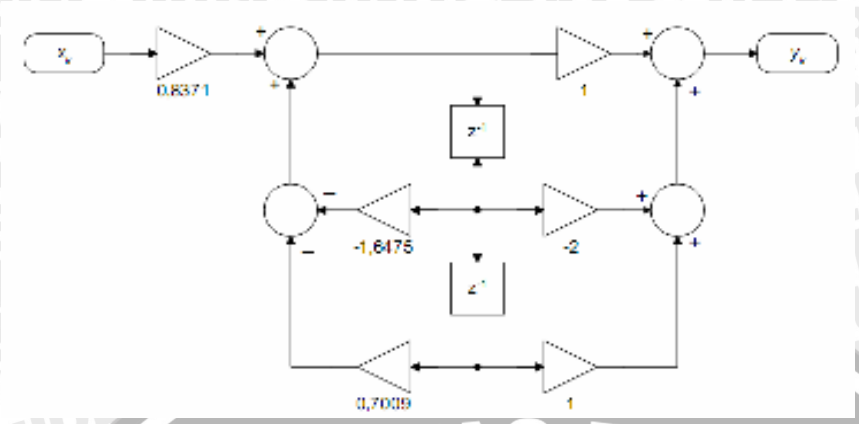
Berdasarkan informasi dari Gambar 4-2, maka dapat dibuat model tapis *Butterworth* menggunakan FDATool yang merupakan kelengkapan alat yang dimiliki MATLAB untuk mendisain dan menganalisis tapis digital. Dengan memasukkan nilai-nilai spesifikasi tapis yang telah ditentukan sebelumnya, maka dapat diperoleh tanggapan magnitude dan tanggapan fasanya seperti yang diperlihatkan dalam Gambar 4-1. Sedangkan dengan mengekspor hasil disain dari FDATool ke workspace MATLAB, maka akan diperoleh fungsi alih tapis sebagai berikut:

$$H(z) = 0{,}8371\,\frac{1 - 2z^{-1} + z^{-2}}{1 - 1{,}6475z^{-1} + 0{,}7009z^{-2}} \qquad (4.1)$$

dan dari fungsi alih tersebut dapat diperoleh ekuivalen persamaan bedanya yaitu :

$$y_k - y_{k-1} + 0{,}7009\,y_{k-2} = 0{,}8371\,(x_k - 2x_{k-1} + x_{k-2}). \qquad (4.2)$$

Dari FDATool juga dapat direalisasikan model tapis digital direct form II-nya, seperti yang terlihat dalam Gambar 4-1. Model ini akan disimulasikan menggunakan SIMULINK yang juga merupakan kelengkapan alat untuk simulasi yang disediakan oleh MATLAB 6.5.

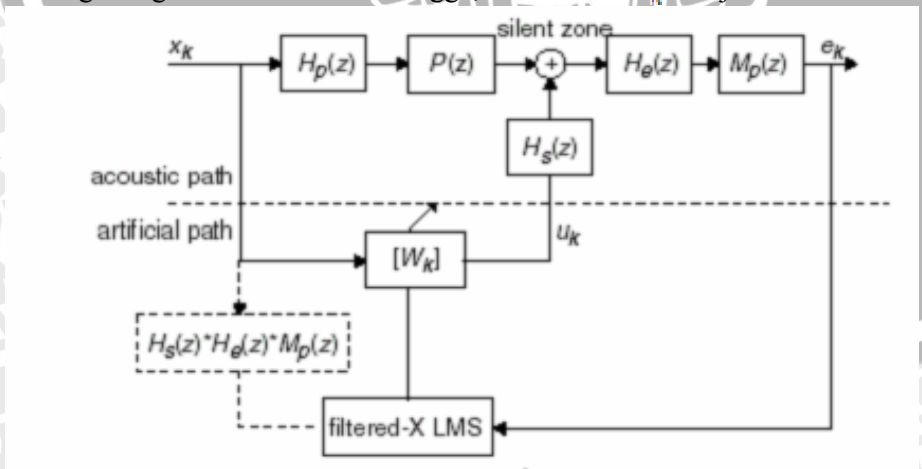


Gambar 4-2. Realisasi model tapis digital orde 2 dengan struktur direct form II.

Sumber: perancangan.

4.2 Algoritma FxLMS untuk Sistem ANC

Untuk tujuan analisis, sistem ANC kanal tunggal dapat diturunkan dalam sebuah diagram blok algoritma FxLMS-nya seperti dalam Gambar 4-3. Orde ke – (N+1) ANC adaptif dengan parameter-parameter bobot W_k akan diperbaharui dengan algoritma FxLMS. Sehingga, derau suara e_k menjadi



Gambar 4-3 Diagram blok sistem ANC tapis FIR FXLMS

Sumber : Chang, YC.,et al, 2003: 417

$$e_k = x_k H_p P H_e M_p + W_k^t X_k H_s H_e M_p \tag{4.3}$$

dimana parameter-parameter bobot dan runtun derau masukannya adalah

$$W_k = [w_{0k}, w_{1k}, \cdots, w_{Nk}]^T$$

$$X_k = [x_k, x_{k-1}, \cdots, x_{k-N}]^T . \tag{4.4}$$

Estimasi gradien yang sesuai dari sistem adaptif pada waktu sampel ke-*k* diberikan oleh persamaan (4.5) sebagai berikut:

$$\hat{\nabla} = \frac{\partial e_k^2}{\partial W_k} = 2e_k \left(X_k H_s H_e M_p \right) \tag{4.5}$$

dimana parameter-parameter bobot yang diperbaharui dengan algoritma LMS menjadi

$$W_{k+1} = W_k - 2\mu\, e_k \left(X_k H_s H_e M_p \right). \tag{4.6}$$

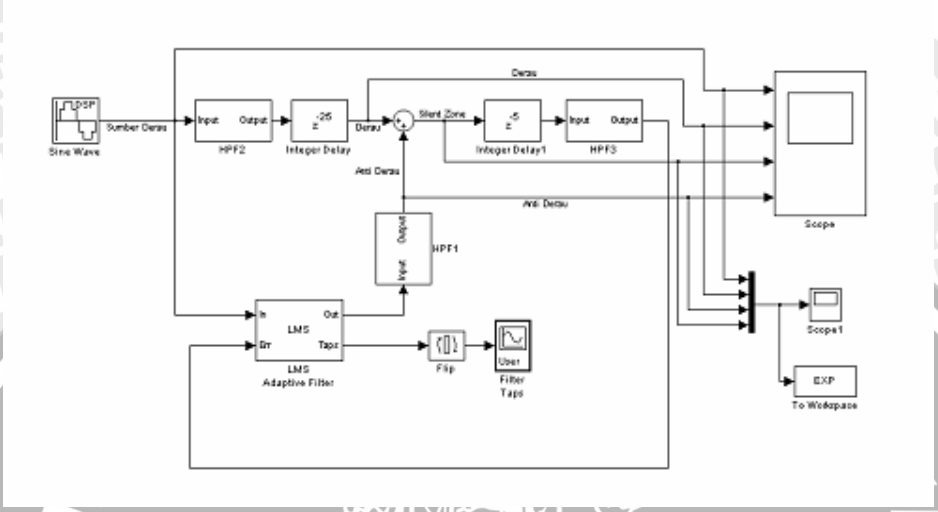
Persamaan (4.6) berbeda dari algoritma LMS yang biasa:

$$W_{k+1} = W_k - 2\mu\, e_k X_k \tag{4.7}$$

Seperti yang terlihat dalam persamaan (4.6) dan (4.7), maka hanya berbeda pada bagian koreksinya. Dimana suku koreksi tersebut sama dengan keluaran dari runtun derau X_k yang datang melalui tapis komposit $H_s H_e M_p$. Kondisi tersebut yang disebut algoritma FXLMS. Dan digunakan μ sebagai penentu laju (step size) perubahan bobot parameter.

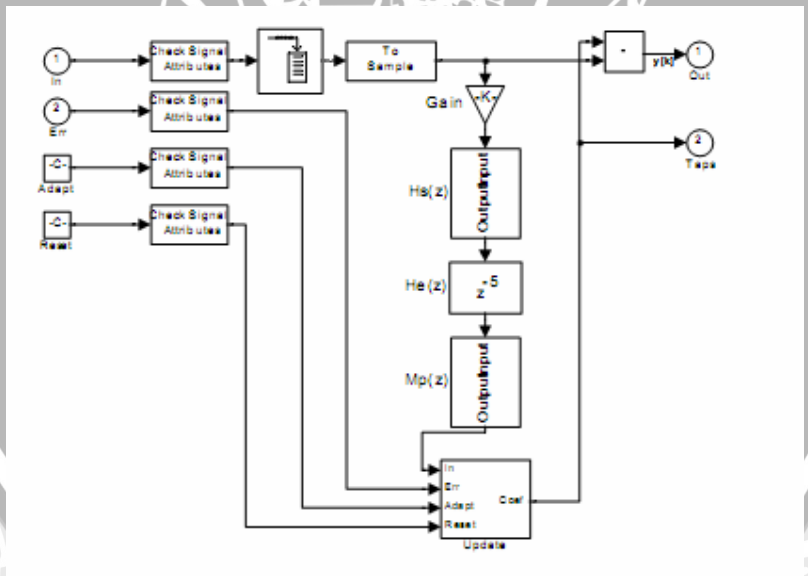
Selanjutnya berdasarkan diagram blok pada Gambar 4.3 maka disusunlah blok-blok diagram SIMULINK, seperti yang terlihat dalam Gambar

4.4. Untuk blok LMS dalam Gambar 4.4 harus ditambahkan bagian koreksi ($H_s H_e M_p$) sebagai realisasi dari algoritma FXLMS terlihat dalam Gambar 4.5.



Gambar 4-4 Diagram blok sistem ANC dengan SIMULINK menggunakan tapis FIR orde 21 FX-LMS

Sumber : Perancangan



Gambar 4-5 Penambahan bagian koreksi $H_s H_e M_p$ pada blok LMS

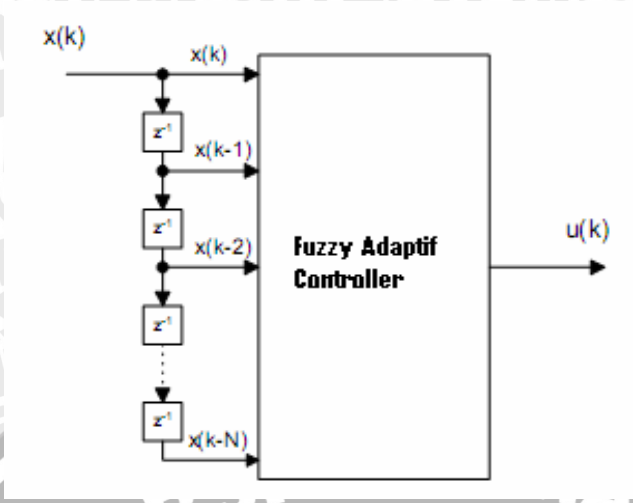
Sumber : Perancangan

4.3 Algoritma Fuzi Adaptif Untuk Sistem ANC

Biasanya, ANC di rancang berdasarkan model matematis yang bersifat linear. Model seperti ini sulit digunakan untuk mengontrol derau dalam lingkungan *plant* yang tak-linear. Hal inilah yang menyebabkan performansi reduksi derau pada frekuensi *broadband* menjadi turun dibanding reduksi derau pada frekuensi *narrowband*. Untuk mengatasi hal ini pada sistem ANC digunakan model matematis yang rumit untuk merancang bagian pengontrol. Misalnya pada FXLMS ANC konvensional sering digunakan beratus-ratus fungsi bobot untuk mengendalikan sinyal gangguan. Sehingga berpotensi menimbulkan kesalahan numeris seperti kesalahan pembulatan dan kesalahan kuantisasi (Chang, YC., et al, 2003).

Sebelum melangkah pada perancangan fuzi adaptif, maka akan dibuat dengan MATLAB fungsi-fungsi plantnya yang dimodelkan oleh tapis *butterworth* orde 2 dengan frekuensi *cut-off* 80Hz, seperti yang diperlihatkan dalam bagan berikut

```
function [y,yOld,xOld]=HPF(x,yOld,xOld)
% x = input
% A = Denumerator
% B = Numerator
%
A =  [1.0000   -1.6475    0.7009];
B =  [0.8371   -1.6742    0.8371];
y =  B(1,1)*x + B(1,2)*xOld(1,1) + B(1,3) * xOld(1,2) - ...
     A(1,2)*yOld(1,1) - A(1,3)*yOld(1,2);
yOld(1,2)=yOld(1,1);
yOld(1,1)=y;
xOld(1,2)=xOld(1,1);
xOld(1,1)=x;


function [y,yOld]=delay(x,yOld)
% x = input
% yOld = jumlah tunda
%
[LxOld,WxOld]=size(yOld);
y=yOld(1,WxOld);
for k=1:WxOld-1,
   yOld(1,WxOld-k+1) = yOld(1,WxOld-k);
end
yOld(1,1) = x;
```



Gambar 4-6 Skema umum masukan-keluaran pada fuzi adaptif

Sumber : perancangan

Skema umum perancangan fuzi adaptif ditunjukkan dalam Gambar 4-6. Secara umum masukan terdiri dari nilai tunda dari masukan yang diinginkan.

Berikut ini dijelaskan langkah-langkah untuk merancang sistem ANC dengan fuzi adaptif *filtered-x* :

Langkah 1 : Keluaran tapis FIR *orde* N+1 diperoleh dari rerata N+1 cuplikan masukan sinyal *noise,* dari sini dapat digunakan masukan sebanyak N+1 cuplikan yang telah lalu untuk merancang tapis *fuzzy*. Tapis fuzi *orde* N+1 dalam sistem ANC dapat didefinisikan dengan *fuzi set* Fi sebanyak M untuk setiap cuplikan data masukan *noise* x_{k-1} dengan fungsi keanggotaan tipe *Gaussian*:

$$\mu_{F_i^l}(x_{k-1}) = \exp\left[-\frac{1}{2}\left(\frac{x_{k-1}-\bar{x}_{i,k}^{1}}{\sigma_{x_i}^{1}}\right)^2\right] \tag{4.8}$$

dengan l =1,2,…,M dan i=0,1,2,…,N, sedangkan k adalah nomor cuplikan. Pada persamaan, $\bar{x}_i^1$ adalah pusat *(center)* dan $\sigma_{x_i}^1$ adalah standar deviasi dari fungsi *Gaussian*. Pada skripsi ini digunakan M=5, dan N=20 dan nilai awal standar deviasi sebesar 0,2. Sehingga setiap cuplik masukan x_{k-i} memiliki lima label linguistik yang terdistribusi pada jangkauan masukan antara [–1 1]; NB *(negative big), NS(negative small), AZ (allmost zero),* PS *(positive small),* dan PB *(positive*

big).

Langkah 2 : Mendefinisikan M aturan fuzi *IF-THEN* dengan tipe seperti berikut:

$$Rule^{l}: IF\ x_k\ is\ F_N^1\ and\ ....and$$

$$x_{k-N}\ is\ F_N^1\ THAN\ y_k\ is\ U^l \qquad (4.9)$$

l =1,2,…,M. Pada persamaan, F_i^l seperti yang didefinisikan sebelumnya dan set U^l berupa *fuzzy singleton* pada θ_k^1. Untuk nilai *crisp* pertama nilai θ_k^1 dapat dipilih sembarang antara –1 sampai dengan 1, sedangkan aturan *fuzzy* diberikan pada tahap berikutnya. Tujuh aturan *fuzzy* membentuk tapis *fuzzy* FIR, yang berlaku sebagai sebuah tapis anti-*noise* untuk meredam gangguan yang tidak diinginkan. Basis aturan dari sistem ini ditunjukkan dalam Tabel 1.

Tabel 1. Aturan fuzi pada sistem ANC

	Antecedence				*Consequents*
	x_k	x_{k-1}	...	x_{k-N}	y_k
Rule 1	NB	NB	NB	NB	NB
Rule 2	NS	NS	NS	NS	NS
Rule 3	AZ	AZ	AZ	AZ	AZ
Rule 4	PS	PS	PS	PS	PS
Rule 5	PB	PB	PB	PB	PB

Langkah 3 : pada tahap ini diberikan ilustrasi mengenai proses pengambilan keputusan *(fuzzy inference)* dan proses *defuzzifikasi.* Pada *fuzzy inference,* digunakan prinsip logika *fuzzy* yang dikombinasikan dengan aturan *fuzzy* yang memetakan masukan *fuzzy set* ke keluaran *fuzzy set*. Misal, pada tapis *fuzzy* saat cuplikan ke k diberi masukan $X_k^* = \left[x_k^*, x_{k-1}^*, ..., x_{k-N}\right]$. Tapis *fuzzy* ini

menggunakan mesin *inference* yang minimal dan *defuzzifikasi* tipe *centroid,* yang diperoleh dengan mengkombinasikan aturan sebanyak *M* yang didefinisikan pada tahap sebelumnya. Tapis ini di beri nama $f(X_k^*)$, dalam hal ini digunakan u_k untuk mengontrol sinyal yang tidak diinginkan.

Langkah 4 : Untuk meminimisasi daya sinyal gangguan, algoritma *fuzzy filtered-x* LMS menggunakan formula sebagai berikut, untuk memperbarui bobot parameter θ_k^1, $\bar{x}_{i,k}^1$, dan $\sigma_{xi,k}^1$ pada setiap waktu cuplikan ke-*k*:

$$\theta_{k+1}^l = \theta_k^l + \lambda . e_k . \frac{p_k^l}{q_k} \tag{4.10}$$

$$\bar{x}_{i,k+1}^l = \bar{x}_{i,k}^l + \lambda . e_k . \gamma_k^l \frac{\left(x_{k-1} - \bar{x}_{i,k}^l\right)}{\left(\sigma_{xi,k}^i\right)^2} \tag{4.11}$$

$$\sigma_{i,k+1}^l = \sigma_{i,k}^l + \lambda . e_k . \gamma_k^l \frac{\left(x_{k-1} - \bar{x}_{i,k}^l\right)^2}{\left(\sigma_{xi,k}^i\right)^3} \tag{4.12}$$

dengan

$$p_k^l = \min\left(\exp\left[-\frac{1}{2}\left(\frac{x_{k-1} - \bar{x}_{i,k}^l}{\sigma_{xi,k}^l} \right)^2 \right] \right) \tag{4.13}$$

$$q_k = \sum_{l=1}^{7} p_k^l \tag{4.14}$$

$$\gamma_k^l = \frac{\theta_k^l + u_k}{q_k} . p_k^l \tag{4.15}$$

dan λ, adalah sebuah konstanta bilangan positif kecil. Bentuk terakhir pada persamaan diatas serupa dengan masukan gangguan x_k yang dikirimkan pada tapis *fuzzy*. Jadi persamaan tersebut disebut sebagai algoritma LMS *fuzzy filtered-x*, yang berbeda dengan algoritma LMS *filtered-x* biasa pada bagian faktor koreksi

dan algoritma *fuzzy*. Pada pengkajian ini digunakan nilai awal parameter sebagai berikut :

$$\bar{x}_{i,0}^{l} = x_{k-i} \; dan \bar{x}_{i,0}^{l} \neq x_{k-i} \; untuk l \neq 1$$

dan dipilih

$$\theta_0^1 = -x_k . \frac{H_p P}{H_s} \tag{4.16}$$

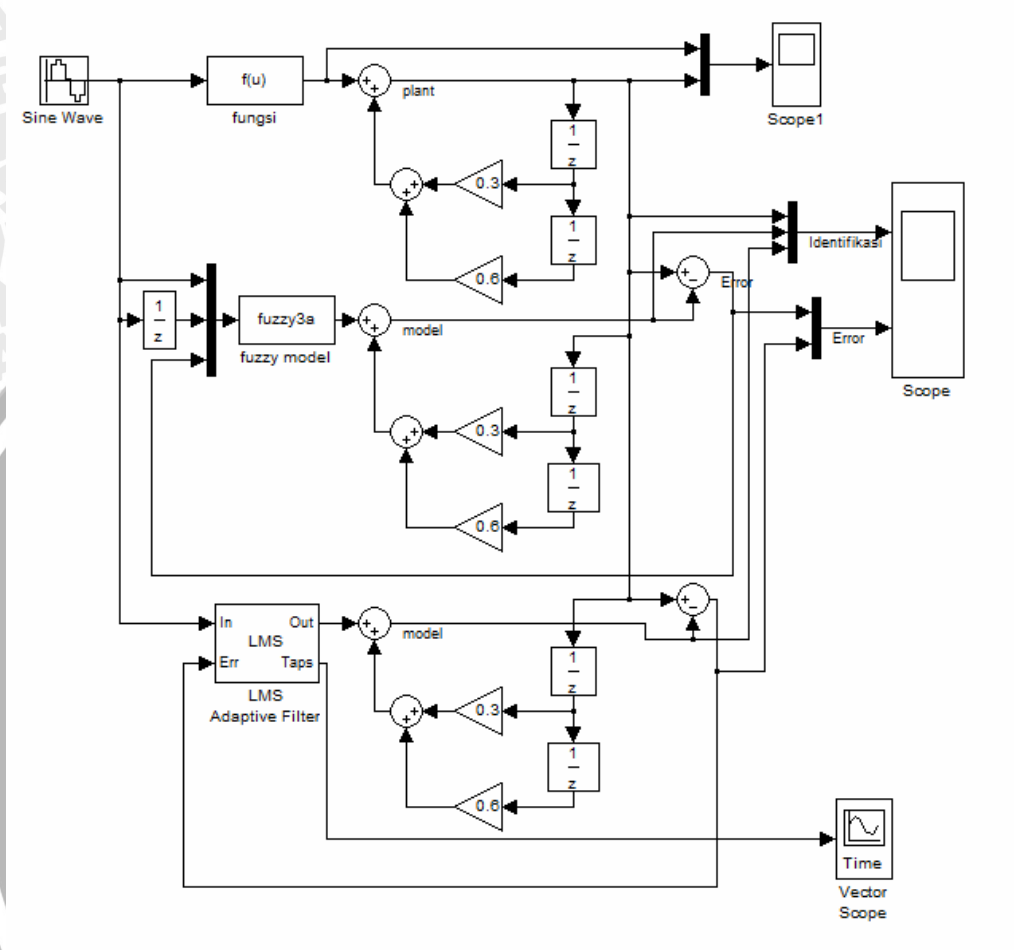
selanjutnya $\theta_0^1 (l \neq 1)$ nol *(zero)*. Jadi dengan persamaanpersamaan di atas *anti-noise* u_k akan sama dengan θ_0^1. Setelah speaker kedua *Hs* aktif, *noise* buatan menjadi - $x_k H_p P$, yang akan mengkompensasi *noise* yang tidak diinginkan pada daerah tenang *(silent zone)*.

Selanjutnya pada persamaan (4.16) dapat diterapkan fungsi alih H_eM_p pada bagian belakang sehingga diperoleh :

$$\theta_0^1 = -x_k . \frac{H_p P H_e M_p}{H_s H_e M_p} \tag{4.17}$$

sebagai *numerator* adalah $H_pPH_eM_p$ dan *denumeratornya* adalah $H_pPH_eM_p$. Keduanya merupakan fungsi alih antara sinyal elektris masukan dengan sinyal elektris keluaran. Dalam hal ini dapat digunakan *white-noise* sabagai sinyal masukan pada *speaker* utama H_p dan kemudian mengumpankan pada *microphone* M_p pada bagian akhir sistem untuk mengukur *numerator*, melalui lintasan udara. Tanggapan keluaran terhadap masukan adalah numerator pada persamaan (4.17). Sehingga dapat digunakan masukan sinyal *white-noise* sebagai masukan bagi H_s dan mengukur sisa *noise* dengan *microphone* M_p. Bagian denumerator pada persamaan tersebut juga dapat ditentukan. Sehingga dapat diperoleh persamaan (4.16) dan *plant* tidak lagi diperlukan.

Berikut adalah blok diagram penggunaan fuzi adaptif dengan tiga *Membership Functions* :



Gambar 4-7 Blok diagram pengguanaan fuzi adaptif

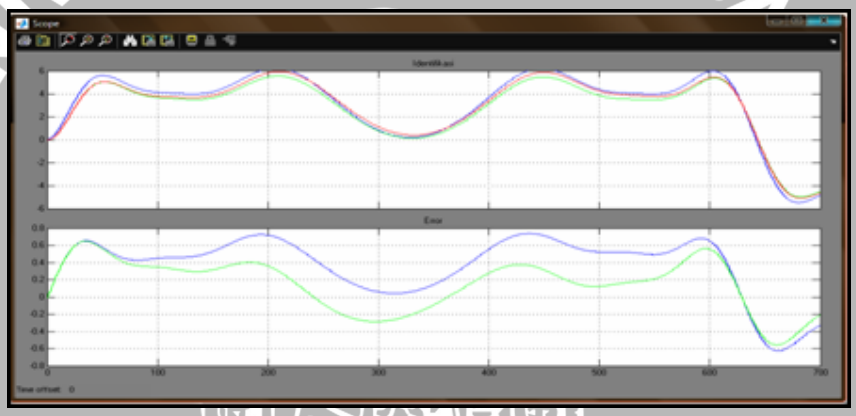
Sumber : perancangan

BAB V

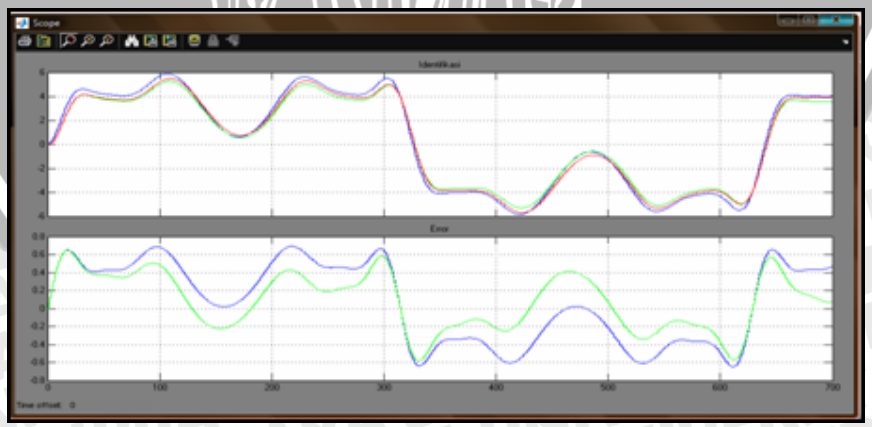
HASIL PENGUJIAN

5.1 Pengujian Fuzi Adaptif

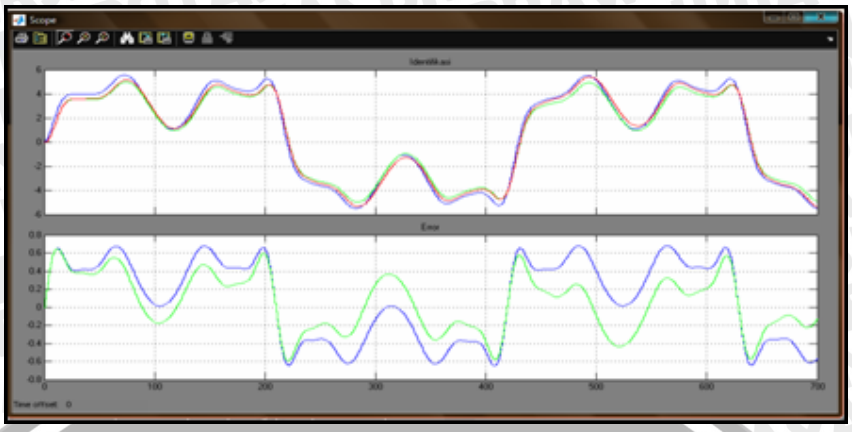
Pada pengujian ini dibangkitkan sinyal noise dengan variasi frekuensi tertentu mulai dari 50 s/d 500 Hz. Pengujian ini dilakukan untuk mengetahui pengaruh perubahan variasi frekuensi terhadap keluaran dari fuzi adaptif yang telah dirancang.



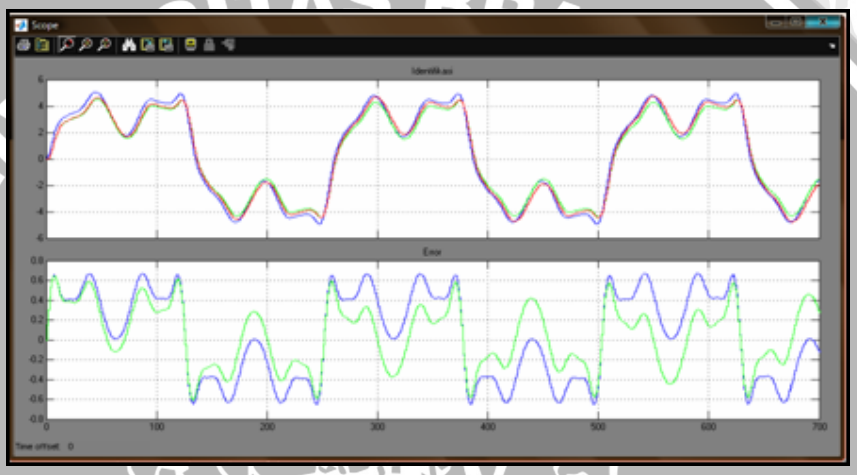
Gambar 5-1 Hasil pengujian menggunakan variasi frekuensi 50 Hz



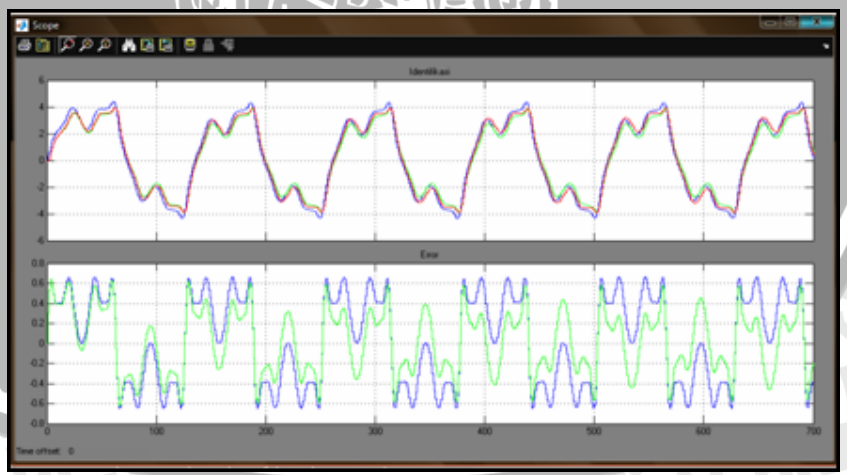
Gambar 5-2 Hasil pengujian menggunakan variasi frekuensi 100 Hz



Gambar 5-3 Hasil pengujian menggunakan variasi frekuensi 150 Hz



Gambar 5-4 Hasil pengujian menggunakan variasi frekuensi 250 Hz

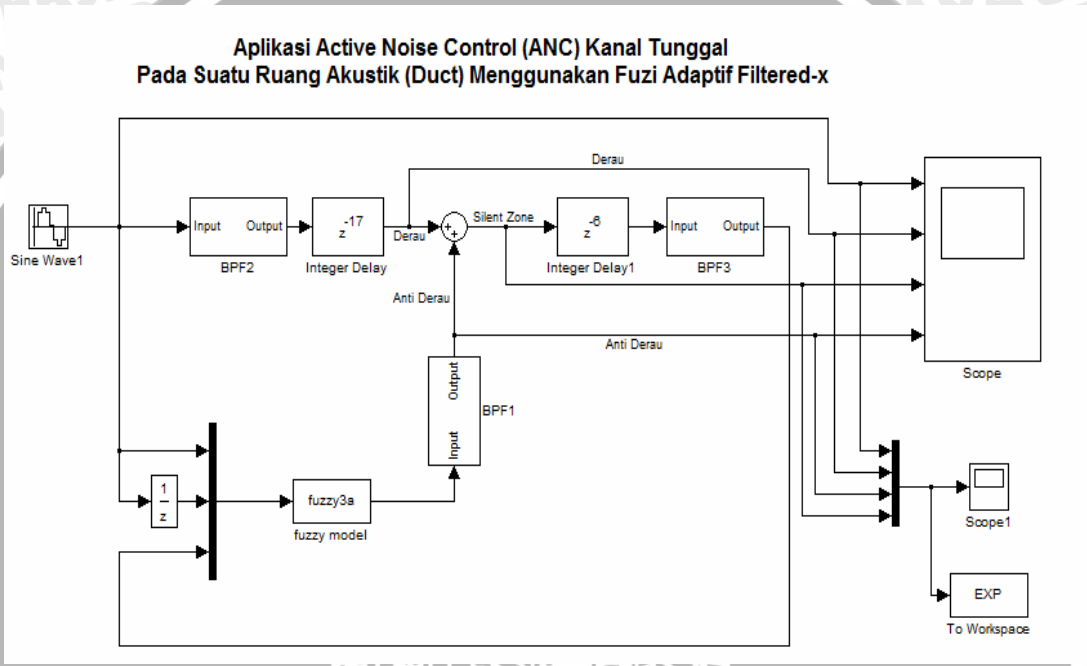


Gambar 5-5 Hasil pengujian menggunakan variasi frekuensi 500 Hz

Berdasarkan kelima gambar diatas dapat disimpulkan bahwa data keluaran dari identifikasi fuzi adaptif mengikuti data masukan. Variasi frekuensi mulai 50Hz hingga 500Hz menunjukkan bahwa gambar data masukan hampir

berimpitan dengan data identifikasi dari fuzi adaptif. Hal ini disebabkan dalam metode fuzi adaptif memiliki kemampuan untuk mengatur parameter-parameter bobotnya secara otomatis. Oleh karena itu, permasalahan atas kelemahan pada kecepatan konvergensi dapat diatasi.

5.2 Pengujian Fuzi Adaptif filtered-x untuk sistem ANC

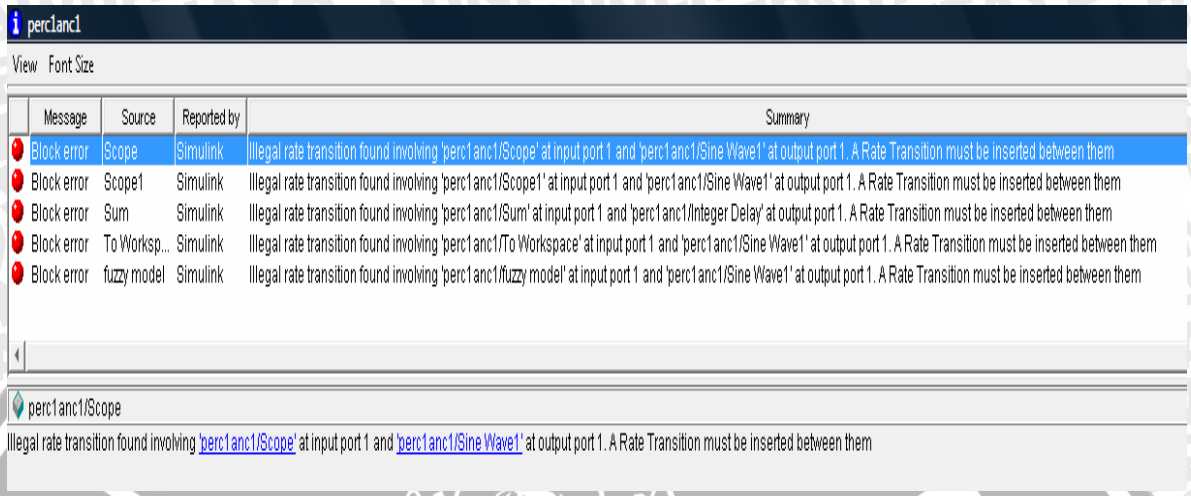


Gambar 5-6 Blok diagram fuzi adaptif pada sebuah *duct Active Noise Control*

Setelah menguji fuzi adaptif yang disusun dari *s-functions*, maka dirangkaikan dengan skema umum dalam sistem *Active Noise Control* seperti yang telah dirancang pada gambar 5-6.

Pengujian dilakukan untuk mendapatkan data kecepatan dan kuantitas derau yang dapat dihilangkan oleh fuzi adaptif di dalam sebuah ruang akustik kanal tunggal.

Setelah program dieksekusi menggunakan Matlab, maka muncul keterangan sebagai berikut :



Gambar 5-7 Kotak dialog yang muncul saat eksekusi fuzi adaptif ANC

Berdasarkan informasi pada gambar 5-7, maka dapat dijelaskan bahwa fuzi adaptif dalam sistem *Active Noise Control* tidak dapat berjalan dengan baik. Hal ini disebabkan karena beberapa hal yaitu adanya *source* yang berbentuk *sine wave,* walaupun sudah berbentuk sinyal diskrit tetapi tidak dapat menghasilkan keluaran yang diinginkan. Hal ini disebabkan proses identifikasi source yang kurang optimal karena pada saat di eksekusi dalam model fuzi adaptif biasa dapat berjalan dengan normal.

Penyebab selanjutnya adalah hasil keluaran dari fuzi adaptif tidak dapat diolah langsung pemodelan speaker antiderau. Maka dibutuhkan sebuah model tertentu untuk menghubungkan sinyal keluaran fuzi adaptif dapat diolah dengan sempurna.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Penerapan fuzi adaptif dalam berbagai macam variasi frekuensi masukan menghasilkan karakteristik sinyal keluaran yang menyerupai sinyal masukan. Hal ini disebabkan dalam metode fuzi adaptif memiliki kemampuan untuk mengatur parameter-parameter bobotnya secara otomatis. Oleh karena itu, permasalahan atas kelemahan pada kecepatan konvergensi dapat diatasi.

Pengujian fuzi adaptif dalam sistem *Active noise control* belum bisa berjalan dengan semestinya disebabkan pengaruh *source* dan tidak adanya model identifikasi speaker anti derau yang sesuai dengan sinyal hasil keluaran fuzi adaptif.

6.2 Saran

Untuk menghasilkan data pengujian yang optimal, maka dibutuhkan banyak literature yang berisi tentang informasi tentang identifikasi *source* yang tepat sebagai masukan pemodelan identifikasi *Active Noise Control*.

DAFTAR PUSTAKA

Hellgren, J., 2002, Analysis of feedback cancellation in hearing aids with filtered-X LMS and direct method of closed loop identification, IEEE Trans. Speed Audio Process, 10,(2), pp. 119-131.

Jang, J.-S.R, dan Sun., C.T, dan Mizutani, E, 1997, Neuro-Fuzzy and Soft-Computing, Prentice-Hall International, Inc., USA, 1997.

Park, Y.M. and Lee, K.Y., 1996, A self-organizing power system stabilizer using fuzzy auto-regressive moving average (FARMA) model, IEEE Trans., Energy Convers, 11, (1), pp. 442-448.

Sen M Kuo dan Dennis R. Morgan, 1999, Active Noise Control : A Tutorial Review, Proceedings of the IEEE, Vol. 87, No. 6, June 1999

Widrow, B., dan Steams, S.D., 1985, Adaptive signal processing, Prentice-Hall, New-Jersey, 1985

Wang, L.X., dan Mendel, J.M., 1993, Fuzzy adaptive filter, with application to nonlinear channel equalization, IEEE Trans. Fuzzy Syst., 1,(3),pp. 161-170.

Lampiran 1

Listing Program *s-functions* Fuzi Adaptif 3MF

```
/*  File    : dsfunc.c
 *  Abstract:
 *      Example C-file S-function for defining a discrete system.
 *
 *      x(n+1) = Ax(n) + Bu(n)
 *      y(n)   = Cx(n) + Du(n)
 *
 *      For more details about S-functions, see simulink/src/sfuntmpl_doc.c.
 *
 *  Copyright 1990-2002 The MathWorks, Inc.
 *  $Revision: 1.12 $
 */


#define S_FUNCTION_NAME fuzzy3a
#define S_FUNCTION_LEVEL 2
#include "simstruc.h"
#include "math.h"
#define U(element) (*uPtrs[element])  /* Pointer to Input Port0 */


static real_T A[1][1]={ { 0.000000000000001  } ,
                };
static real_T B[1][1]={ { 1.0  } ,
                };
```

repository.ub.ac
UNIVERSITAS BRAWIJAYA

```
/*====================*
 * S-function methods *
 *====================*/

/* Function: mdlInitializeSizes ===============================================
 * Abstract:
 *    The sizes information is used by Simulink to determine the S-function
 *    block's characteristics (number of inputs, outputs, states, etc.).
 */
static void mdlInitializeSizes(SimStruct *S)
{
   ssSetNumSFcnParams(S, 0);  /* Number of expected parameters */
   if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S))
     {
      return; /* Parameter mismatch will be reported by Simulink */
     }

   ssSetNumContStates(S, 0);
   ssSetNumDiscStates(S, 83);

   if (!ssSetNumInputPorts(S, 1)) return;
   ssSetInputPortWidth(S, 0, 3);
   ssSetInputPortDirectFeedThrough(S, 0, 1);

   if (!ssSetNumOutputPorts(S, 1)) return;
   ssSetOutputPortWidth(S, 0, 1);

   ssSetNumSampleTimes(S, 1);
   ssSetNumRWork(S, 0);
```

```
    ssSetNumIWork(S, 0);
    ssSetNumPWork(S, 0);
    ssSetNumModes(S, 0);
    ssSetNumNonsampledZCs(S, 0);

    /* Take care when specifying exception free code - see sfuntmpl_doc.c */
    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}

/* Function: mdlInitializeSampleTimes =========================================
 * Abstract:
 *    Specifiy that we inherit our sample time from the driving block.
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, 0.01);
    ssSetOffsetTime(S, 0, 0.0);
}
```

```
#define MDL_INITIALIZE_CONDITIONS
/* Function: mdlInitializeConditions ========================================
 * Abstract:
 *    Initialize both discrete states to one.
 */
static void mdlInitializeConditions(SimStruct *S)
{
  real_T *x0 = ssGetRealDiscStates(S);
  int_T  lp;

  /*=== center membership error ===*/
  for (lp=0;lp<1;lp++)
     {
      *x0++=10;
     }
  for (lp=1;lp<2;lp++)
     {
      *x0++=0;
     }
  for (lp=2;lp<3;lp++)
     {
      *x0++=-10;
     }
  /*=== center membership delta-error ===*/
  for (lp=3;lp<4;lp++)
     {
      *x0++=10;
     }
```

```
for (lp=4;lp<5;lp++)
  {
  *x0++=0;
  }
for (lp=5;lp<6;lp++)
  {
  *x0++=-10;
  }
/* lebar membership error */
for (lp=6;lp<7;lp++)
  {
  *x0++=7.5;
  }
for (lp=7;lp<8;lp++)
  {
  *x0++=7.5;
  }
for (lp=8;lp<9;lp++)
  {
  *x0++=7.5;
  }

/*=== lebar membership delta-error ===*/
for (lp=9;lp<10;lp++)
  {
  *x0++=7.5;
  }
```

repository.ub.ac
UNIVERSITAS BRAWIJAYA

```
for (lp=10;lp<11;lp++)
    {
    *x0++=7.5;
    }
for (lp=11;lp<12;lp++)
    {
    *x0++=7.5;
    }
/* === yl === */
for (lp=12;lp<13;lp++)
    {
    *x0++=680000;
    }
for (lp=13;lp<14;lp++)
    {
    *x0++=0;
    }
for (lp=14;lp<15;lp++)
    {
    *x0++=-680000;
    }
/* === sisa === */
for (lp=15;lp<28;lp++)
    {
    *x0++=0.0;
    }
}
/* Function: mdlOutputs
=============================================
```

repository.ub.ac
UNIVERSITAS BRAWIJAYA

```
* Abstract:
*      y = Cx + Du
*/
static void mdlOutputs(SimStruct *S, int_T tid)
{
  real_T            *y    = ssGetOutputPortRealSignal(S,0);
  real_T            *x    = ssGetRealDiscStates(S);
  InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);

  //UNUSED_ARG(tid); /* not used in single tasking mode */

  x[15]=exp(-((U(0)*x[0])/(x[6]))*((U(0)*x[0])/(x[6])));
  x[16]=exp(-((U(0)*x[1])/(x[7]))*((U(0)*x[1])/(x[7])));
  x[17]=exp(-((U(0)*x[2])/(x[8]))*((U(0)*x[2])/(x[8])));

  x[18]=exp(-((U(1)*x[3])/(x[9])) *((U(1)*x[3])/(x[9] )));
  x[19]=exp(-((U(1)*x[4])/(x[10]))*((U(1)*x[4])/(x[10])));
  x[20]=exp(-((U(1)*x[5])/(x[11]))*((U(1)*x[5])/(x[11])));

  // 3 Rule
  x[21]=x[15]*x[18];
  x[22]=x[16]*x[19];
  x[23]=x[17]*x[20];

  // Rule*bobot output (a)
  x[24]=x[15]*x[18]*x[12];
  x[25]=x[16]*x[19]*x[13];
  x[26]=x[17]*x[20]*x[14];
```

```
    x[27]=x[24]+x[25]+x[26];      /* (a) */
    x[28]=x[21]+x[22]+x[23];      /* (b) */

    y[0]=x[27]/x[28];             /* f=a/b */

}


#define MDL_UPDATE
/* Function: mdlUpdate
=============================================================

 * Abstract:
 *      xdot = x + (-error(1-Yout)*Yout*x)*Alfa
 */
static void mdlUpdate(SimStruct *S, int_T tid)
{
   real_T            tempX[15] = {0.0, 0.0, 0.0,
                                  0.0, 0.0, 0.0,
                                  0.0, 0.0, 0.0,
                                  0.0, 0.0, 0.0,
                                  0.0, 0.0, 0.0};
   real_T            *x      = ssGetRealDiscStates(S);
   real_T            *y    = ssGetOutputPortRealSignal(S,0);
   InputRealPtrsType uPtrs    = ssGetInputPortRealSignalPtrs(S,0);
```

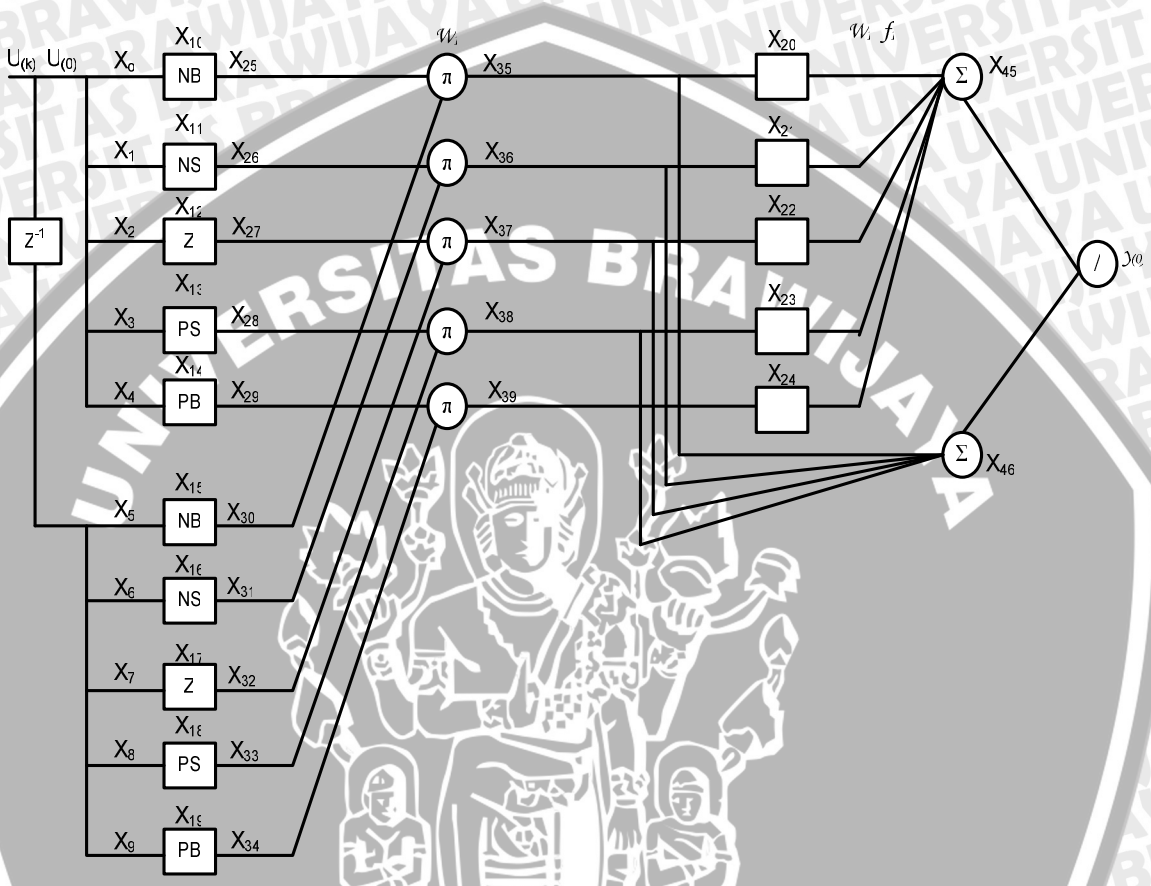
```
/* up-date center membership error */
/*
tempX[0]=x[0]-(A[0][0]*(U(0)/x[28])*(x[12]-U(2))*x[21]*(2*(U(0)-x[0])/(x[6]*x[6])));
tempX[1]=x[1]-(A[0][0]*(U(0)/x[28])*(x[13]-U(2))*x[22]*(2*(U(0)-x[1])/(x[7]*x[7])));
tempX[2]=x[2]-(A[0][0]*(U(0)/x[28])*(x[14]-U(2))*x[23]*(2*(U(0)-x[2])/(x[8]*x[8])));
*/

tempX[0]=x[0]-(A[0][0]*(U(2)/x[28])*(x[12]-y[0])*x[21]*(2*(U(0)-x[0])/(x[6]*x[6])));
tempX[1]=x[1]-(A[0][0]*(U(2)/x[28])*(x[13]-y[0])*x[22]*(2*(U(0)-x[1])/(x[7]*x[7])));
tempX[2]=x[2]-(A[0][0]*(U(2)/x[28])*(x[14]-y[0])*x[23]*(2*(U(0)-x[2])/(x[8]*x[8])));

x[0]=tempX[0];
x[1]=tempX[1];
x[2]=tempX[2];

/* up-date center membership delta-error */
/*
tempX[3]=x[3]-(A[0][0]*(U(0)/x[28])*(x[12]-U(2))*x[21]*(2*(U(1)-x[3])/(x[9]*x[9])));
tempX[4]=x[4]-(A[0][0]*(U(0)/x[28])*(x[13]-U(2))*x[22]*(2*(U(1)-x[4])/(x[10]*x[10])));
tempX[5]=x[5]-(A[0][0]*(U(0)/x[28])*(x[14]-U(2))*x[23]*(2*(U(1)-x[5])/(x[11]*x[11])));
*/
  tempX[3]=x[3]-(A[0][0]*(U(2)/x[28])*(x[12]-y[0])*x[21]*(2*(U(1)-x[3])/(x[9]*x[9])));
tempX[4]=x[4]-(A[0][0]*(U(2)/x[28])*(x[13]-y[0])*x[22]*(2*(U(1)-x[4])/(x[10]*x[10])));
tempX[5]=x[5]-(A[0][0]*(U(2)/x[28])*(x[14]-y[0])*x[23]*(2*(U(1)-x[5])/(x[11]*x[11])));

x[3]=tempX[3];
x[4]=tempX[4];
x[5]=tempX[5];
```

```
/* up-date lebar membership error */

/*

tempX[6]=x[6]-(A[0][0]*(U(0)/x[28])*(x[12]-U(2))*x[21]*(2*((U(0)-x[0])*(U(0)-
x[0]))/(x[6]*x[6]*x[6]));

tempX[7]=x[7]-(A[0][0]*(U(0)/x[28])*(x[13]-U(2))*x[22]*(2*((U(0)-x[1])*(U(0)-
x[1]))/(x[7]*x[7]*x[7]));

tempX[8]=x[8]-(A[0][0]*(U(0)/x[28])*(x[14]-U(2))*x[23]*(2*((U(0)-x[2])*(U(0)-
x[2]))/(x[8]*x[8]*x[8]));

*/


tempX[6]=x[6]-(A[0][0]*(U(2)/x[28])*(x[12]-y[0])*x[21]*(2*((U(0)-x[0])*(U(0)-
x[0]))/(x[6]*x[6]*x[6]));

tempX[7]=x[7]-(A[0][0]*(U(2)/x[28])*(x[13]-y[0])*x[22]*(2*((U(0)-x[1])*(U(0)-
x[1]))/(x[7]*x[7]*x[7]));

tempX[8]=x[8]-(A[0][0]*(U(2)/x[28])*(x[14]-y[0])*x[23]*(2*((U(0)-x[2])*(U(0)-
x[2]))/(x[8]*x[8]*x[8]));



x[6]=tempX[6];

x[7]=tempX[7];

x[8]=tempX[8];



/* up-date lebar membership delta-error */

/*

tempX[9]=x[9]-(A[0][0]*(U(0)/x[28])*(x[12]-U(2))*x[21]*(2*((U(1)-x[3])*(U(1)-
x[3]))/(x[9]*x[9]*x[9]));

tempX[10]=x[10]-(A[0][0]*(U(0)/x[28])*(x[13]-U(2))*x[22]*(2*((U(1)-x[4])*(U(1)-
x[4]))/(x[10]*x[10]*x[10]));

tempX[11]=x[11]-(A[0][0]*(U(0)/x[28])*(x[14]-U(2))*x[23]*(2*((U(1)-x[5])*(U(1)-
x[5]))/(x[11]*x[11]*x[11]));

*/


tempX[9] =x[9] -(A[0][0]*(U(2)/x[28])*(x[12]-y[0])*x[21]*(2*((U(1)-x[3])*(U(1)-
x[3]))/(x[9]*x[9]*x[9]));
```

```
    tempX[10]=x[10]-(A[0][0]*(U(2)/x[28])*(x[13]-y[0])*x[22]*(2*((U(1)-x[4])*(U(1)-
x[4]))/(x[10]*x[10]*x[10])));

    tempX[11]=x[11]-(A[0][0]*(U(2)/x[28])*(x[14]-y[0])*x[23]*(2*((U(1)-x[5])*(U(1)-
x[5]))/(x[11]*x[11]*x[11])));

    x[9]=tempX[9];

    x[10]=tempX[10];

    x[11]=tempX[11];

    /* up-date yl */

    tempX[12]=x[12]-(A[0][0]*(U(2)/x[28])*x[21]);

    tempX[13]=x[13]-(A[0][0]*(U(2)/x[28])*x[22]);

    tempX[14]=x[14]-(A[0][0]*(U(2)/x[28])*x[23]);

    x[12]=tempX[12];

    x[13]=tempX[13];

    x[14]=tempX[14];

}


/* Function: mdlTerminate
=============================================================

 * Abstract:

 *    No termination needed, but we are required to have this routine.

 */

static void mdlTerminate(SimStruct *S)

{

    UNUSED_ARG(S); /* unused input argument */

}

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */

#include "simulink.c"      /* MEX-file interface mechanism */

#else

#include "cg_sfun.h"       /* Code generation registration function */

#endif
```

Lampiran 2

Skema Fuzi Adaptif 3MF



Layer 1 Layer 2 Layer 3 Layer 4

$$\mu A_1(x) = \exp\left(-\left(\frac{x - C_1}{ai}\right)^2\right)$$

$$X_{25} = \exp\left(-\left(\frac{U_o . X_o}{X_{10}}\right)^2\right)$$

$W_1 = \mu A_{1(x)} . \; {}_{\mu}B_{1(x)}$

$X_{35} = X_{25} . \; X_{30}$

$O_1 = W_1 f_1$

$X_{40} = X_{35} . \; X_{20}$

$\overline{\omega}_1 = O_1(w_1 + w_2)$