



UNIVERSITAS INDONESIA

**PERANCANGAN DEKODER VSELP DENGAN
*POSTFILTERING***

SKRIPSI

ASTATINE SUNARDI

040503015X

**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DEPOK
JUNI 2009**

PERNYATAAN ORISINALITAS

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

Nama : Astatine Sunardi
NPM : 040503015X
Tanda Tangan :
Tanggal : 17 Juni 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Astatine Sunardi

NPM : 040503015X

Program Studi : Teknik Elektro

Judul Skripsi : Perancangan Dekoder VSELP dengan *Postfiltering*

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

DEWAN PENGUJI

Pembimbing : Dr. Fitri Yuli Zulkifli S.T.,M.Sc (.....)

Penguji : Dr.Ir. Arman D. Diponegoro (.....)

Penguji : Prof. Dr. Ir. Dadang Gunawan, M.Eng (.....)

Ditetapkan di : Depok

Tanggal : 2 Juli 2009

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, sehingga saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat mata kuliah untuk menyelesaikan program studi di Departemen Teknik Elektro Universitas Indonesia.

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

- (1) Dr. Fitri Yuli Zulkifli S.T.,M.Sc. selaku dosen pembimbing yang memberikan pengarahan dan bimbingan kepada saya dalam penyusunan skripsi ini;
- (2) Dr. Ir. Arman Djohan Diponegoro selaku dosen pembimbing kedua saya yang juga telah meluangkan waktu untuk memberikan saran dan bimbingan dalam penyelesaian skripsi ini;
- (3) orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
- (4) sahabat dan teman – teman saya, Lukas, Dayat, Shimon, dan Abhe yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu selanjutnya.

Depok, 16 Juni 2009

Penulis

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

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

Nama : Astatine Sunardi
NPM : 040503015X
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

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

Perancangan Dekoder VSELP dengan *Postfiltering*

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan 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 : 16 Juni 2009

Yang menyatakan

(Astatine Sunardi)

ABSTRAK

Nama : Astatine Sunardi
Program Studi : Teknik Elektro
Judul : Perancangan Dekoder VSELP dengan *Postfiltering*
Pembimbing : Dr. Fitri Yuli Zulkifli S.T.,M.Sc

Skripsi ini membahas mengenai perancangan dekoder pada pengkodean suara VSELP. Proses dari pendekodean suara ini dimulai dari ekstraksi 27 parameter yang dikirimkan dari enkoder. Parameter – parameter ini kemudian diproses untuk mendapatkan sinyal sintesis. Namun, sinyal sintesis yang dihasilkan mengalami penurunan nilai SNR (*Signal to Noise Ratio*), tingkat *noise* yang tinggi mendekati *white noise* sehingga sangat sulit untuk menjaganya dibawah nilai ambang batas kemampuan suara yang dapat didengar yang dikarenakan terjadi penurunan laju bit. Untuk itu, diperlukannya *filtering* pada sinyal sintesis sehingga didapatkan kembali sinyal suara dengan kualitas yang baik. Dalam skripsi ini dilakukan simulasi pendekodean pada teknik pengkodean suara VSELP dengan bahasa pemrograman Simulink MATLAB. Dari hasil simulasi didapatkan sinyal keluaran dekoder mengalami perlambatan dalam pelafalan kata yang disebabkan karena perbedaan waktu pencuplikan antara sinyal suara asli dengan pencuplikan pada percobaan. Selain itu, proses simulasi membutuhkan waktu yang lama. Untuk mengatasinya algoritma proses dekoder harus dipersingkat, kemampuan prosesor yang cepat harus dipergunakan dan menggunakan bahasa pemrograman yang tidak dipengaruhi dengan waktu pencuplikan.

Kata kunci:
dekoder, sinyal sintesis, rekonstruksi sinyal

ABSTRACT

Name : Astatine Sunardi
Study Program : Electrical Engineering
Title : VSELP Decoder Design with *Postfiltering*
Supervisor : Dr. Fitri Yuli Zulkifli S.T.,M.Sc

This final project discusses about decoder design in VSELP speech coding. The process in decoder starts with the extraction of 27 parameter which is sent by the encoder. Then, these parameters is being processed to get synthesis speech signal. However, the synthesis speech has high noise level approaching white noise, which makes it difficult to keep under the audible threshold value due to the decrease of bit rate. Therefore, the synthesis speech signal has to be filtered to obtain good quality synthesis speech signal. In this final project, the simulation is made to simulate the decoding process in VSELP speech coding with Simulink MATLAB programming language. From the simulation, the synthesis speech signal experiencing slowing output in the word pronunciation which caused by the difference sampling time between the real speech signal dan the sampling time used in decoder processing. Furthermore, the simulation process needs long time. To overcome these problem, the decoder process algorithm should be shortened, the ability of the processor which used must be fast and use use programming language that is not influenced with sampling time.

Key note:
decoder, synthesis speech, signal reconstruction

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN ORISINALITAS.....	ii
PENGESAHAN.....	iii
UCAPAN TERIMA KASIH	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Penulisan	4
1.3 Batasan Masalah	4
1.4 Metode Penelitian	4
1.5 Sistematika Penulisan	4
BAB 2 PRINSIP DASAR TEKNIK PENGKODEAN SUARA VSELP	6
2.1 Pengkodean Suara	6
2.1.1 Dasar – dasar pengkodean suara.....	6
2.1.2 Konsep dasar produksi suara manusia.....	9
2.1.3 Pemodelan sistem produksi suara.....	11
2.2 Pengkodean Suara Analisis Sintesis.....	12
2.2.1 Filter jangka pendek	13
2.2.2 Filter jangka panjang (<i>long term filter</i>)	15
2.2.3.Eksitasi	16
2.3 Dasar Enkoder dan Dekoder Pada VSELP.....	16
2.3.1 Dasar Enkoder VSELP	16
2.3.2 Dasar Dekoder VSELP.....	18
2.4. Struktur Tabel Kode Eksitasi dan Adaptif pada VSELP.....	19
2.4.1 Tabel Kode Eksitasi.....	19
2.4.2 Tabel Kode Adaptif	20
2.4.3 Komposisi Parameter Dalam Bingkai	21

BAB 3 ALGORITMA PERANCANGAN RANGKAIAN DEKODER VSELP.....	23
3.1 Koefisien Prediksi Jangka Pendek	23
3.1.1 Interpolasi.....	25
3.1.2 Stabilitas koefisien refleksi.....	25
3.2 Energi Bingkai.....	26
3.3 Pendekodean Lag	22
3.4 Pendekodean <i>Codeword</i> Eksitasi.....	27
3.5 Pendekodean dan Transformasi GS, P0, P1 Menjadi Penguatan	27
3.6 Perhitungan Kombinasi Eksitasi dan Pembaharuan Keadaan Filter Jangka Panjang (<i>Long Term Filter State</i>).....	28
3.7 Adaptive Pitch Prefilter	29
3.8 Filter Sintesis	29
3.9 <i>Adaptive Spectral Postfilter</i>	30
BAB 4 SIMULASI DAN ANALISA PROSES DEKODER VSELP	33
4.1 Simulasi Proses Pengolahan Suara pada Rangkaian Dekoder VSELP.....	33
4.1.1 <i>Adaptive Pitch Prefilter</i>	35
4.1.2 Filter sintesis LPC, $A(z)$	37
4.1.3 <i>Postfilter</i> Spektral Adaptif	38
4.1.4 Filter <i>Post Emphasis</i>	39
4.1.5 Pengontrolan Penguatan	40
4.2 Hasil Simulasi	41
4.3 Analisa Proses Rangkaian Dekoder VSELP	43
BAB 5 KESIMPULAN	44
DAFTAR REFERENSI.....	45
DAFTAR PUSTAKA.....	49
LAMPIRAN.....	51

DAFTAR GAMBAR

Gambar 2.1 Perbandingan kualitas suara dan laju bit berbagai jenis pengkodean suara	8
Gambar 2.2 Diagram sistem produksi suara manusia.....	9
Gambar 2.3 Model produksi suara	10
Gambar 2.4 Koresponden antara sistem produksi suara manusia dengan sistem yang telah disederhanakan berbasis <i>time-varying filter</i>	12
Gambar 2.5 Pengkodean analisis-sintesis menggunakan LPC.....	13
Gambar 2.6 Penerapan filter LPC pada sinyal ucapan	13
Gambar 2.7 Blok diagram enkoder VSELP	18
Gambar 2.8 Blok diagram dekoder VSELP	19
Gambar 3.1. Algoritma Dekoder VSELP	24
Gambar 3.2. Proses Pembaharuan Kode Buku Adaptif	29
Gambar 3.3. Contoh respon frekuensi filter sintesis LPC termodifikasi $1/[1 - P(z/\alpha)]$ untuk nilai α yang berbeda.	31
Gambar 3.4. Respons frekuensi <i>postfilter</i> jangka pendek $[1 - \mu z^{-1}][1 - P(z/\beta)]/[1 - P(z/\alpha)]$ terhadap spektrum LPC dari Gambar 3.3.	32
Gambar 4.1 Sampel Suara ‘Saya’ dengan basis waktu	34
Gambar 4.2. Sampel Suara ‘Saya pergi ke kampus’ dengan basis waktu.....	34
Gambar 4.3. Proses penyaringan suara sintesis pada dekoder VSELP	35
Gambar 4.4. Spektrum kombinasi sinyal eksitasi, $ex(n)$	36
Gambar 4.5. Sinyal keluaran <i>adaptive pitch prefilter</i> pada simulasi dekoder VSELP	36
Gambar 4.6. Sinyal keluaran filter sintesis, $A(z)$	37
Gambar 4.7. Sinyal keluaran <i>postfilter</i> spektral adaptif	39
Gambar 4.8. Sinyal keluaran filter <i>post emphasis</i>	40
Gambar 4.9. Spektrum sinyal sintesis dekoder VSELP	41
Gambar 4.10. Sinyal sintesis dekoder VSELP	42
Gambar 4.11. Sinyal sintesis suara “saya pergi ke kampus” dekoder VSELP	43

DAFTAR TABEL

Tabel 2.1 Parameter dasar pengkodean VSELP	6
Tabel 2.2 Alokasi bit untuk setiap bingkai VSELP	6
Tabel 3.1 Koefisien pelembutan spektral, $w_p(i)$	33



BAB I PENDAHULUAN

1.1. Latar Belakang

Pada sistem komunikasi bergerak, penyedia layanan komunikasi terus menghadapi tantangan untuk dapat mengakomodasi lebih banyak jumlah pelanggan informasi suara yang terus mengalami pertumbuhan dengan lebar pita yang terbatas. Hal ini disebabkan karena tingginya biaya untuk pembangunan media transmisi kurang dapat mengimbangi pertumbuhan teknologi komunikasi digital yang menuntut penggunaan lebar pita frekuensi yang sangat besar dalam saluran transmisi namun tetap memiliki kualitas yang baik, terutama dalam pentransmisi informasi suara yang merupakan informasi telekomunikasi yang terbesar dalam penggunaannya [1],[2],[3].

Karena itu diperlukan suatu proses untuk mengefesienkan media transmisi yang dipakai untuk mentransmisikan informasi suara. Disamping dapat mengurangi biaya operasi juga dapat menyediakan lebih banyak kanal transmisi dengan digunakannya lebar pita frekuensi yang kecil namun tetap memiliki kualitas yang baik [3].

Alternatif dari penyelesaian masalah tersebut adalah dengan menerapkan teknik pengkodean suara (*speech coding*) yang merupakan salah satu bagian dari blok enkoder pada sistem telekomunikasi nirkabel. Sistem pengkodean suara dirancang untuk mendukung aplikasi telekomunikasi dengan frekuensi terbatas antara 300 – 3400 Hz yang merupakan frekuensi suara manusia normal. Berdasarkan teorema Nyquist, frekuensi sampling minimal harus dua kalinya lebar pita sinyal yang kontinu terhadap waktu untuk menghindari *aliasing*. Nilai 8 KHz biasanya dipilih sebagai standar frekuensi sampling pengkodean suara [1]. Sedangkan pengkonversian sampel analog ke dalam format digital menggunakan kuantisasi seragam (*uniform quantization*) dengan sampling dari 8 – 16 bits/sampel. Sehingga secara normal, data informasi suara yang ditransmisikan memiliki laju bit 64 - 128 kbps [1],[4],[5]. Laju bit 64 – 128 kbps inilah yang akan kita kompresikan dengan teknik pengkodean suara sehingga didapatkan laju bit serendah mungkin tapi tetap memiliki kualitas suara yang baik.

Sistem - sistem pengkodean suara yang ada antara lain *long term-predictive* 13 kbit/s yang digunakan pada *Public Land Mobile Radio (PLMR)* digital, *Reguler Pulse Excited (LTP-RPE)* 13 kbit/s pada pengkodean GSM, *Vector Sum Excited Linear Predictive (VSELP)* 7,95 kbit/s pada pengkodean *Digital American Mobile Public System (DAMPS)*, dan VSELP 6,7 kbit/s pada pengkodean sistem *Japanese Digital Communication (JDC)*. Selain yang telah disebutkan di atas, teknologi *audio data compression* yang terbaru saat ini ialah *Adaptive Multi Rate (AMR)* yang telah diadaptasikan sebagai standar pengkodean suara oleh 3GPP pada Oktober 1998 dan sekarang banyak digunakan pada GSM dan UMTS. AMR tergolong dalam pengkodean suara hybrid yang menggunakan suatu link yang dapat beradaptasi dengan memilih satu dari delapan laju bit yang bervariasi dari 1,2 s/d 12,2 kbit/s pada teknik pengkodean suara ACELP (*Algebraic Code Excited Linear Program*) yang sesuai dengan kondisi pengkodean suara yang dibutuhkan [9]. ACELP sendiri juga merupakan salah satu pengembangan dari teknik pengkodean CELP yang telah mengalami perkembangan.

VSELP (*Vector Sum Excited Linear Prediction*) telah dipilih menjadi standar untuk digunakan dalam sistem komunikasi digital *North America* [1] sebagai salah satu jenis teknik pengkodean suara. Teknik pengkodean suara VSELP ini dikembangkan oleh TIA (*Telecommunication Industries Association*) USA yang merupakan salah satu metode pengkodean suara paling efisien [1] untuk memungkinkan terjadinya penurunan laju bit (*bit rate*) informasi suara yang cukup besar dengan mutu atau kualitas suara yang dihasilkan cukup baik. Saat ini, teknik pengkodean suara ini telah diterapkan sebagai standar pengkodean suara pada beberapa sistem telepon seluler digital yaitu: IS-54, Japan Digital Cellular dan Half-rate GSM. Metode ini juga dipakai untuk *voice modem* dimana data dan suara dapat ditransmisikan secara bersama – sama melalui jaringan PSTN (*Public Switched Telephone Network*) [1].

Teknik pengkodean suara VSELP ini memungkinkan terjadinya penurunan laju bit terhadap suara yang akan ditransmisikan karena teknik ini bekerja dengan memanfaatkan kombinasi tabel kode (*codebook*) untuk mendapatkan sinyal sintesis. Kombinasi nilai tabel kode (*codebook*) yang

ditransmisikan ini merupakan hasil pembobotan error terkecil terhadap perbandingannya dengan sinyal suara yang akan ditransmisikan. Parameter – parameter penghasil sinyal sintesis inilah yang ditransmisikan sebagai pengganti dari suara asli. Proses untuk mendapatkan sinyal sintesis ini terdapat pada bagian enkoder VSELP, yang juga menghasilkan 27 nilai parameter dari sinyal sintesis yang dihasilkan [1],[2],[4],[5],[8].

Setelah ditransmisikan, parameter – parameter ini diekstraksi pada dekoder sehingga dihasilkan sinyal sintesis. Namun, sinyal sintesis yang dihasilkan perlu ditingkatkan kualitas suaranya. Hal ini diperlukan karena sinyal yang dienkodkan dengan pengkodean suara laju bit rendah menghasilkan sinyal suara dengan nilai SNR yang rendah, tingkat *noise* yang tinggi mendekati *white noise* sehingga sangat sulit untuk menjaganya dibawah nilai ambang batas kemampuan suara yang dapat didengar. Untuk itu, pada sinyal sintesis dilakukan *pitch postfiltering* dan *postfiltering* secara spektral [9].

Hal ini menjadikan dekoder salah satu bagian yang penting dalam pengkodean suara, karena berfungsi untuk meningkatkan kualitas sinyal yang ditransmisikan sehingga menjadi dapat didengar (*audible*), dan informasi suara yang dikirim dapat diterima oleh pendengar [9]. Teknik pengkodean suara VSELP yang telah digunakan saat ini menggunakan bahasa pemrograman C [5],[13] dalam pengaplikasiannya. Namun dalam skripsi ini, penulis menggunakan bahasa pemrograman Simulink MATLAB karena dapat diaplikasikan secara langsung ke dalam prosesor DSP TMS320C5x sebagai salah satu prosesor pengolahan sinyal terbaru saat ini.

Karena alasan tersebut, penulis tertarik untuk mensimulasikan proses pendekodean suara laju bit rendah dengan algoritma VSELP dengan bahasa pemrograman Simulink, yang memiliki tingkat kompresi minimum, dengan tujuan sebagai pemahaman dalam masalah perkonstruksian sinyal untuk mendapatkan hasil sintesis suara dengan kualitas yang baik.

1.2. Tujuan Penulisan

Tujuan dari skripsi ini adalah perancangan simulasi proses komputasi pada bagian dekoder dari teknik pengkodean suara VSELP dengan menggunakan

Simulink, serta menganalisa proses *postfiltering* dekoder untuk mendapatkan sinyal sintesis.

1.3. Batasan Masalah

Pada skripsi ini, permasalahan dibatasi pada dasar teknik pengkodean VSELP serta perancangan dekoder VSELP tanpa memperhatikan kualitas masukan sinyal dari enkoder.

1.4. Metodologi Penelitian

Metodologi penelitian yang saya gunakan yaitu:

1. studi literatur melalui buku, jurnal, dan tesis yang membahas mengenai *speech coding* terutama VSELP (*Vector Sum Excited Linear Prediction*) dan *adaptive filtering*
2. melakukan penelitian perancangan dekoder dari algoritma VSELP 8kbps
3. membuat simulasi untuk algoritma dekoder VSELP yang diimplementasikan pada SIMULINK MATLAB.

1.5. Sistematika Penulisan

BAB I

PENDAHULUAN

Menjelaskan tentang latar belakang dan dasar pemikiran pengkodean suara dengan laju bit rendah, tujuan penulisan, batasan masalah dalam pembahasan tugas skripsi ini, beserta metodologi penelitian yang digunakan dan sistematika penulisan makalah seminar ini.

BAB II

PRINSIP DASAR TEKNIK PENGKODEAN SUARA VSELP

Menjelaskan tentang dasar teori dan teknik – teknik dasar dalam pengkodean suara umumnya, proses pembentukan suara manusia dan dasar teori dari teknik pengkodean suara VSELP

BAB III

ALGORITMA PERANCANGAN RANGKAIAN DEKODER VSELP

Menjelaskan tentang algoritma dasar dalam pembentukan dekoder VSELP, beserta komputasinya.

BAB IV

SIMULASI DAN ANALISA PROSES DEKODER VSELP

Menjelaskan hasil simulasi, serta menganalisa hasil simulasi dalam proses perbaikan sinyal pada dekoder VSELP dengan *post filtering*.

BAB V

KESIMPULAN

BAB II

PRINSIP DASAR TEKNIK PENGKODEAN SUARA VSELP

2.1. Pengkodean Suara

2.1.1. Dasar - dasar pengkodean suara

Secara umum, pengkodean suara merupakan sebuah prosedur untuk merepresentasikan sinyal suara yang terdigitalisasi dengan nilai bit sekecil mungkin, namun tingkat kualitas suara yang dihasilkan tetap terjaga pada waktu yang sama. Pengkodean suara dikenal juga dengan istilah kompresi suara (*speech compression*).

Pada diagram sistem pengkodean suara, pada bagian awal terdapat sinyal suara analog kontinyu terhadap waktu dari suatu sumber yang diberikan didigitalisasi dengan suatu standar hubungan yaitu filter (untuk mengurangi *aliasing*), *sampler* (untuk konversi diskrit terhadap waktu), dan konverter dari analog ke digital (diasumsikan kuantisasi seragam). Keluaran dari sistem ini yaitu sinyal suara diskrit terhadap waktu yang nilai sampelnya juga didiskritkan. Sinyal ini dirujuk sebagai suara digital (*digital speech*).

Seluruh pengkodean suara didisain untuk mengurangi laju bit 128 kbps menjadi nilai yang lebih rendah. Namun terdapat variasi dari nilai laju bit ini, yang dapat kita lihat bahwa nilai laju bit yang berbeda – beda dipicu karena adanya bermacam – macam jenis pengkodean suara. Beberapa metode yang diberikan dapat bekerja dengan baik pada kisaran laju bit tertentu, tapi kualitas dari suara yang didekodekan akan jatuh secara drastis jika besar laju bitnya turun hingga mencapai di bawah nilai *threshold* tertentu. Laju bit minimum yang dapat dicapai suatu pengkodean suara dibatasi oleh konten informasi dari sinyal suara [1].

Selain dari daerah kerja laju bitnya, pengkodean suara juga dapat diklasifikasikan berdasarkan teknik pemrosesannya, yaitu:

- a. Pengkodean bentuk gelombang (*waveform coding*)

Pengkodean ini dilakukan dengan memanfaatkan fakta yang ada bahwa sinyal suara berubah relative lambat terhadap waktu, sehingga sampel

yang diambil tidak jauh berbeda dengan sampel sebelumnya. Pengkodean ini lebih cocok untuk pengkodean laju bit tinggi (antara 24 – 64 kbit/s), karena performanya akan turun dengan tajam dengan menurunnya laju bit. SNR dapat diutilisasi untuk mengukur kualitas dari pengkodean bentuk gelombang. Beberapa contoh pengkodean bentuk gelombang yaitu berbagai macam jenis *pulse code modulation* (PCM) dan ADPCM.

b. Vocoding atau pengkodean parametrik (*parametric coder*)

Pada pengkodean ini, sinyal suara dianggap hasil dari sebuah model, yang dikontrol oleh parameter – parameter. Selama proses *encoding*, parameter dari model diestimasi dari masukan sinyal suara, dimana parameter – parameter ini ditransmisikan dengan pengkodean arus bit (*bit-stream*). Dengan menggunakan pengkodean tipe ini maka tidak ada upaya yang dapat dilakukan untuk menjaga bentuk asli dari bentuk gelombang dan pengukuran kualitas dengan SNR percuma untuk dilakukan. Kualitas perceptual dari sinyal yang didekodekan terhubung secara langsung dengan keakuratan dan kecanggihan dari model yang bersangkutan. Karena keterbatasan ini, pengkodean ini bersifat spesifik terhadap sinyal dan memiliki kinerja yang buruk untuk sinyal selain sinyal suara.

Prediksi linier (*linear prediction*) merupakan pendekatan pemodelan yang paling sesuai untuk pengkodean parametrik. Mekanisme produksi sinyal suara manusia diringkas menggunakan *time-varying filter*, dengan koefisien filternya didapatkan dengan menggunakan prosedur analisis prediksi linier.

Pengkodean jenis ini dapat bekerja dengan baik untuk laju bit rendah (antara 2 – 5 kbps). Secara normal, peningkatan laju bit tidak ditafsirkan menjadi peningkatan kualitas yang lebih baik, karena hal ini terbatas dengan model yang dipilih.

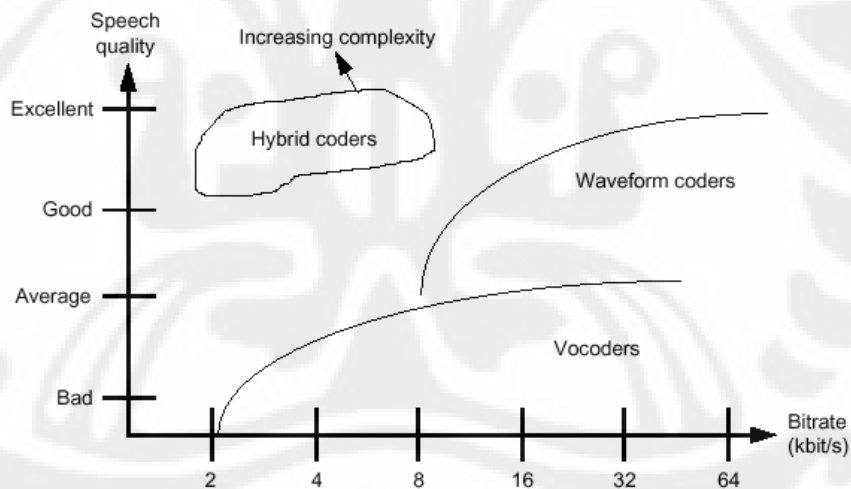
c. Hybrid Coding (analisis-sintesis)

Pengkodean ini dilakukan dengan cara menggabungkan teknik *waveform coding* dan *parametric coding*. Seperti pada pengkodean parametrik, pengkodean hybrid tergantung pada produksi model suara; selama *encoding*, parameter dari model dipindahkan. Parameter tambahan dari

model dioptimasi dengan cara tertentu sehingga suara yang telah didekodekan mendekati bentuk gelombang aslinya, dimana pendekatannya diukur menggunakan pembobotan error kesalahan sinyal. Seperti pada pengkodean bentuk gelombang (*waveform coding*), upaya untuk menyocokkan sinyal asli dengan sinyal yang didekodekan pada domain waktu juga dibuat.

Pengkodean jenis ini mendominasi pada pengkodean dengan laju bit menengah, dimana algoritma CELP dan variasinya merupakan salah satu contohnya. Pengkodean hybrid bertindak seperti *waveform coder* untuk laju bit tinggi dan seperti *parametric coder* untuk laju bit rendah, dengan kualitas yang cukup baik untuk laju bit menengah [1],[2],[3].

Perbandingan kualitas suara dan laju bit dari berbagai jenis pengkodean suara dapat dilihat pada Gambar 2.1.

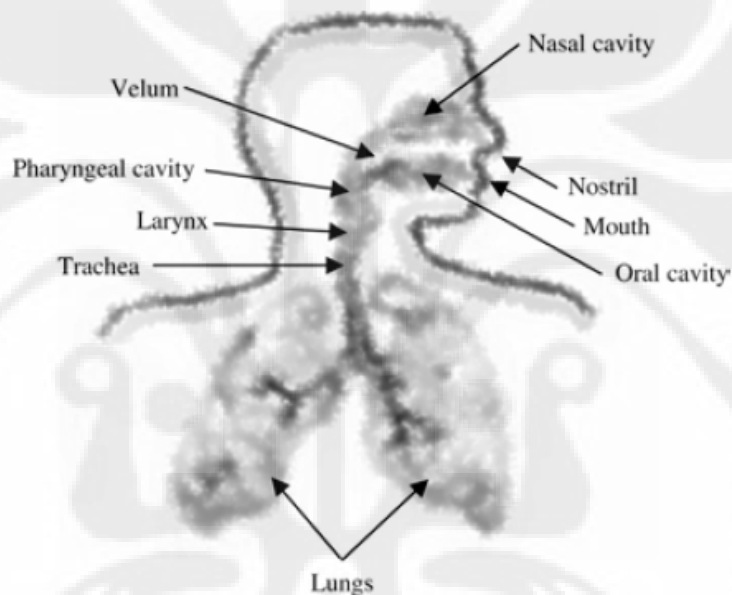


Gambar 2.1. Perbandingan kualitas suara dan laju bit berbagai jenis pengkodean suara [1]

2.1.2. Konsep dasar produksi suara manusia

Bentuk gelombang suara adalah sebuah tekanan gelombang suara yang berasal dari pergerakan struktur anatomi yang dibuat sistem produksi suara manusia (Gambar 2.2.). Suara (*speech*) secara dasar dihasilkan sebagai gelombang akustik yang dipancarkan dari lubang hidung (*nasal*) dan mulut ketika udara

dikeluarkan dari paru – paru dengan hasil aliran udara mendesak karena adanya desakan di dalam tubuh. Hal ini berguna untuk menginterpretasikan produksi suara dalam hal penyaringan akustik. Tiga rongga utama dari sistem produksi suara yaitu *nasal*, *oral* dan *pharyngeal* yang membentuk filter akustik utama. Filter dieksitasi oleh udara dari paru – paru dan diisi pada keluaran utama filter tersebut dengan tahanan yang berkaitan dengan bibir.



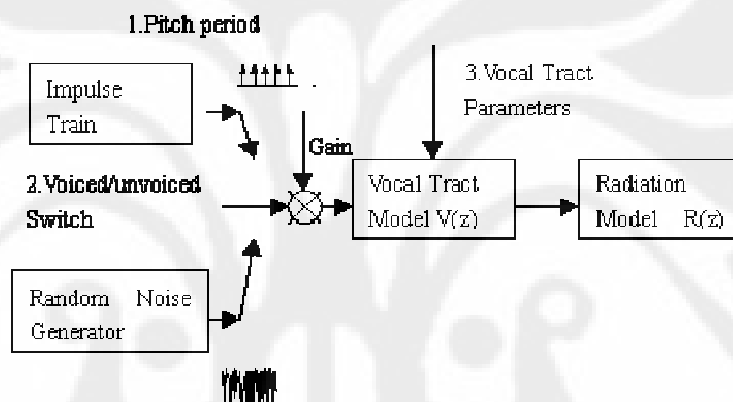
Gambar 2.2. Diagram sistem produksi suara manusia [1]

Rongga suara (*vocal tract*) merujuk pada *pharyngeal* dan rongga mulut (*oral cavity*) yang dikelompokkan bersama – sama. *Nasal tract* dimulai dari bagian *velum* dan berakhir pada lubang hidung (*nostrils*). Ketika *velum* merendah, *nasal tract* berpasangan secara akustik dengan rongga suara untuk menghasilkan suara nasal.

Wujud dan bentuk dari *vocal* dan *nasal tract* berubah kontinyu terhadap waktu, menciptakan sebuah filter akustik dengan respon frekuensi terhadap variasi waktu. Saat udara dari paru – paru berjalan melalui *tracts*, spektrum frekuensi dibentuk oleh selektivitas frekuensi dari *tract – tract* ini. Frekuensi resonansi dari

pipa rongga suara disebut frekuensi formant (*formant frequency*) atau formant, yang tergantung pada bentuk dan dimensi dari rongga suara.

Di dalam laring (*larynx*), terdapat salah satu komponen paling penting dari sistem produksi suara yaitu pita suara (*vocal cords*). Pita suara adalah sepasang pita elastis otot dan membran muskos yang membuka dan menutup dengan cepat selama produksi suara. Kecepatan pita dalam membuka dan menutup unik untuk masing – masing individu dengan suara tertentu.



Gambar 2.3. Model produksi suara [1]

Saat berbicara, sinyal suara dapat diklasifikasikan menjadi ucapan bersuara (*voiced*) dan ucapan tidak bersuara (*unvoiced*). Ucapan bersuara dihasilkan ketika pita suara bergetar sehingga aliran udara dari paru – paru terinterupsi secara periodik, menciptakan deretan pulsa untuk mengeksitasi *vocal tract*. Dengan pita suara yang tetap (tidak bergetar), terjadi turbulensi yang diciptakan dengan adanya aliran udara yang melewati rongga suara yang terdesak sehingga menghasilkan ucapan tak bersuara. Pada domain waktu, ucapan bersuara dikarakteristikan dengan periodesitas yang kuat yang muncul di dalam sinyal, dengan frekuensi dasar yang disebut sebagai frekuensi puncak (*pitch frekuensi*) atau *pitch*. Untuk laki – laki, kisaran *pitch* berada pada 50 – 250 Hz, sedangkan untuk wanita kisarannya pada interval 120 – 500 Hz. Ucapan tak bersuara,

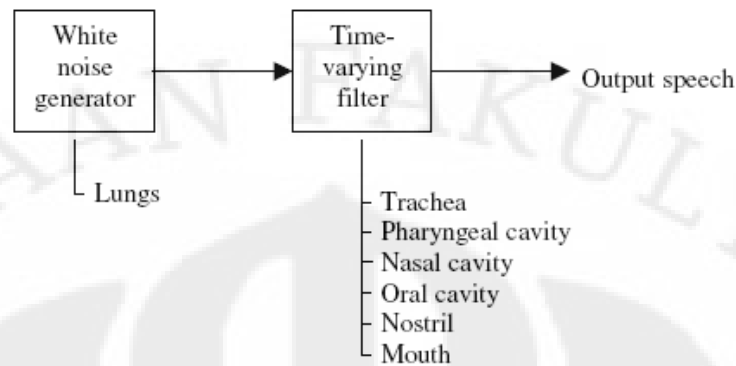
sebaliknya tidak menunjukkan adanya periodesitas dan pada dasarnya acak secara alami.

Perlu diketahui bahwa untuk mengklasifikasikan antara ucapan bersuara/tidak bersuara tidak dapat terlalu jelas dibedakan untuk seluruh bingkai, karena selama transisi (dari ucapan bersuara ke ucapan tak bersuara atau sebaliknya) akan timbul ketidakaturan dan quasiperiodesitas yang sulit untuk ditentukan dengan tegas apakah itu ucapan bersuara atau ucapan tidak bersuara.

Pada sebagian besar pengkodean suara, sinyal diproses berbasis bingkai ke bingkai (*frame-by-frame*) dimana sebuah bingkai terdiri dari sejumlah sampel yang terbatas. Panjang dari bingkai dipilih dengan cara tertentu dimana statistik sinyal tetap konstan selama interval. Panjangnya biasanya antara 20 – 30 ms, atau 160 – 180 sampel untuk sampling 8 kHz [1].

2.1.3. Pemodelan sistem produksi suara

Sistem produksi suara manusia dapat dimodelkan menggunakan struktur yang sederhana; paru – paru (menghasilkan udara atau energi untuk mengeksitasi *vocal tract*) diwakilkan dengan sumber *white noise*. Jalur akustik di dalam tubuh dengan seluruh komponennya berkaitan dengan filter yang bervariasi waktu (*time-varying filter*). Konsep ini dilustrasikan pada Gambar 2.4. Model sederhana ini merupakan inti struktur dari berbagai macam algoritma pengkodean suara. Dengan menggunakan teknik identifikasi sistem disebut dengan prediksi linier (*linear prediction*), memungkinkan untuk memperkirakan parameter *time-varying filter* dari sinyal yang diobservasi.



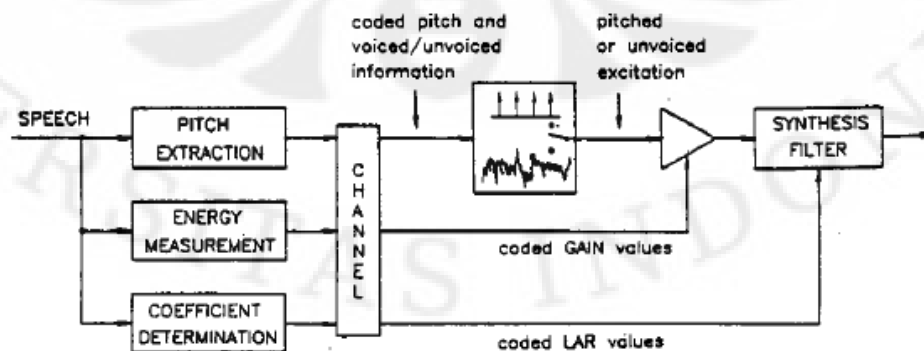
Gambar 2.4. Koresponden antara sistem produksi suara manusia dengan sistem yang telah disederhanakan berbasis *time-varying filter* [1]

Asumsi dari model ialah bahwa distribusi energi dari sinyal suara pada domain frekuensi seluruhnya karena *time-varying filter*, dengan paru – paru yang memproduksi sinyal eksitasi memiliki *white noise* dengan spektrum yang rata. Model ini cukup efisien dan banyak peralatan analitik yang telah dikembangkan si sekitar konsep. Ide ini cukup dikenal sebagai *autoregressive model* [1].

2.2. Pengkodean Suara Analisis Sintesis

Seperti yang telah dijelaskan sebelumnya bahwa VSELP merupakan salah satu pengkodean suara analisis-sintesis atau *hybrid coding*. Proses analisis – sintesis ini terbagi menjadi tiga komponen yaitu:

- Filter jangka pendek (*Short Term Filter*)
- Filter jangka panjang (*Long Term Filter*)
- Eksitasi



Gambar 2.5. Pengkodean analisis-sintesis menggunakan LPC [1]

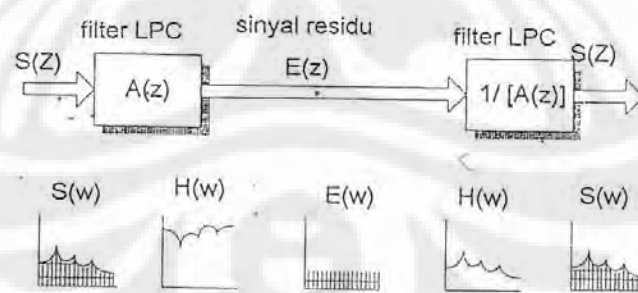
2.2.1. Filter Jangka pendek (*Short Term Filter*)

Filter jangka pendek (*short term filter*) dalam hal ini merupakan filter sintesis *Linear Predictive Coding* (LPC) yang menghasilkan keluaran suara sintesis. Fungsi transfer untuk filter jangka pendek merupakan filter all-pole, hal ini dikarenakan *vocal tract* dimodelkan oleh LPC dengan sebuah all-pole filter yang memprediksi suara dengan kombinasi dari input dan keluaran sebelumnya secara rekursif yaitu:

$$A(z) = \frac{1}{1 - \sum_{i=1}^{N_p} \alpha_i z^{-i}} \quad (2.1)$$

Parameter prediksi jangka pendek adalah nilai α_i pada filter jangka pendek atau filter sintesis. Nilai ini merupakan koefisien standar filter *direct form*. Parameter prediksi jangka pendek dikomputasi dari sinyal suara masukan. Tingkat prediksinya adalah 10 ($N_p=10$). LPC 10 adalah standar bagi algoritma kompresi yang lain untuk mempergunakan sepuluh koefisien untuk analisa jangka pendek.

Apabila diberikan sinyal digital $S(z)$ dan dilewatkan pada filter LPC dengan fungsi alih $A(z)$, maka dengan mengacu pada Gambar 2.5. akan didapatkan sinyal keluaran $E(z)$. $E(z)$ didefinisikan sebagai sinyal sisa (residu) dari sinyal $S(z)$ yang dilewatkan pada filter LPC dengan fungsi alih, $H(z) = E(z)/S(z) = A(z)$.



Gambar 2.6. Penerapan filter LPC pada sinyal ucapan [2]

$$E(z) = A(z)S(z) \quad (2.2)$$

$$E(z) = \left[\frac{1}{1 - \sum_{i=1}^{N_p} \alpha_i z^{-i}} \right] S(z) \quad (2.3)$$

Model filter LPC pada Gambar 2.6. haruslah memiliki sinyal sampel spektral yang dapat mewakili fungsi kebalikan dari fungsi spektral dari sinyal $S(z)$, agar didapatkan bentuk spektral $E(z)$ yang rata untuk setiap komponen frekuensinya.

Dengan menghasilkan nilai fungsi spektral $E(z)$ yang sama untuk setiap saat maka apabila $E(z)$ dilewatkan pada filter LPC dengan fungsi alih, $H(z) = S(z)/E(z) = 1/A(z)$, maka akan didapatkan $S(z)$ yang memiliki kualitas sinyal sintesis yang sama dengan kualitas sinyal aslinya. Konsep inilah yang menjadi acuan dasar bagi pengkodean suara dengan menggunakan LPC sebagai filter yang mewakili rongga ucapan.

Untuk menentukan koefisien filter jangka pendek, digunakan algoritma FLAT (*Fixed Point Covariance Lattice Algorithm*). Sampel sinyal masukan berada pada interval analisis $0 \leq n \leq N_A - 1$. Panjang analisis yang digunakan untuk komputasi parameter adalah 170 sampel ($N_A = 170$). Sehingga didapatkan 10 koefisien refleksi ($r_1 - r_{10}$) yang merepresentasikan parameter prediksi jangka pendek yang dikuantisasi secara terpisah dengan menggunakan 10 tabel kode kuantisasi koefisien refleksi (dokumen TIA) dan didapatkan koefisien refleksi terkuantisasi (LPC1 – LPC10). Indeks pada tabel kode dari koefisien refleksi yang hasil kuantisasi ini merupakan salah satu parameter yang dikirimkan ke dekoder. Sedangkan nilai koefisien refleksi yang telah dikuantisasi kemudian dikonversi menjadi koefisien filter *direct form*, α_i dan dinterpolasi untuk mengecek kestabilan filter.

Pada bagian dekoder, indeks dari koefisien refleksi yang telah dikuantisasi ini, dikonversi kembali untuk mendapatkan nilai koefisien filter *direct form*, α_i . Hal ini dicapai dengan menggunakan indeks dari tiap koefisien refleksi yang digunakan untuk mendapatkan nilai kode – kode LPC1 – LPC10, kemudian didekodekan menjadi 10 koefisien refleksi ($r_1 - r_{10}$) kembali dengan menggunakan tabel data koefisien refleksi pada dokumen TIA. Kemudian dikonversikan ke dalam α_i dengan menggunakan filter *direct form*.

2.2.2. Filter Jangka Panjang (Long Term Filter)

Salah satu parameter terpenting di dalam aplikasi analisis, sintesis dan pengkodean suara adalah frekuensi fundamental, atau pitch atau bunyi suara. Frekuensi pitch secara langsung berhubungan dengan Fundamental frekuensi berkenaan langsung dengan pembicara dan memberikan warna suara bagi pembicara. Suara tercipta ketika udara mengalir dari paru-paru diinterupsi oleh getaran pita suara secara periodik. Periode dimana pita suara terbuka adalah frekuensi fundamental manusia atau pitch. *Lag* adalah rentang waktu dimana fundamental frekuensi berulang.

Filter jangka panjang (*long term predictor*) atau juga disebut *pitch prediction lag* merupakan suatu model yang dapat menentukan priodesitas sinyal yang direpresentasikan dalam *lag*. Untuk mendapatkan besarnya *lag* dapat digunakan dua jenis metode, yaitu:

- Metode *open loop*

Long term prediction lag ditentukan secara langsung dari sinyal masukan atau sinyal masukan atau sinyal residu (*pitch extraction*), metode ini biasa digunakan dalam *vocoder LPC*.

- Metode *close loop*

Long term prediction lag ditentukan dengan menggunakan sebuah tabel kode adaptif sebagai sumber *lag*. Metode ini biasa digunakan dalam pengkodean analisis-sintesis.

Pria memiliki fundamental frekuensi yang berada pada rentang 50 sampai 250 Hz sedangkan wanita berada pada rentang 120 dan 500 Hz. Pada domain waktu, pria berada pada rentang 4 sampai 20 ms sedangkan wanita berada pada rentang waktu 2 sampai 8 ms.

Dalam prakteknya, sebagian besar *lag* mempunyai jarak antara 2,5 ms (frekuensi dasar 400 Hz) sampai 18,25 ms (frekuensi dasar 55 Hz). Daerah kerja *lag* dapat dikonversikan ke dalam nilai 20 sampai 146 sampel, sehingga terdiri dari 127 macam nilai *lag*

Pada semua teknik kompresi suara, nilai *lag* akan memberikan warna pada suara sintesis pembicara. *Lag* merupakan bagian yang penting karena merupakan letak dari fundamental frekuensi suara yang akan memberikan hasil keluaran suara sintesis yang sesuai dengan suara yang diestimasi.[1],[9],[11].

2.2.3. Eksitasi

Di dalam usaha untuk menghasilkan suara sintesis yang mendekati suara aslinya, maka pada pengkodean analisis-sintesis pengolahan yang paling utama dilakukan di bagian eksitasi. Reproduksi eksitasi atau sinyal residu dapat dilakukan dengan beberapa metode seperti, MPLPC (*Multiple LPC*), RPE-LPC (*Regular Pulse Excited LPC*), SEV (*Self Excited LPC*), dan tabel eksitasi (*codebook excitation*). VSELP menggunakan tabel eksitasi, yang mana tabel itu berisi sejumlah sampel sari eksitasi aperiodik.

Sinyal residu ini dianalisa dengan cara mencocokkan data suara persubbingkai dengan kombinasi tabel kode. Nilai estimasi yang memiliki tingkat kecocokan tertinggi ini akan dimasukkan kembali ke dalam LPC sehingga akan memberikan sinyal suara sintesis yang mendekati sinyal aslinya.

2.3. Dasar Enkoder dan Dekoder Pada VSELP

2.3.1. Dasar enkoder pada VSELP

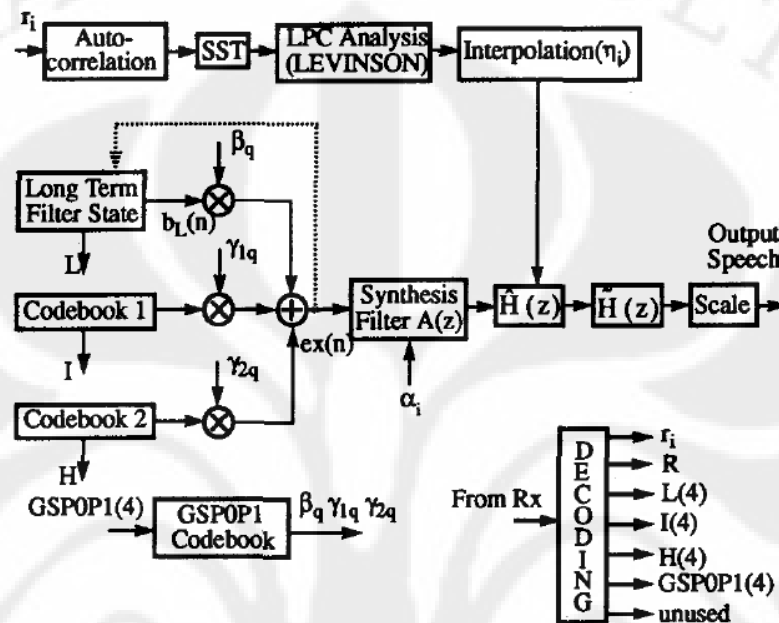
Gambar 2.7. merupakan sisi enkoder dari proses pengkodean suara VSELP, dimana pada sisi enkoder ini menggunakan dua tabel kode eksitasi, satu tabel kode adaptif untuk lag, dan satu tabel kode penguatan yang mengandung tiga nilai penguatan yang dioptimalkan bersama – sama untuk tabel kode eksitasi dan adaptif. Sinyal suara analog yang masuk dikonversi ke bentuk sinyal digital via *onchip analog-to-digital converter* (ADC) dengan *sampling rate* 8 KHz. Kemudian sinyal digital difilter dengan filter *highpass* Chebyshev orde keempat tipe II. Matriks kovarian diperoleh dari sinyal suara terfilter $s(n)$. Analisis LPC menggunakan *fixed point covariance lattice algorithm* (FLAT) digunakan untuk memperoleh koefisien refleksi. Sebelum analisis LPC, *spectral smoothing technique* (SST) menggunakan *binomial window* diaplikasikan kepada matriks kovarian. Energi bingkai dihitung dari matriks kovarian sebelum SST.

Prosedur algoritma FLAT menghasilkan koefisien refleksi yang dikonversi ke dalam koefisien prediksi α_i yang ditunjukkan di dalam blok interpolasi. Koefisien prediksi diinterpolasi untuk subbingkai pertama, kedua dan ketiga untuk setiap frame sebelum mereka digunakan di dalam filter sintesis pembobotan LPC ,

$H(z)$. Filter sintesis pembobotan dieksitasi oleh setiap vektor kode dari setiap tabel kode dan sinyal sintesis, $p'(n)$ dibandingkan dengan sinyal pembobotan, $p(n)$ yang diperoleh dengan melewati $s(n)$ melalui filter pembobotan, $W(z)$. Vektor kode yang memproduksi total error pembobotan minimum adalah vektor kode yang dipilih.

VSELP menggunakan dua tabel kode eksitasi, masing - masing mengandung M vektor basis ($M=7$ pada standar Amerika Utara). Dua vektor kode untuk tiap tabel kode dikonstruksi dari M vektor basis. VSELP menggunakan satu tabel kode adaptif untuk prediksi lag jangka panjang. Sebagai tambahan, VSELP menggunakan satu tabel kode penguatan yang mana tiga nilai penguatan dioptimasi bersamaan untuk prediksi lag jangka panjang dan dua vektor kode eksitasi dihasilkan di dalam bentuk vektor tunggal. Prosedur pencarian kode yang menyeluruh ini diaplikasikan untuk menemukan nilai prediksi jangka panjang yang optimal, dua vektor kode eksitasi dari tabel kode eksitasi, dan nilai penguatan dari tabel kode penguatan. Sebelum perbandingan antara sinyal pembobotan dan sinyal eksitasi, *zero input response*, $H(z)$, dikurangi dari sinyal pembobotan untuk menjaga kelembutan antara subbingkai selanjutnya. VSELP menggunakan teknik analisis-sintesis (lup tertutup), dengan kata lain, enkoder VSELP menggunakan sinyal sintesis untuk setiap vektor kode untuk mencari vektor kode paling optimal. Karena itu, dekoder VSELP dapat diinterpretasikan sebagai subset dari enkoder VSELP.

penguatan melewati *postfilter* spektral adaptif. Sinyal sintesis dikirim pada keluaran dari tahap penskalaan (*scaling stage*).



Gambar 2.8. Blok diagram dekoder VSELP [9]

2.4. Struktur Tabel Kode Eksitasi dan Adaptif VSELP

2.4.1. Tabel kode eksitasi

Tabel kode eksitasi pada pengkodean VSELP menggunakan dua tabel kode eksitasi yang masing – masing terdiri dari M vektor basis, tiap vektor basis terdiri dari N elemen (sample) yang didapat dari hasil pengkuantisasian. Pengaruh ini dapat dilihat dalam pencarian vektor kode pada tabel kode (1 tabel kode = 2^M vektor kode), dimana pembentukannya merupakan kombinasi linier dari M vektor basis. VSELP menggunakan vektor basis ($M=7$) per tabel kode, sehingga untuk tiap tabel kode terdiri dari 128 vektor kode. Dari pernyataan di atas, maka untuk vektor basis $v_{k,m}(n)$, besar vektor kode dapat diekspresikan ke dalam persamaan 2.4.,

$$u_{k,t}(n) = \sum_{m=1}^M \theta_{tm} \cdot v_{k,m}(n) \quad (2.4)$$

dimana $0 \leq i \leq 2^M - 1$ dan $0 \leq n \leq N - 1$ ($N=40$), sedangkan $k=1$ untuk tabel kode pertama dan $k=2$ untuk tabel kode kedua, sedangkan θ_{im} adalah parameter yang menentukan kombinasi linier pembentukan vektor kode dengan ketentuan yaitu:

$$\theta_{im} = +1 \text{ jika bit } m \text{ dari codeword } i = 1$$

$$\theta_{im} = -1 \text{ jika bit } m \text{ dari codeword } i = 0$$

pembentukan vektor kode I berasal dari penjumlahan M basis vektor yang tanda operasinya (+/-) ditentukan oleh nilai bit dalam indeks i [1],[2],[4],[5],[8],[9],[11]. Misalkan untuk indeks 1 (dalam biner 7 bit: 0000001) mempunyai $\theta_{11} = 1$ dan $\theta_{12} = \theta_{13} = \dots = \theta_{17} = -1$, maka besar vektor kode 1 dalam tabel kode pertama dapat dituliskan ke dalam Persamaan 2.5. :

$$u_{1,1}(n) = \sum_{m=1}^M \theta_{im} \cdot v_{1,m}(n) = v_{11}(n) - v_{12}(n) - \dots - v_{17}(n) \quad (2.5)$$

contoh dari penggunaan persamaan di atas, misalkan untuk tabel kode yang berukuran 2^2 atau tabel kode yang terdiri dari 2 vektor basis v_1 dan v_2 , maka kombinasi liniernya didapatkan:

$$u_i = u_{00} = \theta_{01} \times v_1 + \theta_{02} \times v_2 = v_1 + v_2$$

$$u_i = u_{01} = \theta_{11} \times v_1 + \theta_{12} \times v_2 = -v_1 + v_2$$

$$u_i = u_{10} = \theta_{21} \times v_1 + \theta_{22} \times v_2 = v_1 - v_2$$

$$u_i = u_{11} = \theta_{31} \times v_1 + \theta_{32} \times v_2 = -v_1 + v_2$$

dari contoh di atas dapat dilihat bahwa $u_0 = -u_2$ dan $u_1 = -u_3$, keadaan ini disebut juga vector table kode yang saling komplementen (complementary codebook vector). Hal ini kemudian akan mengurangi komputasi dalam proses pencarian indeks tabel kode yang sesuai [11].

2.4.2. Tabel kode adaptif

VSELP menggunakan sebuah tabel kode adaptif yang didapatkan dari *long term predictor lag* (L). Lag merupakan nilai tertinggi dari suatu sampel suara, dimana nilai lag ini dihasilkan oleh *long term filter state*.

Ada dua cara untuk menentukan *long term predictor lag* (L), yang pertama menggunakan metode lup terbuka dimana lag ditentukan dari masukan sinyal atau langsung dari sinyal sisa (*residual signal*) lag itu sendiri. Cara kedua menggunakan metode lup tertutup dimana lag besarnya ditentukan pada *rate*

subbingkai dari data yang di-enkodekan terhadap data *long term filter* sebelumnya dan masukan sinyal suara saat ini untuk subbingkai. Dengan menggunakan data yang telah diencodekan, penentuan dari nilai *long term predictor lag* berdasarkan keadaan *long term filter* sebenarnya yang juga terdapat pada *synthesizer*. Lag dapat dibuat sehingga nilai pembobotan kesalahannya dapat minimal. Hal ini menunjukkan bahwa metode dengan lup tertutup memberikan kinerja yang lebih baik dalam penentuan nilai lag dibandingkan metode lup terbuka.

Dalam prakteknya, kebanyakan lags berada di kisaran 2,5 msec (dasar frekuensi *pitch* 400 Hz) sampai dengan 18,25 msec (dasar frekuensi *pitch* 55 Hz). Rentang nilai lag dapat dikonversi ke nilai 20-146 dari sampel, yang membuat terdapat 127 tingkat. Dengan konfigurasi lup tertutup, menjadi relatif mudah untuk menemukan lag yang lebih besar daripada atau sama dengan menilai subbingkai ($L \geq N$). Waktu untuk setiap subbingkai adalah 5 msec, sehingga panjang nilai untuk lag harus kurang dari panjang subbingkai.

2.4.3. Komposisi parameter dalam bingkai

Pengkodean yang dilakukan dalam VSELP yaitu dengan cara mereduksi informasi suara sebesar 160 sampel menjadi 27 parameter. Dimana alokasi bit untuk tiap parameternya berbeda – beda. Nilai ke-27 parameter tersebut diterjemahkan ke dalam 160 bit yang alokasi bit masing – masing parameternya dapat dilihat dalam tabel berikut ini atau untuk lebih jelasnya dapat dilihat dalam lampiran. Informasi reduksi tersebut ada yang berlaku untuk tiap bingkai dan tiap subbingkai.

Tabel 2.1. berisi ringkasan dari parameter dasar untuk pengkodean suara VSELP 8 kbps tanpa perbaikan error. Terdapat 160 bits per bingkai (20ms) untuk pengkodean suara 8 kbps, yang mana pengalokasiannya terlihat pada Tabel 2.2.

Tabel 2.1. Parameter dasar pengkodean VSELP [9]

F_s	Sampling rate	8 kHz
N	Panjang subbingkai	40 sampel (5msec)
N_f	Panjang bingkai	160 sampel (20msec)

N_p	Tingkat prediksi jangka pendek	10
N_a	Interval analisis FLAT: $N_f + N_p$	170 sampel
L	# taps untuk orde prediksi jangka panjang	1
M_1	# bits pada <i>codeword</i> 1 (vektor basis)	7
M_2	# bits pada <i>codeword</i> 2 (vektor basis)	7
λ	Parameter filter pembobotan	0,8

Tabel 2.2. Alokasi bit untuk setiap bingkai VSELP [5]

Parameter	Bit/subbingkai (5ms)	Bit/bingkai (20 ms)
10 koefisien LPC(10)		38
Energi bingkai (R(0))		5
Lag L (4)	7	28
Indeks tabel kode I,H(8)	7+7	56
GSP0P1(4)	8	32
Tidak terpakai		1
TOTAL	29	160

Hubungan dari enkoder dan dekoder dilakukan dengan pengiriman parameter – parameter dari Tabel 2.2. di atas.

BAB III

ALGORITMA PERANCANGAN RANGKAIAN DEKODER VSELP

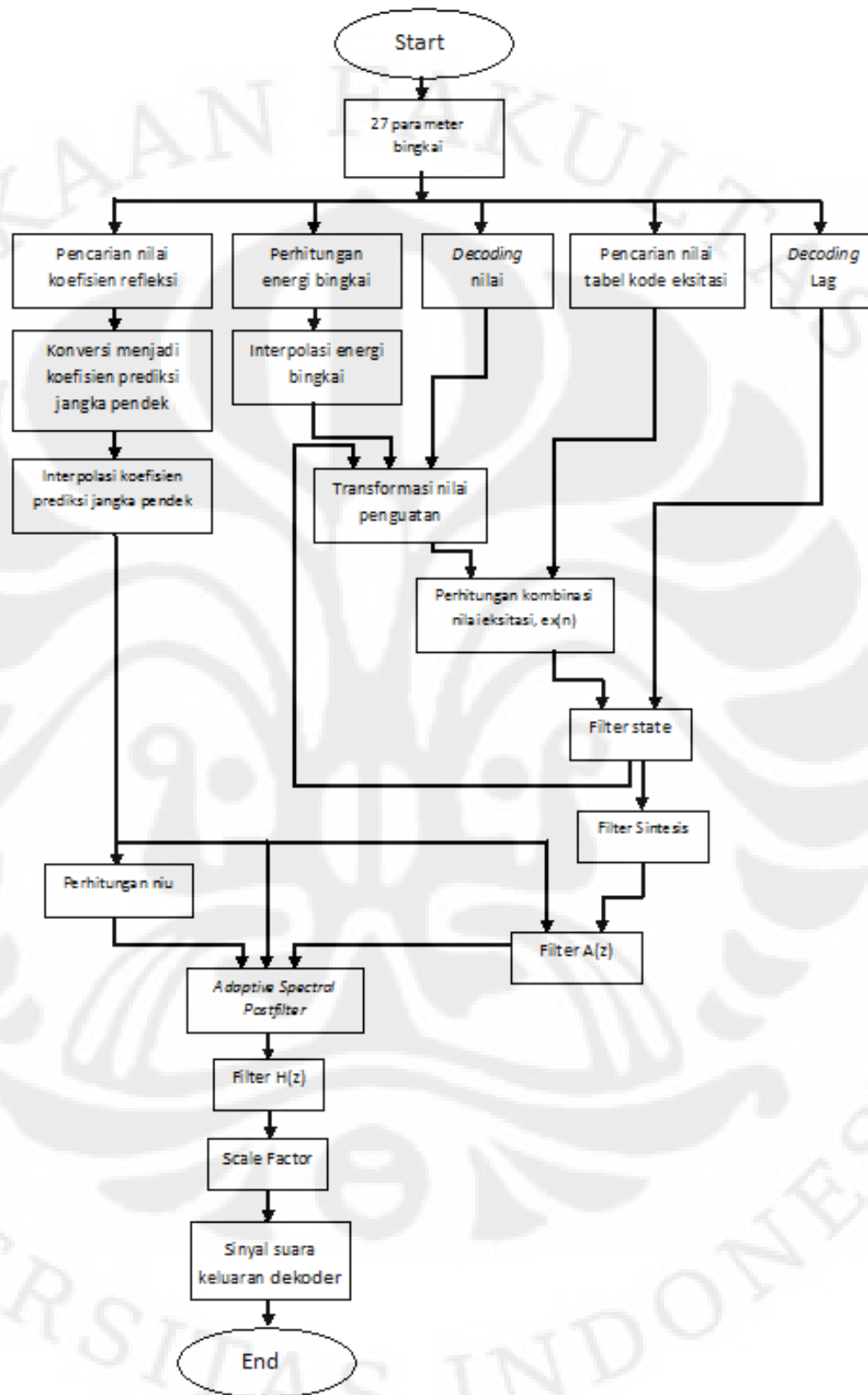
Setelah seluruh parameter subbingkai dan bingkai ditentukan, maka seluruh 27 parameter per bingkai ini (Lampiran 1) dikirimkan dari enkoder ke dekoder. Perancangan pada rangkaian dekoder VSELP diawali dengan pengekstrasian seluruh 27 parameter – parameter ini, yang kemudian diproses sesuai dengan fungsinya masing – masing sesuai dengan algoritma dari simulasi dekoder VSELP yang terlihat pada Gambar 3.1.

Parameter – parameter ini merupakan nilai data indeks dari kuantisasi tabel setiap parameter. Jumlah bit yang dipakai merupakan nilai data indeksnya. Setiap parameter dikuantisasi sesuai dengan banyaknya bit kuantisasi indeks. Untuk N bit berarti ada 2^N data kuantisasi setiap parameter. Setiap data tabel kuantisasi didapatkan dari standar IS54 yang dipakai pada standar telekomunikasi Amerika Utara (Lampiran 1).

3.1. Koefisien Prediksi Jangka Pendek

Koefisien prediksi jangka pendek dikodekan sebagai koefisien refleksi terkuantisasi, dan parameter yang dikirimkan dari enkoder ke dekoder berupa indeks tabel kode. Dengan menggunakan indeks tabel kode ini, didapatkan kode – kode 10 koefisien refleksi ($r_1 - r_{10}$) dari tabel kode koefisien kuantisasi pada Lampiran 2.

Koefisien refleksi kemudian dikonversi menggunakan filter *direct form*, sehingga didapatkan koefisien prediksi, α_i . Pada koefisien prediksi ini kemudian dilakukan proses interpolasi untuk mengecek kestabilan filter LPC.



Gambar 3.1. Algoritma Dekoder VSELP

3.1.1. Interpolasi

Parameter LPC yang didapatkan merupakan hasil perolehan untuk satu bingkai (*frame*), maka apabila proses sintesis yang dilakukan per-subbingkai (*per subframe*,) dimana 1 bingkai terdiri dari 4 subbingkai, maka perlu dilakukan interpolasi untuk memenuhi parameter tiap subbingkainya. Proses interpolasi didahului dengan merubah koefisien refleksi (r_i) yang berasal dari bagian ekstraksi parameter (*short term prediction parameter*) menjadi koefisien prediksi (α_i).

Pengkodean suara menginterpolasi secara linier α_i untuk subbingkai pertama, kedua dan ketiga dari setiap bingkai. Bingkai keempat menggunakan α_i yang belum terinterpolasi untuk bingkai tersebut. Untuk setiap α_i dihitung menggunakan Persamaan 3.1. – 3.4. :

$$\alpha_i = (0,75)\alpha_{i,(sebelum)} + (0,25)\alpha_{i,(saat\ ini)} \quad \text{untuk subbingkai 1} \quad (3.1)$$

$$\alpha_i = (0,5)\alpha_{i,(sebelum)} + (0,5)\alpha_{i,(saat\ ini)} \quad \text{untuk subbingkai 2} \quad (3.2)$$

$$\alpha_i = (0,25)\alpha_{i,(sebelum)} + (0,75)\alpha_{i,(saat\ ini)} \quad \text{untuk subbingkai 3} \quad (3.3)$$

$$\alpha_i = \alpha_{i,(saat\ ini)} \quad \text{untuk subbingkai 4} \quad (3.4)$$

Proses interpolasi juga dilakukan pada parameter lainnya yaitu parameter energi bingkai (R). $R'_q(0)$ didefinisikan sebagai nilai dari bingkai yang telah diproses dan $R_q(0)$ adalah nilai koefisien dari bingkai sebelum diproses, dengan ketentuan Persamaan 3.5. – 3.7. :

$$R'_q(0) = R_q(0)_{\text{bingkai sebelumnya}} \quad \text{untuk subbingkai 1} \quad (3.5)$$

$$R'_q(0) = R_q(0)_{\text{bingkai saat ini}} \quad \text{untuk subbingkai 3 \& 4} \quad (3.6)$$

$$R'_q(0) = \sqrt{R_q(0)_{\text{bingkai sebelumnya}} \cdot R_q(0)_{\text{bingkai saat ini}}} \quad \text{untuk subbingkai 2} \quad (3.7)$$

3.1.2. Stabilitas koefisien refleksi

Setelah proses interpolasi, dilakukan pengetesan stabilitas terlebih dahulu dengan merubah kembali koefisien prediksi (α_i) menjadi koefisien refleksi (r_i), dimana stabilitas filter dikatakan baik bila nilainya $-1 \leq r_i \leq 1$. Jika salah satu koefisien refleksi (r_i) ada yang tidak stabil ($|r_i| > 1$) maka akan dilakukan penyesuaian dengan ketentuan sebagai berikut, jika:

- $r_{i(1)}$ tidak stabil maka akan digunakan koefisien $r_{i, \text{sebelum}}$

- $r_{i(3)}$ tidak stabil maka akan digunakan koefisien $r_{i, \text{saat ini}}$
- $r_{i(2)}$ tidak stabil maka akan digunakan koefisien $r_{i, \text{sebelum}}$ atau koefisien $r_{i, \text{saat ini}}$ tergantung mana yang lebih besar energi bingkainya, jika sama maka $r_{i(2)}$ akan menggunakan koefisien $r_{i, \text{sebelum}}$

Hasil dari pengetesan stabilitas koefisien dari tiap bingkai kemudian dirubah kembali menjadi koefisien prediksi (α_i) yang kemudian dikirim ke filter sintesis LPC.

3.2. Energi Bingkai

Energi bingkai dihitung dan dienkodekan untuk setiap bingkai. Nilai energi, $R(0)$, merefleksikan energi rata – rata sinyal di dalam interval 20 ms setiap masukan suara yang terletak di tengah – tengah dari keempat subbingkai.

Parameter energi bingkai yang dikirimkan dari enkoder menuju dekoder berupa indeks energi bingkai (R_0) yang telah dikuantisasi 32 tingkat. Nilai indeks ini kemudian digunakan untuk mencari nilai R_{dB} yang berada di daerah minimum -64 (nilai indeks 1 untuk R_0) sampai dengan maksimum 4 (nilai indeks 31 untuk R_0) dengan interval 2 dB dan energi 0 untuk $R(0) = 0$. Nilai R_0 berupa nilai integer dari 0 sampai 31.

Setelah didapatkan nilai R_{dB} , maka kita perlu mengkonversikannya kembali menjadi nilai energi bingkai $R(0)$ dengan menggunakan Persamaan 3.8.,

$$R_{dB} = 10 \log_{10} \left(\frac{R(0)}{R_{max}} \right) \quad (3.8)$$

Proses interpolasi juga dilakukan pada parameter energi bingkai, $R(0)$. $R'_q(0)$ didefinisikan sebagai nilai dari bingkai yang telah diproses dan $R_q(0)$ adalah nilai koefisien dari bingkai sebelum diproses, dengan ketentuan yang terdapat pada Persamaan 3.9 – 3.10.,

$$R'_q(0) = R_q(0)_{\text{bingkai sebelumnya}} \quad \text{untuk subbingkai 1} \quad (3.9)$$

$$R'_q(0) = R_q(0)_{\text{bingkai saat ini}} \quad \text{untuk subbingkai 3 \& 4} \quad (3.10)$$

$$R'_q(0) = \sqrt{R_q(0)_{\text{bingkai sebelumnya}} \cdot R_q(0)_{\text{bingkai saat ini}}} \quad \text{untuk subbingkai 2} \quad (3.11)$$

3.3. Pendekodean Lag

Nilai lag, L , untuk masing – masing keempat subbingkai bernilai dari 20 sampai 146, sehingga terdapat 127 kode. Untuk mengkodekan masing – masing lag memerlukan 7 bit dimana kode ke 128 digunakan untuk mengindikasikan bahwa prediksi *pitch* untuk subbingkai tidak aktif.

Pada dekoder, nilai lag, L , untuk subbingkai x dapat ditentukan dari kode LAG_x sebagai berikut:

$$L = LAG_x + 19 \quad \text{jika } LAG_x \neq 0$$

$$\text{Prediktor jangka panjang tidak aktif} \quad \text{jika } LAG_x = 0$$

3.4. Pendekodean Codeword Eksitasi

Pada enkoder, dilakukan pencarian kombinasi dari tabel kode eksitasi yang menghasilkan error pembobotan minimum terhadap sinyal residu, $p(n)$. Kemudian indeks dari nilai *codeword* eksitasi yang terpilihlah yang dikirimkan ke dekoder. Karena itu, pada dekoder dilakukan proses pencarian *codeword* eksitasi pada tabel kode eksitasi (Lampiran 3 dan 4) berdasarkan nilai indeks yang dikirimkan dari enkoder.

Nilai kode untuk tabel kode pertama, $CODE1_x$, adalah *codeword* I yang diturunkan melalui prosedur pencarian tabel kode. Bit terkecil merupakan nilai vektor basis pertama pada tabel kode. Sedangkan, nilai kode untuk tabel kode kedua, $CODE2_X$, adalah *codeword* H yang diturunkan melalui prosedur pencarian tabel kode, yang mana nilai bit terkecil merupakan vektor basis pertama pada tabel kode.

3.5. Pendekodean dan Transformasi GS, P0, P1 Menjadi Penguatan

Nilai indeks GSP0P1 yang dikirimkan dari encoder menuju dekoder, digunakan untuk mendapatkan nilai GSP0P1 yang terdapat pada tabel GSP0P1 pada Lampiran 5. Kemudian nilai GSP0P1 ini ditransformasi menjadi nilai – nilai penguatan ($\beta, \gamma_1, \gamma_2$). Transformasi penguatan ini digunakan untuk mengecilkan data atau informasi penguatan dari ketiga tabel kode.

Untuk mentransformasikan nilai GS, P0, P1 menjadi nilai β , γ_1 , dan γ_2 dapat menggunakan Persamaan 3.12 – 3.14 untuk prediksi jangka panjang yang aktif,

$$\beta = \sqrt{\frac{RS \cdot GS \cdot P0}{R_x(0)}} \quad (3.12)$$

$$\gamma_1 = \sqrt{\frac{RS \cdot GS \cdot P1}{R_x(1)}} \quad (3.13)$$

$$\gamma_2 = \sqrt{\frac{RS \cdot GS (1-P0-P1)}{R_x(2)}} \quad (3.14)$$

Kondisi khusus timbul ketika prediksi jangka panjang tidak aktif untuk subbingkai tertentu. Hal ini diakibatkan ketika tidak ada korelasi positif yang ditemukan selama pencarian lag atau ketika keadaan prediksi jangka panjang sebagian besar nilainya 0. Maka itu, untuk kondisi dengan prediksi jangka panjang yang tidak aktif, digunakan Persamaan 3.15 - 3.17.,

$$\beta = 0 \quad (3.15)$$

$$\gamma_1 = \sqrt{\frac{RS \cdot GS \cdot P1}{R_x(1)}} \quad (3.16)$$

$$\gamma_2 = \sqrt{\frac{RS \cdot GS (1-P1-P0)}{R_x(2)}} \quad (3.17)$$

Pada persamaan transformasi penguatan tersebut, RS adalah aproksimasi energi sinyal residu dan $R_x(k)$ adalah aproksimasi energi vektor eksitasi. Dengan nilai- nilai GS, P0, dan P1 yang terdapat dalam tabel GSP0P1. [1],[2],[4],[5],[8],[9],[11],[13].

3.6. Perhitungan Kombinasi Eksitasi dan Pembaharuan Keadaan Filter Jangka Panjang (*Long Term Filter State*)

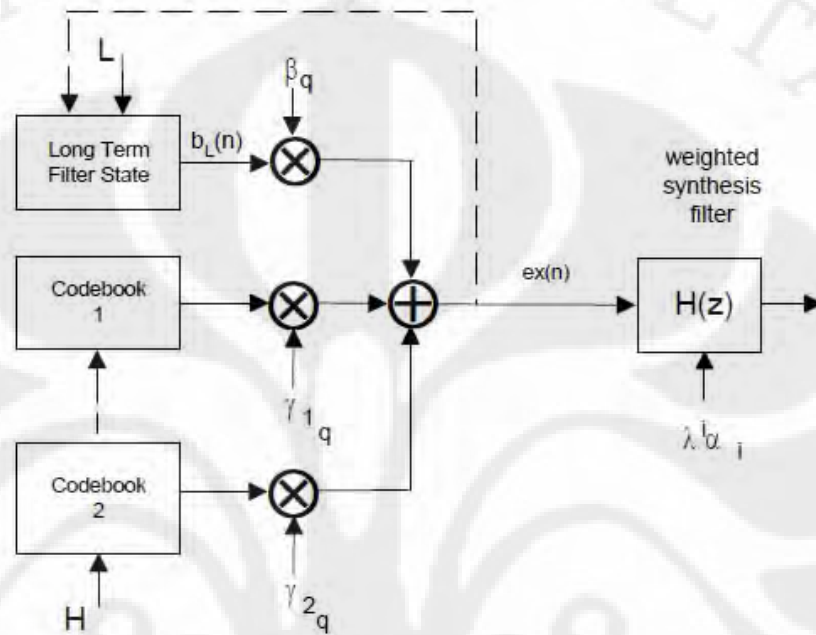
Kombinasi eksitasi, $ex(n)$, dihitung menggunakan Persamaan 3.18.,

$$ex(n) = \beta_q b_L(n) + \gamma_1 c_1 + \gamma_2 c_2 \quad (3.18)$$

dimana c_1 dan c_2 adalah kode buku eksitasi, $b_L(n)$ adalah vektor filter state, dan dengan penguatan yang terkuantisasi. Pembaharuan kode buku adaptif akan dilakukan dengan Persamaan 3.19. – 3.20,

$$r(n) = r(n + 40) \quad ; -147 < n < 41 \quad (3.19)$$

$$r(n) = ex(n + 40) ; -41 < n < -2 \quad (3.20)$$



Gambar 3.2. Proses Pembaharuan Kode Buku Adaptif [8]

3.7. Adaptive Pitch Prefilter

Sinyal kombinasi eksitasi, $ex(n)$, diproses menggunakan *adaptive pitch prefilter* untuk meningkatkan periodesitas lag dari sinyal eksitasi. Transfer fungsi dari *adaptive pitch prefilter* digambarkan melalui Persamaan 3.21.,

$$H_p(z) = \frac{1}{1 - \zeta z^{-L}} \quad (3.21)$$

dimana $\zeta = \varepsilon \text{Min}[\beta, \sqrt{P_0}]$ dan $\varepsilon = 0,4$

3.8. Filter Sintesis

Kombinasi eksitasi, $ex(n)$, difilter dengan filter sintesis untuk menghasilkan sinyal suara. Filter sintesis berupa filter *all-pole* orde 10 dengan koefisien filternya untuk tiap subbingkai menggunakan koefisien prediksi, α_i , yang telah didefinisikan pada subbab 3.1. Filter sintesis menggunakan filter jenis

direct form IIR. Transfer fungsinya digambarkan melalui Persamaan 3.22.,

$$A(z) = \frac{1}{1 - \sum_{i=1}^{N_p} \alpha_i z^{-i}} \quad (3.22)$$

3.9. Adaptive Spectral Postfilter

Kualitas persepsi dari sinyal sintesis dapat ditingkatkan dengan menggunakan dengan menggunakan *adaptive spectral postfilter* pada tahap proses terakhir. Bentuk persamaan postfilter spektral adaptif dijelaskan, bila diketahui transfer fungsi prediktor LPC $P(z) = \sum_{i=1}^M \alpha_i z^{-i}$, dimana α_i adalah koefisien prediksi LPC ke- i dan M adalah orde dari prediktor LPC, yang biasanya dipilih 10. Maka filter sintesis LPC yang bersesuaian memiliki transfer fungsi $1/[1 - P(z)]$, dan respons frekuensinya biasanya mengacu sebagai spektrum LPC. Gambar 4.2. memperlihatkan contoh dari spektrum LPC untuk suara berbunyi (*voiced sound*).

Pembilang polinomial pada persamaan tersebut merupakan versi yang telah dilembutkan secara spektral dari penyebut polinomial. Untuk menurunkan koefisien pembilang, autokorelasi dari respon impuls filter *all-pole* sesuai dengan penyebut dari Persamaan (3.23) dihitung untuk lag 0 sampai 10. Urutan autokorelasi kemudian di-*window* menggunakan *binomial window* dan koefisien pembilang (η_i untuk $i = 1$ sampai 10) dihitung dari urutan autokorelasi yang telah di-*window* via algoritma rekursi Levinson. Alternatif lainnya, koefisien autokorelasi dapat dihitung secara langsung dari koefisien via rekursi yang berhubungan dengan rekursi Levinson.

Untuk memiliki kontrol yang lebih pada kejelasan suara postfilter, sebuah filter orde satu yang dinyatakan dengan Persamaan 3.24.,

$$\hat{H}(z) = 1 - \mu z^{-1} \quad (3.24)$$

Filter ini dipasang berurutan dengan postfilter. Parameter berikut ini digunakan untuk postfilter, $\nu = 0,8$, $B_{eq} = 1200$ Hz dan $\mu = 0,4$. B_{eq} ialah faktor perluasan *bandwidth* yang menentukan derajat pelembutan yang dilakukan pada penyebut untuk menghasilkan pembilang.

Koefisien pelembutan spektral (*window* autokorelasi), $w_p(i)$ untuk 1200 Hz yaitu

Tabel 3.1. Koefisien pelembutan spektral, wp(i) [8]

wp(0)	1.000000
wp(1)	0.923077
wp(2)	0.725275
wp(3)	0.483516
wp(4)	0.271978
wp(5)	0.127990
wp(6)	0.049774
wp(7)	0.015718
wp(8)	0.003930
wp(9)	0.000748
wp(10)	0.000102

Untuk mengurangi permintaan komputasi dari koefisien pembilang yang telah dilembutkan secara spektral, *Spectral Smoothing Operation* (SST) dapat dilakukan satu kali per bingkai pada koefisien penyebut sesuai dengan koefisien yang belum diinterpolasi. Hal ini akan menghasilkan koefisien untuk pembilang dari postfilter spektral untuk subbingkai empat. Koefisien pembilang untuk subbingkai satu, dua dan tiga diinterpolasi menggunakan skema interpolasi yang sama seperti yang digunakan pada koefisien sintesis LPC pada subbab 3.1.1.

Untuk memastikan kesatuan daya penguatan antara masukan, $\hat{s}(n)$, dan keluaran, $\hat{s}_p(n)$, dari postfilter spektral, faktor skala penguatan dihitung dan digunakan untuk menskalakan sinyal keluaran postfilter. S_{scale} , faktor skala postfilter dinyatakan melalui Persamaan 3.25.,

$$S_{scale} = \frac{\sqrt{\sum_{n=0}^{N-1} \hat{s}^2(n)}}{\sqrt{\sum_{n=0}^{N-1} \hat{s}_p^2(n)}} \quad (3.25)$$

Faktor skala, S_{scale} , adalah akar kuadrat rasio energi sinyal masukan terhadap energi sinyal keluaran di sepanjang subbingkai. Faktor skala ini tidak digunakan secara langsung. Faktor skala dilewatkan melalui filter *lowpass* orde 1. Persamaan filter ini dinyatakan melalui Persamaan 3.26.,

$$S'_{scale} = (0,9875 * S'_{scale}(n-1) + (0,0125 * S_{scale})) \quad (3.26)$$

Keluaran dari postfilter spektral, $\hat{s}_p(n)$, kemudian dikalikan dengan S'_{scale} sebagai langkah terakhir dari rekonstruksi sinyal suara pada dekoder yang menghasilkan sinyal sintesis.



BAB IV

SIMULASI DAN ANALISA PROSES DEKODER VSELP

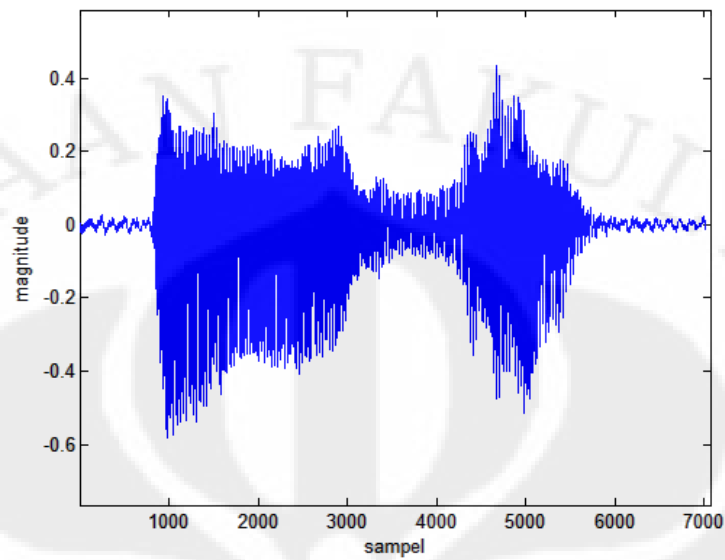
Pada simulasi dekoder pengkodean suara VSELP dilakukan dengan perangkat lunak SIMULINK oleh MATLAB. Dua puluh tujuh parameter VSELP perbingkai yang dikirimkan dari enkoder ditempatkan di dalam file transmisi data1.mat. Contoh suara yang digunakan adalah kata “saya” dan kalimat “saya pergi ke kampus” yang diolah dengan proses analisis-sintesis pada enkoder, sehingga menghasilkan sinyal suara sintesis. Proses simulasi dilakukan untuk memberikan gambaran pembangkitan dan rekonstruksi sinyal VSELP yang ditransmisikan dari enkoder menuju dekoder.

Kata “saya” mempunyai informasi suara sebanyak 7026 sampel, pencuplikan 8000 Hz dengan durasi 22 ms dan kalimat “saya pergi ke kampus” memiliki 11739 sampel dengan pencuplikan 8000 Hz. File suara terkode untuk kata tersebut 37 bingkai analisa suara yang masing – masing terdiri dari 27 parameter.

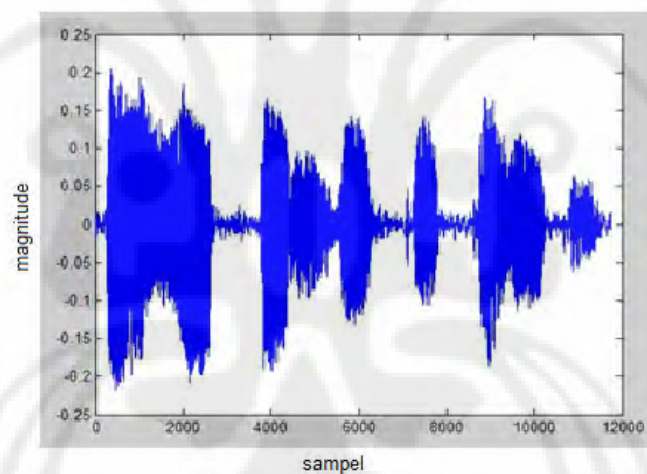
4.1. Simulasi Proses Pengolahan Suara pada Rangkaian Dekoder VSELP

Simulasi rangkaian dekoder VSELP menggunakan peranti lunak Simulink dilakukan secara *offline* atau dilakukan pengkompresian tidak secara kontinu. Simulasi dilakukan dengan terlebih dahulu merekam suara dan melakukan satu persatu kompresi bukan dilakukan dengan data suara yang kontinu dan langsung dikompresi.

Proses Simulasi yang dilakukan mempergunakan sumber suara rekaman. Sumber suara tersebut direkam mempergunakan alat perekam dengan format 32 bit kuantisasi PCM dan 8000 Hz. Format data suara tersebut sudah memberikan pemodelan terhadap data suara pada komunikasi GSM. Sampel sinyal suara rekaman kata “saya” terhadap basis waktu terdapat pada Gambar 4.1.



Gambar 4.1 Sampel Suara 'Saya' dengan basis waktu



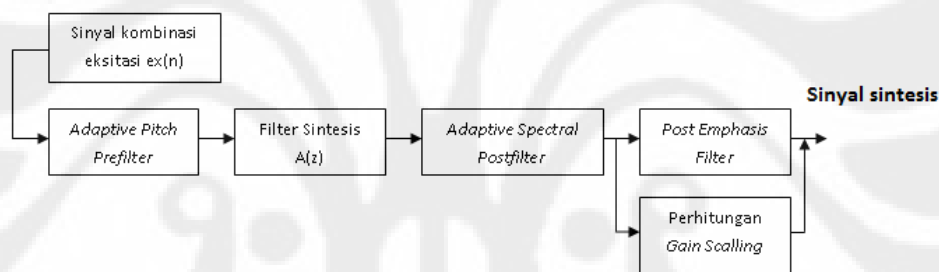
Gambar 4.2 Sampel Suara 'Saya pergi ke kampus' dengan basis waktu

Dua puluh tujuh parameter bingkai dari sinyal suara yang telah diolah pada enkoder diekstraksi sesuai dengan fungsinya masing – masing, sehingga dapat digunakan untuk proses analisis dan sintesis pada dekoder. Pada proses analisis dilakukan proses penyaringan LPC dan pada proses sintesis dilakukan untuk menghasilkan sinyal suara sintesis (subbab 3.1. – 3.6.).

Setelah didapatkan sinyal suara sintesis pada dekoder, kemudian sinyal suara sintesis ini ditingkatkan kualitas suaranya. Hal ini diperlukan karena sinyal

yang dienkodkan dengan pengkodean suara laju bit rendah menghasilkan sinyal suara dengan nilai SNR yang rendah, tingkat *noise* yang tinggi mendekati *white noise* sehingga sangat sulit untuk menjaganya dibawah nilai ambang batas kemampuan suara yang dapat didengar [14], [15].

Pendekatan perceptual yang didorong untuk diusulkan dalam menyelesaikan masalah ini adalah menggunakan filter adaptif pada keluaran pendekodean sinyal. Penggunaan filter adaptif dibandingkan filter tetap didasarkan pada kebutuhan untuk merubah operasi penyaringan yang disesuaikan dengan karakter lokal dari spektrum suara. *Postfilter* ini dibangun dengan tujuan pelemahan antara puncak – puncak formant dan puncak harmonik *pitch* tapi tanpa merusakkan spektral. Proses penyaringan sinyal suara sintesis menggunakan filter diperlihatkan pada Gambar 4.2.

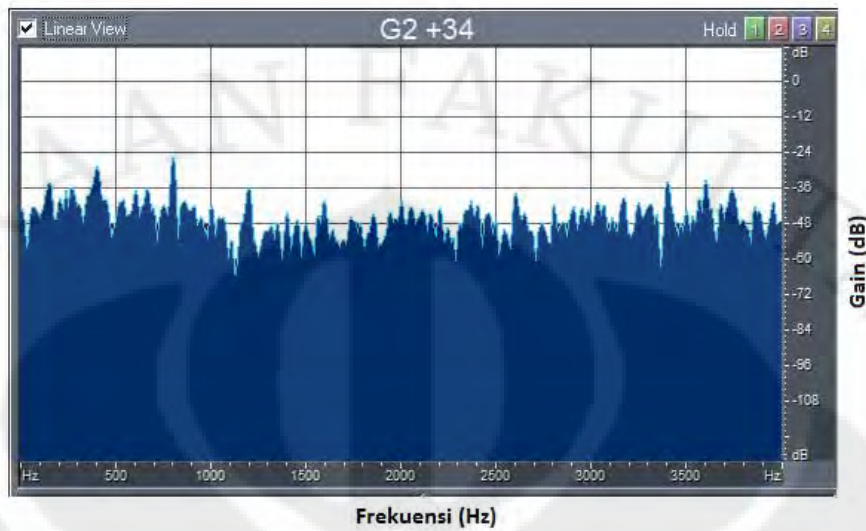


Gambar 4.3. Proses penyaringan suara sintesis pada dekoder VSELP

4.1.1. Adaptive Pitch Prefilter

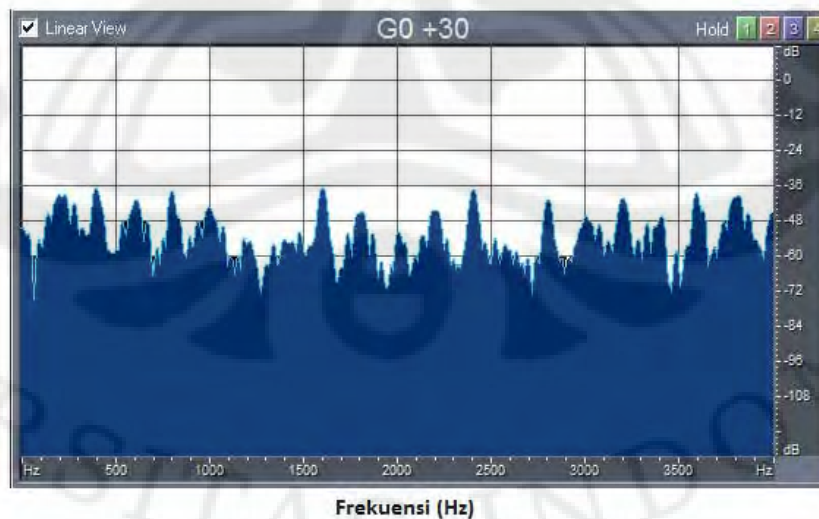
Dekoder VSELP menghasilkan kombinasi sinyal eksitasi, $ex(n)$, dari *long term filter state* dan dua tabel kode eksitasi. Kombinasi eksitasi ini kemudian diproses dengan sebuah *adaptive pitch prefilter* (Persamaan 3.21).

Adaptive pitch prefilter berfungsi untuk meningkatkan amplitudo gelombang pada frekuensi tertentu sehingga dengan kata lain mempertegas nilai lag. Peningkatan periodesitas dilakukan pada residu sinyal sintesis daripada gelombang sinyal sintesis. Hal ini dilakukan untuk mengurangi secara signifikan kerusakan pada sinyal yang direkonstruksi dalam kaitannya dengan bentuk gelombang diskontinyu yang terkadang timbul karena *postfiltering* pada *pitch*.



Gambar 4.4. Spektrum kombinasi sinyal eksitasi, $ex(n)$

Pada simulasi dekoder VSELP didapatkan bentuk spektrum keluaran *adaptive pitch prefilter* pada Gambar 4.3. Dari Gambar 4.3. dapat kita lihat bahwa terjadi peningkatan nilai lag pada frekuensi – frekuensi tertentu. Maka hal sesuai dengan fungsi *adaptive pitch filter* yaitu mempertegas nilai lag [4].

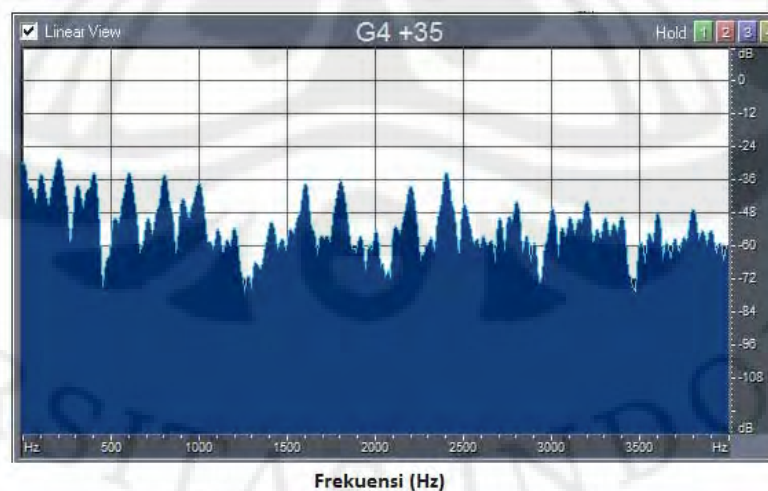


Gambar 4.5. Sinyal keluaran *adaptive pitch prefilter* pada simulasi dekoder VSELP

4.1.2. Filter sintesis LPC, $A(z)$

Filter sintesis, $A(z)$, digunakan untuk menghasilkan sinyal suara (*speech*) dari kombinasi sinyal eksitasi yang telah dikuatkan nilai lagnya. Hal ini dikarenakan filter sintesis LPC memiliki respons frekuensi yang mendekati *envelope* spektral dari sinyal suara masukan (sinyal LPC), dimana pada fungsi transfernya menggunakan koefisien prediktor LPC. Koefisien filter akan berganti dari subbingkai ke subbingkai lainnya, sehingga kondisi filter (*filter state*) harus dapat menjaga antar pergantian koefisien subbingkai. Gambar 4.4. menunjukkan bentuk keluaran spektrum dari filter sintesis.

Mengacu pada subbab 2.3.1. apabila diberikan sinyal digital $S(z)$ dan dilewatkan pada filter LPC dengan fungsi alih $A(z)$, maka dengan mengacu pada Gambar 2.5. akan didapatkan sinyal keluaran $E(z)$. $E(z)$ didefinisikan sebagai sinyal sisa (residu) yang dalam hal ini adalah kombinasi sinyal eksitasi, $ex(n)$, dari sinyal $S(z)$ yang dilewatkan pada filter LPC dengan fungsi alih, $H(z) = E(z)/S(z) = A(z)$. Dengan menghasilkan nilai fungsi spektral $E(z)$ yang sama untuk setiap saat maka apabila $E(z)$ dilewatkan pada filter LPC dengan fungsi alih, $H(z) = S(z)/E(z) = 1/A(z)$ pada dekoder, maka akan didapatkan $S(z)$ yang memiliki kualitas sinyal sintesis yang sama dengan kualitas sinyal aslinya. Konsep inilah yang menjadi acuan dasar bagi pengkodean suara dengan menggunakan filter sintesis LPC sebagai filter yang mewakili rongga ucapan atau filter sintesis.



Gambar 4.6. Sinyal keluaran filter sintesis, $A(z)$

4.1.3. *Postfilter* Spektral Adaptif

Respons frekuensi dari *postfilter* jangka pendek atau *postfilter* spektral adaptif yang ideal seharusnya dapat mengikuti puncak dan lembah *envelope* spektral dari sinyal suara tanpa memberikan kerusakan menyeluruh yang berarti. Pada pengkodean suara prediktif mempergunakan prediksi linier, filter sintesis (biasa disebut filter LPC) memiliki respons frekuensi yang mengikuti secara dekat *envelope* spektral dari sinyal masukan. Karena itu, untuk menurunkan persamaan *postfilter* jangka pendek diambil dari prediktor LPC.

Postfilter spektral adaptif pada dekoder VSELP merupakan *postfilter* pole-zero dimana penyebutnya $1 - 1/[1 - P(z/\alpha)]$ dimana α pada Persamaan 3.23 dengan nilai 0,8. Nilai α yang optimal tergantung pada laju bit dan tipe pengkodean suara yang digunakan, dan nilai ini umumnya ditentukan secara empiris berdasarkan tes pendengaran subjektif.

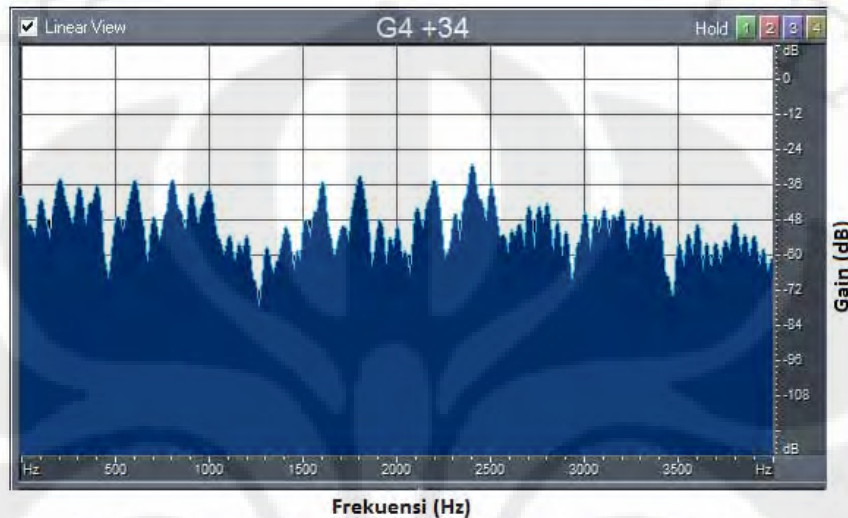
Bagian penyebut ini merepresentasikan penekanan *pitch* pada spektrum sinyal sintesis, ketika dilakukan penipisan pada lembah spektral. Pembilang mencoba untuk membatalkan bentuk spektral umum yang ditimbulkan oleh penyebut.

Dalam prakteknya, polinomial pembilang hanya sukses sebagian dalam melacak bentuk spektral dari penyebut (sebagai efek, karakteristik spektral rusak terhadap waktu), dan perbedaan ini secara khas menunjukkan sendiri sebagai modulasi yang bervariasi terhadap waktu dari kejelasan suara yang telah di-*postfilter*.

Di dalam satu perwujudan koefisien filter yang merepresentasikan penyebut dikonversi ke dalam domain autokorelasi. Teknik pelembutan spektral yang dibuat menggunakan fungsi perluasan *bandwidth* diaplikasikan kepada urutan autokorelasi melalui rekursi Levinson.

Perluasan *bandwidth pitch (fomant)* tidak mengubah *envelope* spektral yang dilembutkan. Dengan begitu, pembilang yang telah dilembutkan spektralnya efektif dalam meminimalkan kerusakan spektral yang bervariasi terhadap waktu dan mengizinkan pembilang secara adaptif melacak bentuk spektral umum penyebut dan membatalkannya [13].

Gambar di bawah ini merupakan keluaran filter spektral adaptif, dimana masukan filternya merupakan sinyal yang tergambarkan pada Gambar 4.4. Pada Gambar 4.5. ini dapat kita lihat bahwa terjadi pelembutan pada puncak spektrum dan lembah dari sinyal masukan.

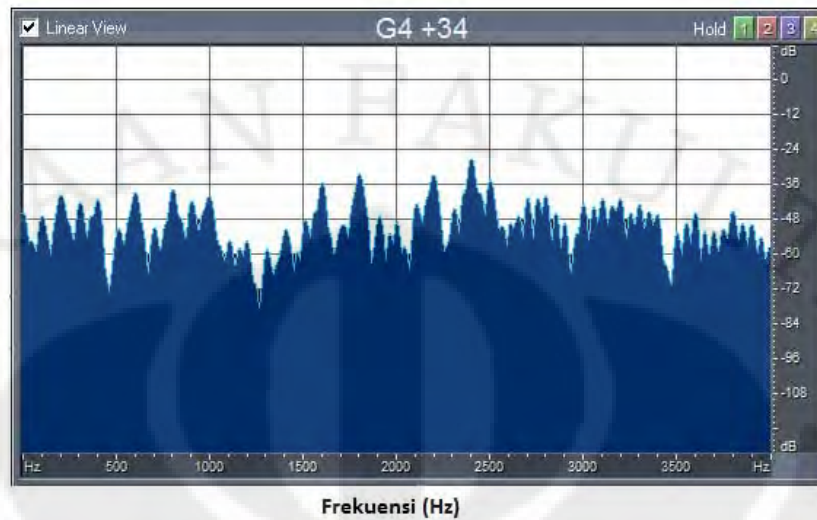


Gambar 4.7. Sinyal keluaran *postfilter* spektral adaptif

4.1.4. Filter *Post Emphasis*

Pada tes pendengaran secara informal, didapatkan suara masih sedikit teredam. Untuk pengurangan lebih lanjut, kita menambahkan filter orde 1 dengan fungsi transfer $[1 - uz^{-1}]$ beriringan dengan filter adaptif spektral.

Pada dekoder VSELP, filter ini bersifat adaptif karena ditujukan untuk dapat melacak kerusakan spektrum lebih baik lagi dari keluaran filter sebelumnya. Selain itu, filter ini disediakan untuk dapat lebih banyak melakukan pengontrolan kejelasan (*brighthness*) suara *postfilter*.

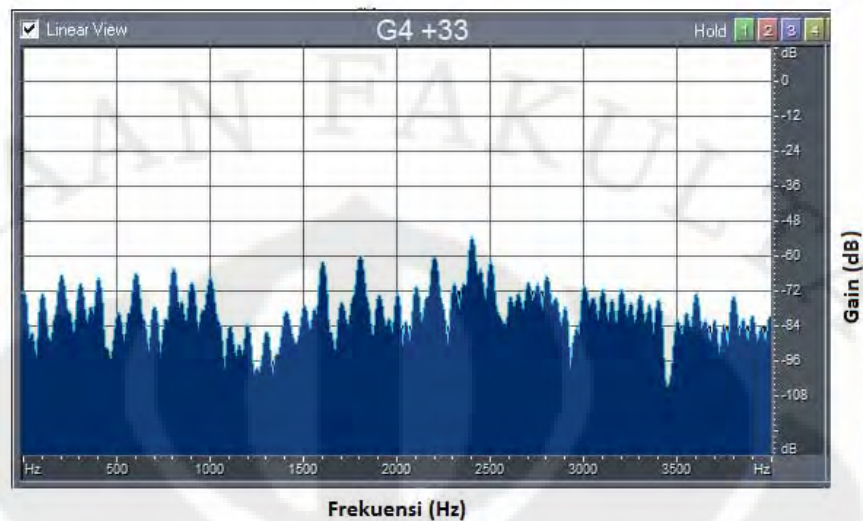


Gambar 4.8. Sinyal keluaran filter *post emphasis*

4.1.5. Pengontrolan Penguatan

Metode pengontrolan penguatan otomatis diperlukan untuk memastikan penguatan menyeluruh di sepanjang spektrum keluaran filter. Nilai penguatan dihitung dengan menggunakan Persamaan 3.26. Dengan demikian, seluruh bingkai sekarang secara kasar memiliki penguatan daya yang sama, sehingga suara sinyal yang telah difilter terdengar lebih jelas.

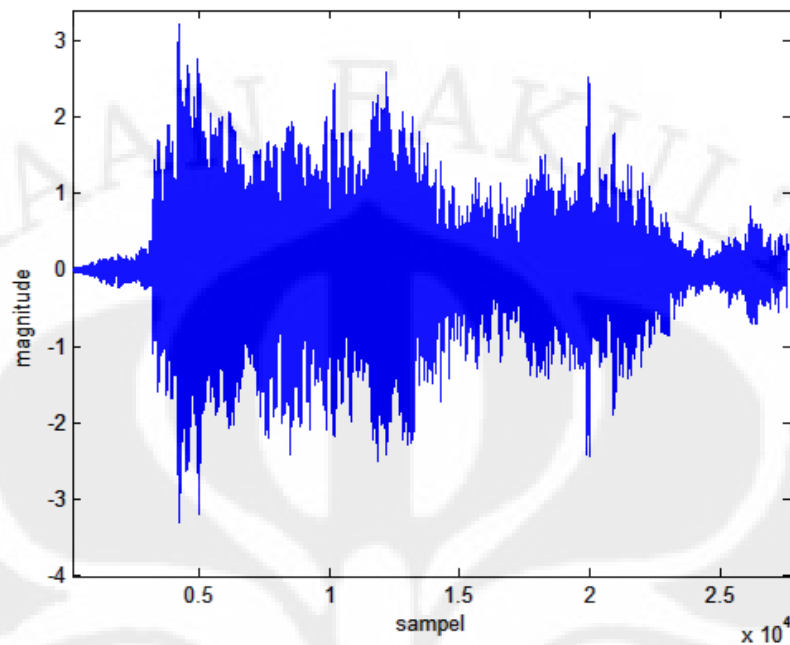
Pengaruh nilai penguatan ini terhadap spektrum suara dapat dilihat pada Gambar 4.7. Terlihat pada grafik ini dibandingkan masukannya amplitudo meningkat di sepanjang spektrumnya.



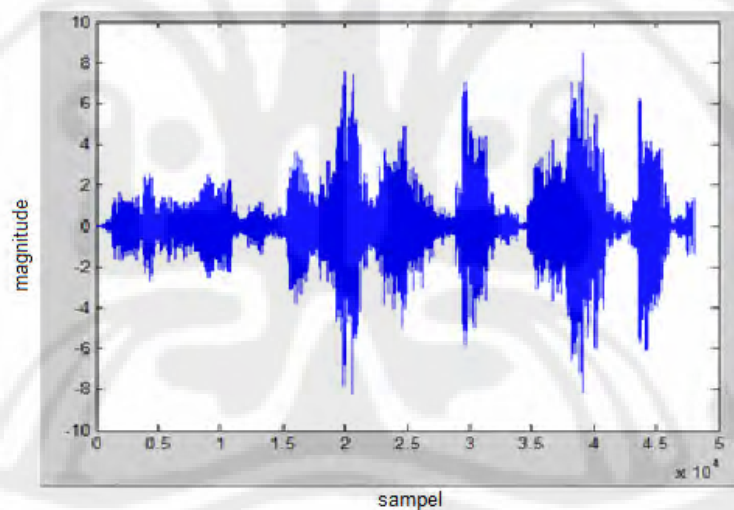
Gambar 4.9. Spektrum sinyal sintesis dekoder VSELP

4.2. Hasil Simulasi

Simulasi rangkaian dekoder bertujuan untuk mendapatkan sinyal sintesis yang dibangkitkan dan melihat proses – proses pengekstrasian parameter serta rekonstruksi sinyal sintesis tersebut. Pada bentuk gelombang hasil keluaran dekoder VSELP terlihat kecocokan bentuk periodisitas nilai lag pada gelombang sinyal suara tersebut, sebagai pemberi informasi yang utama pada pengkompresian suara, dengan Gambar 4.1. dan Gambar 4.2. yang merupakan bentuk sinyal asli kata “saya” dan untuk kalimat “saya pergi ke kampus”, meskipun besarnya gain belum maksimal. Namun, terdapat perbedaan yang besar pada waktu perambatan sinyal suara yang mengakibatkan pelebaran sinyal suara terhadap basis waktu. Sehingga, hal ini menyebabkan tes pendengaran informal dari sinyal sintesis mengalami perlambatan dalam pelafalan kata “saya” dan juga terdengar banyak *noise* bila dibandingkan dengan sinyal aslinya. Jumlah sampel sinyal keluaran dekoder pun meningkat dari 6000 sampel menjadi 24040 sampel untuk kata “saya” dan dibandingkan sinyal suara aslinya.



Gambar 4.10. Sinyal sintesis suara “saya” dekoder VSELP



Gambar 4.11. Sinyal sintesis suara “saya pergi ke kampus” dekoder VSELP

Proses simulasi rangkaian dekoder ini dengan sumber suara “saya” membutuhkan waktu sekitar 5 detik dan untuk suara “saya pergi ke kampus” membutuhkan waktu sekitar 16,8 detik. Proses rangkaian dekoder memerlukan proses yang panjang karena mempergunakan algoritma yang tidak sederhana sehingga waktu yang dibutuhkan untuk memproses suara lebih lama. Waktu untuk pemrosesan kalimat dibandingkan dengan kata pun juga menjadi lebih panjang.

Hal tersebut tidak mungkin dapat dilakukan untuk memberikan proses komunikasi yang baik karena keterlambatan penerimaan suara akan terjadi.

4.3. Analisa Proses Rangkaian Dekoder VSELP

Sumber daya frekuensi yang terbatas dapat teratasi dengan penggunaan kompresi suara VSELP tersebut. Proses kompresi yang mencapai 8 kbps akan mampu memberikan kapasitas yang cukup pemanfaatan kanal suara. Namun dari hasil simulasi, didapatkan hasil sinyal keluaran sintesis pada dekoder VSELP mengalami perlambatan dalam pelafalan kata “saya” dan “saya pergi ke kampus” dan terdapat banyak *noise* bila dibandingkan dengan sinyal suara aslinya. Hal ini dikarenakan perbedaan *sampling time* di dalam percobaan ini. Pada sinyal suara asli dalam satu detik diambil 8000 sampel dan diproses untuk setiap 40 data, sedangkan pada percobaan ini dalam 1 detik diambil 27 sampel untuk diproses setiap 40 data. Perbedaan *sampling time* ini dikarenakan pada bagian dekoder melakukan ekstraksi 27 parameter yang dikirimkan dari enkoder untuk diproses lebih lanjut. Namun, bila dalam simulasi dekoder ini disatukan dengan enkoder dalam *file* model yang sama, keterlambatan pelafalan ini dapat diatasi. Alternatif lainnya yaitu dengan menggunakan bahasa pemrograman yang berbeda yang tidak dipengaruhi dengan permasalahan *sampling time* (waktu pencuplikan).

Selain itu, didapatkan proses dekoder VSELP juga membutuhkan waktu yang cukup lama bila mengacu pada hasil simulasi tersebut. Proses komputasi yang panjang harus diatasi agar keterlambatan penerimaan suara tidak terjadi pada rangkaian penerima. Algoritma proses dekoder harus dipersingkat dan kemampuan prosesor yang cepat harus dipergunakan untuk dapat mengatasi masalah tersebut.

BAB 5

KESIMPULAN

Dari pembahasan yang telah dilakukan, dapat disimpulkan:

1. Sinyal suara sintesis pengkodean suara laju bit rendah memiliki tingkat *noise* yang tinggi mendekati *white noise*, sehingga pada proses dekoder VSELP memerlukan tambahan *postfilter* untuk memperbaiki sinyal suara.
2. Fungsi dari filter yaitu membuang bagian – bagian yang tidak diinginkan dari suatu sinyal, seperti *noise* acak, atau mengekstraksi bagian yang berguna dari sinyal seperti komponen yang terdapat pada daerah frekuensi tertentu.
3. Sinyal sintesis hasil simulasi ini mengalami perlambatan dalam pelafalan kata bila dibandingkan dengan suara aslinya, namun sudah didapatkan kecocokan pada bentuk *envelope*-nya.
4. Hasil simulasi yang didapatkan tidak maksimal dikarenakan adanya perbedaan waktu pencuplikan suara asli pada percobaan ini, sehingga disarankan untuk menggunakan bahasa pemrograman yang berbeda yang tidak dipengaruhi oleh waktu pencuplikan.
5. Proses dekoder memerlukan komputasi yang kompleks sehingga waktu untuk proses pendekodean cukup lama dan diperlukan prosesor yang cepat untuk menangani proses komputasi VSELP.

DAFTAR REFERENSI

- [1] Chu, C Wai. 2003. *Speech Coding Algorithms : Foundation and Evolution of Standardized*. New Jersey : John Wiley & Sons Inc
- [2] Saleh, Muhammad. 1996. *Implementasi Teknik VSELP Pada Pengkodean Suara Laju Bit Rendah*. Depok: Universitas Indonesia
- [3] Wong, W. T K, R.M. Mack, B.M.G. Cheetham dan X.Q. Sun. 1996. *Low Rate Speech Coding*. Inggris : Liverpool University
- [4] Gerson, Ira A, Mark A Jasiuk. April 1990. *Vector Excited Linear Prediction (VSELP) Speech Coding at 8 kbps*. Proc. IEEE Int. Conf. Acoustic, Speech and Signal Processing.
- [5] Gerson, Ira A, Mark A Jasiuk. 1990. *Vector Excited Linear Prediction (VSELP) Speech Coding at 4,8 kbps*. Chicago Corporate Result and Development Centre Motorola Inc.
- [6] Li, Wei, Anu Sridhar, Tina Teng. 1999. *Comparison of Speech Coding Algorithms: ADPCM, CELP, and VSELP*. Project for EE 6390, University of Texas
- [7] Le Blanc, W.P dan S.A. Mahmoud. *Structured Codebook Design in CELP*. Carleton University. Ottawa, Kanada
- [8] TIA/EIA Standard 136-420. 1999. IS-54. TDMA Celluler/PCS VSELP.
- [9] Chen, Juin-Hwey and Allen Gersho. (Januari 1995). Adaptive Postfiltering for Quality Enhancement of Coded Speech. *Journal of IEEE Transactions on Speech and Audion Processing*, Vol.3, No.1.
- [10] Macres, Jason Victor. 1994. *Theory and Implementation of the Digital Cellular Standard Voice Coder: VSELP on the TMS320C5x*. Texas Instruments. Oktober
- [11] *Telecommunication Application With the TMS320C5x DSPs*. 1994. Texas Instruments.
- [12] Chrshtie, Mansoor A. Oktober 1994. *U.S. Digital Cellular Error-Correction Coding Algorithm Implementation on the TMS320C5x*. Texas Instrument.

[13] Sunwoo, Myung H, Sangil Park, dan Kent Terry.1991. *A Real-Time Implementation of Key VSELP Routines on a 16-bit DSP Chip*. Motorola, Inc. Austin, Texas



DAFTAR PUSTAKA

- Chu, C Wai. 2003. *Speech Coding Algorithms : Foundation and Evolution of Standardized*. New Jersey : John Wiley & Sons Inc
- Chrshtie, Mansoor A. Oktober 1994. *U.S. Digital Cellular Error-Correction Coding Algorithm Implementation on the TMS320C5x*. Texas Instrument.
- Macres, Jason Victor. 1994. *Theory and Implementation of the Digital Cellular Standard Voice Coder: VSELP on the TMS320C5x*. Texas Instruments. Oktober
- Gerson, Ira A, Mark A Jasiuk. April 1990. *Vector Excited Linear Prediction (VSELP) Speech Coding at 8 kbps*. Proc. IEEE Int. Conf. Acoustic, Speech and Signal Processing.
- Gerson, Ira A, Mark A Jasiuk. 1990. *Vector Excited Linear Prediction (VSELP) Speech Coding at 4,8 kbps*. Chicago Corporate Result and Development Centre Motorola Inc.
- Gerson, Ira, Mark A, Jasiuk. Augustus 1993. Digital Speech Decoder Having A Postfilter With Reduced Spectral Distortion. *United State Patent*. 31. No. Patent: 5,241,650
- Le Blanc, W.P dan S.A. Mahmoud. *Structured Codebook Design in CELP*. Carleton University. Ottawa, Kanada
- Li, Wei, Anu Sridhar, Tina Teng. 1999. *Comparison of Speech Coding Algorithms: ADPCM, CELP, and VSELP*. Project for EE 6390, University of Texas
- Saleh, Muhammad. 1996. *Implementasi Teknik VSELP Pada Pengkodean Suara Laju Bit Rendah*. Depok: Universitas Indonesia
- Sunwoo, Myung H, Sangil Park. November 1991. *Real-Time Implementation of The VSELP on A 16-bit DSP Chip*. IEEE.
- Sunwoo, Myung H, Sangil Park, dan Kent Terry. 1991. *A Real-Time Implementation of Key VSELP Routines on a 16-bit DSP Chip*. Motorola, Inc. Austin, Texas
- TIA/EIA Standard 136-420. 1999. IS-54. TDMA Celluler/PCS VSELP.

Wong, W. T K, R.M. Mack, B.M.G. Cheetham dan X.Q. Sun. 1996. *Low Rate Speech Coding*. Inggris : Liverpool University



Lampiran 1: Parameter yang Dikodekan Untuk Setiap Bingkai 20 ms

Nama Parameter	Jumlah bit	Keterangan
R0	5	Energi bingkai
LPC 1	6	Koefisien refleksi 1
LPC 2	5	Koefisien refleksi 2
LPC 3	5	Koefisien refleksi 3
LPC 4	4	Koefisien refleksi 4
LPC 5	4	Koefisien refleksi 5
LPC 6	3	Koefisien refleksi 6
LPC 7	3	Koefisien refleksi 7
LPC 8	3	Koefisien refleksi 8
LPC 9	3	Koefisien refleksi 9
LPC 10	2	Koefisien refleksi 10
Lag_1	1	Lag subbingkai 1
Lag_2	1	Lag subbingkai 2
Lag_3	1	Lag subbingkai 3
Lag_4	1	Lag subbingkai 4
CODE 1_1	7	Indeks tabel kode eksitasi 1 subframe 1
CODE 1_2	7	Indeks tabel kode eksitasi 1 subframe 2
CODE 1_3	7	Indeks tabel kode eksitasi 1 subframe 3
CODE 1_4	7	Indeks tabel kode eksitasi 1 subframe 4
CODE 2_1	7	Indeks tabel kode eksitasi 2 subframe 1
CODE 2_2	7	Indeks tabel kode eksitasi 2 subframe 2
CODE 2_3	7	Indeks tabel kode eksitasi 2 subframe 3
CODE 2_4	7	Indeks tabel kode eksitasi 2 subframe 4
GSP0P1 1	8	Indeks tabel penguatan subframe 1
GSP0P1 2	8	Indeks tabel penguatan subframe 2
GSP0P1 3	8	Indeks tabel penguatan subframe 3
GSP0P1 4	8	Indeks tabel penguatan subframe 4

Lampiran 2: Tabel Kode Koefisien Kuantisasi

	LPC1	LPC2	LPC3	LPC4	LPC5
1	-0,9867044	-0,7547349	-0,8606231	-0,4505537	-0,6570091
2	-0,9810331	-0,5826730	-0,8046579	-0,2582354	-0,5339469
3	-0,9762308	-0,4569369	-0,7523137	-0,1276191	-0,4366474
4	-0,9711074	-0,3481136	-0,7056828	-0,0189162	-0,3503954
5	-0,9655631	-0,2492763	-0,6582848	0,0735594	-0,2729992
6	-0,9597059	-0,1585778	-0,6130494	0,1581760	-0,2044571
7	-0,9536623	-0,0772643	-0,5684005	0,2381239	-0,1395732
8	-0,9471911	-0,0050969	-0,5247685	0,3138437	-0,7738069
9	-0,9406522	0,0652718	-0,4832728	0,3864110	-0,1441931
10	-0,9339897	0,1329884	-0,4436872	0,4573097	0,4928109
11	-0,9266146	0,1978288	-0,4048591	0,5282919	0,1149447
12	-0,9190770	0,2600679	-0,3659545	0,5986876	0,1853068
13	-0,9110740	0,3186268	-0,3276270	0,6679506	0,2629969
14	-0,9032389	0,3747462	-0,2901177	0,7368154	0,3558021
15	-0,8951876	0,4288900	-0,2521926	0,8054330	0,4658669
16	-0,8865973	0,4810179	-0,2139451	0,8715001	0,6091661
17	-0,8775992	0,5302551	-0,1760749		
18	-0,8679655	0,5760165	-0,1370347		
19	-0,8578170	0,6193221	-0,0963786		
20	-0,8468435	0,6593915	-0,0557086		
21	-0,8350598	0,6967787	-0,0134266		
22	-0,8232460	0,7315258	0,0291324		
23	-0,8109311	0,7650170	0,0724380		
24	-0,7979615	0,7966733	0,1183170		
25	-0,7842678	0,8267812	0,1668848		
26	-0,7699829	0,8543012	0,2185972		
27	-0,7545527	0,8798118	0,2741918		
28	-0,7377605	0,9037305	0,3353715		
29	-0,7188396	0,9251339	0,4032742		
30	-0,6990101	0,9448361	0,4797808		
31	-0,6768131	0,9636774	0,5761803		
32	-0,6533867	0,9816108	0,6969622		
33	-0,6262213				
34	-0,5971571				
35	-0,5684631				
36	-0,5378258				
37	-0,5058868				
38	-0,4740323				
39	-0,4414438				
40	-0,4054018				
41	-0,3682330				
42	-0,3293100				
43	-0,2894667				

44 -0,2428349
 45 -0,1948891
 46 -0,1466630
 47 -0,0915260
 48 0,0269218
 49 0,0372767
 50 0,1093523
 51 0,1758577
 52 0,2397777
 53 0,3001486
 54 0,3555985
 55 0,4108348
 56 0,4679427
 57 0,5202129
 58 0,5746787
 59 0,6337371
 60 0,6966888
 61 0,7613552
 62 0,8211135
 63 0,8759806
 64 0,9311733

	LPC6	LPC7	LPC8	LPC9	LPC10
1	-0,3351014	-0,5834863	-0,3832901	-0,5373111	-0,1508205
2	-0,1157702	-0,4363778	-0,2111644	-0,3927697	0,0465244
3	0,0326587	-0,3007811	-0,0724298	-0,2594878	0,1958054
4	0,1545226	-0,1750740	0,0490140	-0,1391014	0,3621688
5	0,2685564	-0,0584889	0,1637184	-0,0293514	
6	0,3836386	0,0571851	0,2812095	0,0825513	
7	0,5067841	0,1880215	0,4174391	0,2107033	
8	0,6525010	0,3601036	0,5824320	0,3958447	

Lampiran 3: Tabel Kode Eksitasi 1

	BV 1	BV 2	BV 3	BV 4	BV 5	BV 6	BV 7
1	0,34512	-0,16493	0,12223	-0,20025	-0,64836	0,34142	-0,99150
2	-0,19089	-0,12806	-0,02444	0,30951	-0,01472	0,23058	-0,04214
3	-0,18638	0,20847	0,09521	0,11933	-0,39261	-0,11542	-0,20654
4	-0,28091	0,02897	0,20513	0,25020	-0,03888	0,04632	-0,06140
5	-0,05950	-0,12293	-0,26477	0,07345	0,02080	-0,06644	-0,02154
6	-0,28981	0,09743	-0,04311	0,28950	0,19575	0,08361	0,10041
7	-0,20070	0,05553	-0,27443	0,04940	-0,09889	-0,09363	0,18118
8	-0,26247	0,32947	0,04826	-0,16728	0,38250	-0,02741	0,05004
9	-0,23192	-0,13464	-0,14419	0,17612	0,08439	0,07113	0,25630
10	-0,18100	0,12565	0,17755	0,07342	0,30035	-0,12547	-0,15232
11	0,03789	-0,11684	0,22276	-0,17812	0,17688	0,07751	-0,02675
12	-0,21478	0,00623	0,41995	0,04386	0,22680	0,01797	-0,12967
13	0,12418	0,07953	0,34077	-0,04660	0,38401	0,50265	0,04938
14	-0,12186	0,31938	0,31649	-0,07254	-0,12177	0,07896	0,10620
15	0,02360	-0,27460	0,13951	-0,29067	0,14547	0,42794	0,19914
16	0,27133	-0,01363	-0,20506	-0,06229	-0,05188	0,21770	0,26956
17	0,16327	-0,17662	0,06458	-0,13099	0,11412	0,22893	-0,02002
18	-0,04970	-0,19182	-0,35379	0,07658	-0,02478	0,04618	0,29058
19	0,19028	-0,17463	-0,05985	0,17439	0,09596	0,16568	-0,00313
20	-0,01335	-0,20696	-0,29406	0,18551	0,14327	-0,05426	0,10164
21	0,09224	-0,15702	0,16444	0,35238	-0,05414	0,11283	0,12994
22	-0,05935	-0,26793	-0,09713	0,11465	0,00395	-0,03722	0,23030
23	0,02033	-0,02101	0,23452	0,15899	-0,25869	0,07740	0,29024
24	-0,20718	-0,20320	-0,11410	0,22222	-0,02090	-0,24436	0,18969
25	0,02096	-0,20447	0,32471	-0,01127	-0,21015	-0,18949	0,32532
26	-0,15494	-0,00484	-0,07059	-0,26819	-0,06068	0,14775	0,02585
27	-0,21406	-0,18205	-0,14674	-0,36624	-0,02704	-0,50434	-0,00263
28	-0,23299	-0,07024	-0,10935	-0,37178	0,06947	0,02714	-0,07936
29	-0,03249	-0,05201	0,12904	0,02992	0,13133	-0,13505	-0,09870
30	0,12201	-0,12365	-0,30674	-0,35340	0,06358	-0,03649	-0,16585
31	0,18095	0,10069	0,06986	-0,04227	0,02392	-0,10411	-0,10429
32	0,06743	-0,40631	0,33801	0,06729	0,06551	-0,27498	-0,15930
33	0,11957	0,13148	0,10607	-0,03312	0,05598	-0,05221	0,07002
34	0,39984	-0,00170	0,00421	0,35889	-0,04799	-0,33431	0,00916
35	0,27503	0,38494	0,25291	-0,23170	-0,06142	-0,34611	0,00556
36	0,13968	0,37618	-0,00532	0,11108	-0,31128	-0,03607	-0,02852
37	-0,22649	0,18326	-0,09390	-0,10489	-0,39184	0,08886	0,00439
38	0,24597	0,22552	-0,14084	0,06802	-0,08221	-0,02546	-0,07426
39	-0,14785	0,39523	-0,19451	0,03554	0,00293	0,09428	-0,11090
40	0,13496	0,19653	-0,03808	0,02773	-0,14520	-0,00650	-0,10565

Lampiran 4: Tabel Kode Eksitasi 2

	BV 1	BV 2	BV 3	BV 4	BV 5	BV6	BV 7
1	-1,07847	-0,61713	-0,70374	0,07885	0,04358	0,67365	-0,69421
2	0,32202	0,10425	0,17802	0,16417	-0,05806	0,18363	0,11140
3	0,35721	0,19043	0,13863	0,27696	0,45011	-0,11692	0,60137
4	0,60321	-0,20039	0,01774	0,31391	-0,20821	-0,23851	0,19518
5	0,32913	0,23642	0,31941	-0,18080	-0,35824	-0,03071	0,51339
6	-0,20682	-0,39313	-0,81562	-0,66968	-0,26136	0,36133	0,11329
7	-0,07851	-0,07452	-0,03730	0,23479	-0,67455	0,40771	0,14373
8	0,07766	-0,11137	0,11827	0,44272	0,53828	0,23598	-0,23927
9	-0,55393	-0,32968	-0,42213	0,39966	-0,01478	0,04120	-0,44718
10	-0,44997	0,05506	0,10084	0,54982	-0,05668	-0,18809	0,32583
11	-0,28748	-0,22215	0,34346	-0,18588	0,18989	-0,16328	0,42806
12	0,11916	0,60633	-0,35263	-0,29865	0,35351	-0,06837	0,18975
13	-0,12739	0,19497	-0,23516	0,09813	0,04258	-0,04736	-0,17928
14	0,28625	-0,09999	-0,22036	-0,42056	-0,20107	-0,03099	-0,28041
15	0,22143	-0,24792	-0,00318	0,40107	-0,38876	0,10772	-0,04632
16	0,25922	-0,11129	0,17064	0,03348	-0,32109	0,28602	0,50346
17	-0,19624	0,07109	0,07823	0,25402	0,07393	-0,31818	0,16471
18	-0,05959	-0,03128	-0,50833	0,43090	0,47131	0,22756	0,17461
19	0,17456	-0,00350	-0,25818	0,11130	-0,07108	-0,51571	0,22351
20	-0,00833	0,41162	-0,51180	0,23183	-0,19357	0,01488	-0,45597
21	-0,14028	0,62896	0,26346	-0,27892	0,16608	0,19602	0,30636
22	-0,24199	-0,01263	0,30516	-0,01046	-0,03512	0,56129	-0,40959
23	0,09169	0,35411	0,12222	0,13811	0,14178	-0,21365	-0,26872
24	-0,24956	-0,21920	-0,11149	-0,44804	-0,25010	-0,17230	-0,30716
25	-0,24196	-0,41449	-0,35463	0,08657	-0,15130	-0,34826	0,24646
26	0,16448	0,03219	-0,08889	-0,29839	0,11576	-0,24685	0,11071
27	-0,25597	0,10441	-0,19793	0,19232	0,44935	-0,02466	0,05172
28	0,56466	0,12665	-0,40407	0,22312	0,02846	-0,20396	-0,17795
29	-0,02104	0,47003	0,01048	0,06418	-0,20961	0,55954	-0,44897
30	0,09073	0,20696	0,06411	-0,18149	-0,38936	-0,10731	0,43420
31	-0,32780	0,25220	0,27500	0,24039	-0,23741	-0,03461	-0,37575
32	0,24784	0,05424	0,34009	0,03522	-0,61849	0,05145	0,07283
33	0,29520	-0,16857	-0,02861	0,26413	-0,38516	0,12684	-0,46425
34	0,24857	-0,59575	-0,34430	0,25907	0,46193	-0,26427	0,08351
35	-0,12420	0,14841	-0,40392	-0,01734	0,09302	0,67546	0,23927
36	-0,58027	0,37585	-0,37985	0,14409	-0,35006	0,13809	0,30201
37	-0,36261	0,36992	0,32029	0,58460	0,01615	-0,42508	0,01004
38	-0,03764	0,26253	-0,21012	-0,19695	-0,15594	-0,60288	-0,35488
39	-0,08316	-0,21968	0,42620	-0,22358	-0,18109	-0,18222	-0,29365
40	-0,12957	0,22269	0,33374	-0,30345	-0,16665	-0,29091	-0,09987