

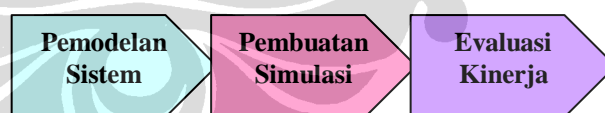
## BAB III

### PEMODELAN SISTEM DAN PERANCANGAN SIMULASI

#### 3.1 Tahapan Implementasi

Pada bagian ini akan dibahas mengenai tahapan-tahapan implementasi untuk sistem yang diajukan di atas. Tahapan-tahapan implementasi tersebut dapat dilihat pada Gambar 3.1 dan berikut akan dijelaskan satu-persatu:

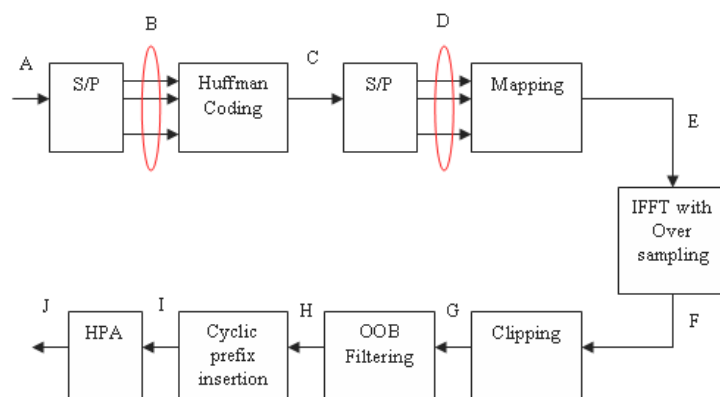
1. **Pemodelan sistem yang diajukan**  
Tahap pertama adalah merancang diagram blok sistem yang meliputi penggabungan *Huffman coding* sebagai *source coding* dan *clipping-filtering* setelah IFFT dalam diagram blok *transmitter OFDM*.
2. **Pembuatan simulasi sistem yang diajukan**  
Simulasi akan dibuat dengan menggunakan *m-file* dengan perangkat lunak MATLAB versi 7.0.4.
3. **Evaluasi kinerja sistem yang diajukan**  
Pada penelitian ini akan dianalisis mengenai *power spectrum density* (psd), *complementary cumulative distribution function* (CCDF), pengaruh rasio kompresi terhadap penurunan PAPR, dan pengaruh IBO terhadap *out-of-band spectrum*.



Gambar 3.1. Tahapan implementasi

#### 3.2 Pemodelan Sistem

Blok diagram sistem yang diajukan dapat dilihat pada Gambar 3.2. Penelitian ini bermaksud untuk mengoptimalkan penurunan PAPR agar pemanfaatan penguat daya sebagai divais *nonlinear* dapat semaksimal mungkin.



Gambar 3.2. Sistem yang diajukan

Algoritma sistem dapat diuraikan sebagai berikut, aliran bit-bit yang masuk (A) dibentuk menjadi  $N$  simbol dengan  $N = 2^n$ , dengan  $n$  adalah jumlah bit tiap simbol (B). Setelah itu simbol-simbol tersebut dikodekan dengan menggunakan *Huffman coding* sehingga menghasilkan aliran bit 0 dan 1 yang baru (C) yang dibentuk lagi menjadi  $N$  simbol (D). Selanjutnya simbol-simbol tersebut akan dipetakan ke domain frekuensi menggunakan modulasi digital, diantaranya QPSK, 16-QAM, atau 32-QAM (E). IFFT kemudian diterapkan sehingga menghasilkan keluaran dalam domain waktu (F) yang kemudian akan dilakukan proses *clipping* (G) dan *filtering* (F). *Filtering* dilakukan dalam domain frekuensi sehingga sebelum *filtering* perlu dilakukan proses FFT terlebih dahulu. Setelah *filtering*, IFFT kembali diterapkan karena proses pengiriman harus dalam domain waktu. Proses *filtering* ini seperti yang ditunjukkan pada Gambar 2.17. Tahap terakhir adalah penambahan CP (I) dan penguatan sinyal dengan menggunakan penguat model Rapp (J).

### 3.3 Pembuatan Simulasi

Seperti yang telah dijelaskan di atas, pembuatan simulasi ini menggunakan perangkat lunak MATLAB dengan menggunakan fasilitas *m-file*. Oleh karena itu, ada beberapa detail dalam simulasi ini yang meliputi pembuatan *Huffman coding*, proses *clipping* dan *filtering*, pengukuran PAPR dengan menggunakan CCDF,

pengukuran psd, dan pengaruh IBO atau SSPA terhadap psd. Berikut akan dijelaskan satu-persatu.

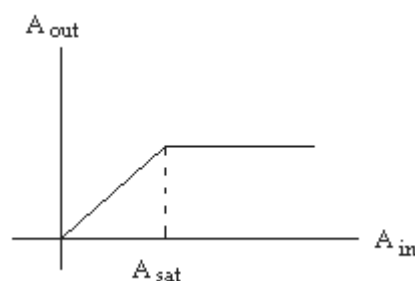
### 3.3.1 Simulasi *Huffman Coding*

*Huffman coding* dilakukan dengan menggunakan fungsi `huffmanenco` dan `huffmandict`. Algoritma untuk *Huffman coding* adalah sebagai berikut:

1. *Generate* simbol sebanyak yang diinginkan
2. Hitung probabilitas kemunculan masing-masing simbol
3. Buat *Huffman coding* dengan membentuk "*Huffman dictionary*" menggunakan perintah `huffmandict(symbols,p)`
4. Kodekan simbol-simbol yang telah di-*generate* dengan menggunakan perintah `huffmanenco(sig,dict)`

### 3.3.2 Simulasi *Clipping* dan *Filtering*

Prinsip dasar *clipping* adalah memotong sinyal yang mempunyai amplitudo lebih besar dari suatu nilai tertentu. Nilai tertentu tersebut disebut *threshold* atau nilai maksimum. Fungsi alih *clipping* dapat dilihat pada Gambar 3.3. Sinyal yang berada dibawah *threshold* akan dilewatkan tanpa dipotong sehingga nilai *input* sama dengan nilai *output*. *Clipping* akan memotong amplitudo saja, sedangkan fasa dari sinyal tersebut tidak berubah, seperti diperlihatkan pada persamaan (2.24).



Gambar 3.3. Fungsi alih *clipping*

Proses selanjutnya adalah menentukan berapa nilai maksimum untuk proses *clipping* tersebut. Pertama, nilai *clipping ratio* (CR) ditentukan terlebih dahulu, setelah itu gunakan persamaan (3.1) berikut untuk menentukan nilai maksimum atau *threshold*.

$$A_{\max} = CR \times \sqrt{N_c} \quad \dots \quad (3.1)$$

$N_c$  = jumlah subcarrier

Algoritma untuk simulasi *clipping* sangat sederhana dan dapat dijelaskan sebagai berikut.

1. Cari amplitudo dan fasa (bentuk polar) untuk sinyal hasil keluaran dari IFFT.

Misalkan sinyal hasil keluaran IFFT adalah

$$y = a + jb \quad \dots \quad (3.2)$$

Maka, bentuk polar sinyal tersebut adalah  $y = |y| \angle \phi$  dengan  $|y|$  dan  $\angle \phi$  dapat dicari dengan menggunakan persamaan:

$$|y| = \sqrt{a^2 + b^2} \quad \dots \quad (3.3)$$

$$\angle \phi = \tan^{-1} \frac{b}{a} \quad \dots \quad (3.4)$$

2. Bandingkan  $|y|$  dengan  $A_{\max}$ .
3. Jika  $|y| > A_{\max}$  maka  $|y| = A_{\max}$  dan  $\angle \phi$  tetap tidak berubah.
4. Jika  $|y| \leq A_{\max}$  maka  $|y|$  dan  $\angle \phi$  tetap tidak berubah.
5. Kembalikan lagi ke bentuk kartesian dengan menggunakan persamaan:

$$a = |y| \cos \phi \quad \dots \quad (3.5)$$

$$b = |y| \sin \phi \quad \dots \quad (3.6)$$

### 3.3.3 Simulasi psd

Simulasi *power spectrum density* dilakukan pada bagian (J) pada blok diagram pada Gambar 3.1. Simulasi psd dilakukan dengan menggunakan perintah `pwelch(x,window,noverlap,nfft,fs)` yang berdasarkan metode *Welch periodogram* [19-20] yang dapat dijelaskan sebagai berikut:

1. Vektor sinyal masukan  $x$  dibagi menjadi  $k$  segmen yang saling *overlap* berdasarkan nilai window dan *noverlap*. Nilai  $k$  dapat dihitung sebagai berikut:

$$k = \frac{m-o}{l-o} \quad \dots \quad (3.7)$$

dengan  $m$  adalah panjang vektor  $x$ ,  $o$  adalah banyaknya sampel yang *overlap* (*noverlap*), dan  $l$  adalah panjang setiap segmen (panjang *window*)

2. *Hamming window* diterapkan pada setiap segmen  $x$
3. Setelah itu dilakukan proses IFFT pada data yang telah di-*window*
4. Periodogram dari setiap segmen yang telah di-*window* dihitung
5. Kemudian dirata-rata untuk mendapatkan spektrum estimasi  $S e^{j\omega}$
6. Estimasi spektrum tersebut diskala untuk menghitung psd  $S e^{j\omega} / F$ , dengan  $F$  adalah frekuensi sampling  $f_s$

### 3.3.4 Simulasi PAPR

PAPR dilihat pada bagian (H) pada Gambar 3.1 dengan menggunakan parameter CCDF seperti yang dijelaskan pada subbab 2.6. PAPR dirumuskan sebagai [21]:

$$PAPR = \frac{\max [x \ t \ x^* \ t]}{E [x \ t \ x^* \ t]} \quad \dots \quad (3.8)$$

dengan  $*$  menunjukkan konjugat.

Dari penjelasan sebelumnya telah diketahui bahwa sinyal OFDM merupakan penjumlahan dari sejumlah sinyal-sinyal sinusoidal yang mempunyai spasi frekuensi  $1/T$  dan setiap sinusoidal membawa informasi  $a_k$ . Secara matematis, sinyal yang dikirim adalah

$$x \ t = \sum_0^{K-1} a_k e^{\frac{j2\pi kt}{T}} \quad \dots \quad (3.9)$$

dengan  $K$  adalah banyaknya *subcarrier*. Dengan mengasumsikan  $a_k = 1$ , maka dapat dihitung:

$$\begin{aligned}
\max [x \ t \ x^* \ t] &= \max \left[ \sum_0^{K-1} a_k e^{\frac{j2\pi kt}{T}} \sum_0^{K-1} a_k^* e^{\frac{-j2\pi kt}{T}} \right] \\
&= \max \left[ a_k a_k^* \sum_0^{K-1} \sum_0^{K-1} e^{\frac{j2\pi kt}{T}} e^{\frac{-j2\pi kt}{T}} \right] \\
&= K^2
\end{aligned}$$

Rata-rata daya sinyal adalah:

$$\begin{aligned}
E [x \ t \ x^* \ t] &= E \left[ \sum_0^{K-1} a_k e^{\frac{j2\pi kt}{T}} \sum_0^{K-1} a_k^* e^{\frac{-j2\pi kt}{T}} \right] \\
&= E \left[ a_k a_k^* \sum_0^{K-1} \sum_0^{K-1} e^{\frac{j2\pi kt}{T}} e^{\frac{-j2\pi kt}{T}} \right] \\
&= K
\end{aligned}$$

Sehingga, PAPR maksimum untuk sistem OFDM adalah

$$PAPR = \frac{K^2}{K} = K \quad \dots \quad (3.10)$$

Sebagai contoh, jika jumlah *subcarrier* adalah 52 maka PAPR maksimum adalah 52 atau sekitar 17 dB.

Jika menggunakan parameter CCDF maka yang dilihat sebenarnya adalah kumulatif dari nilai PAPR dan probabilitasnya. Karena  $CCDF = 1 - CDF$  maka perlu menghitung CDF terlebih dahulu. CDF merupakan kumulatif dari nilai PAPR. Setelah didapatkan kumulatif CDF maka dicari CCDF kemudian hasilnya dibagi dengan jumlah simbol yang dikirimkan untuk mendapatkan probabilitas munculnya nilai PAPR lebih dari nilai PAPR tertentu. Hasil tersebut akan digambarkan dengan grafik CCDF terhadap Probabilitas.

### 3.3.5 Simulasi Penguat Daya Rapp

Penguat daya Rapp mempunyai fungsi alih seperti pada Gambar 3.4. Jika dibandingkan dengan Gambar 3.3, maka terlihat kemiripan. Penguat daya juga berfungsi *clipping* namun daerah transisinya lebih halus dibandingkan dengan *clipping* yang dipaparkan pada subbab 3.3.2. Oleh sebab itu, *clipping* dengan fungsi alih seperti pada Gambar 3.4 disebut “*soft clipping*” dan *clipping* dengan

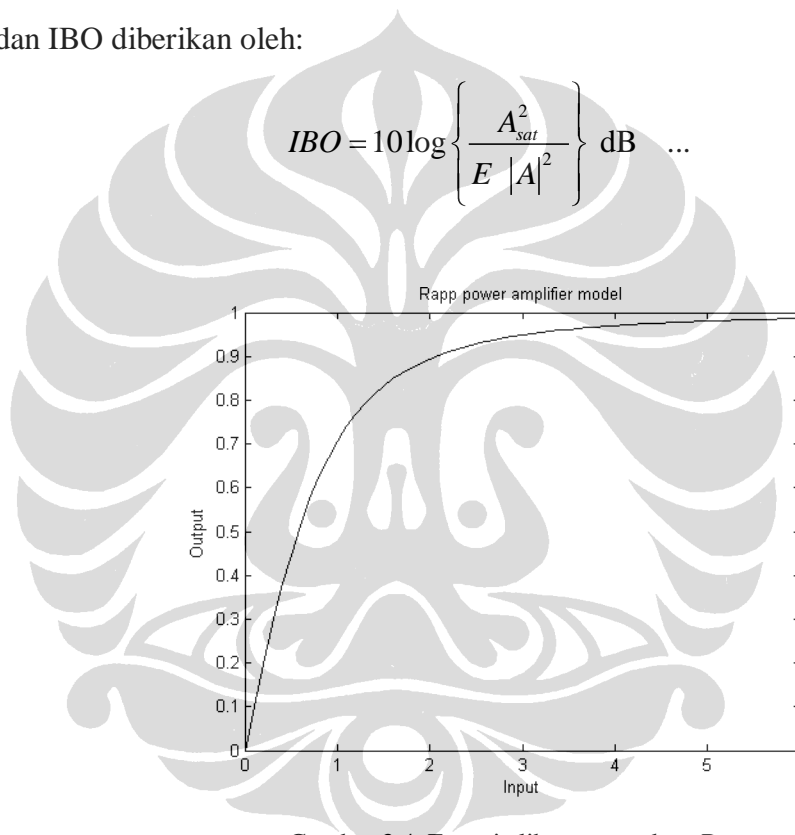
fungsi alih seperti pada Gambar 3.3 disebut “*hard clipping*”. Kehalusan daerah transisi *soft clipping* diatur oleh parameter  $p$ . Nilai parameter  $p$  adalah  $2 \leq p \leq 3$ .

Jika dilihat dari Gambar 3.4 maka penguat daya tersebut mempunyai nilai saturasi sama dengan satu. Nilai saturasi tersebut dapat diatur dengan menggunakan parameter *input back-off* (IBO). Seperti telah diberikan, fungsi AM/AM untuk penguat daya Rapp adalah [15]:

$$G_A = \frac{g_0 A}{\left[1 + \left(\frac{A}{A_{sat}}\right)^{2p}\right]^{1/2p}} \quad \dots \quad (3.11)$$

dan IBO diberikan oleh:

$$IBO = 10 \log \left\{ \frac{A_{sat}^2}{E |A|^2} \right\} \text{ dB} \quad \dots \quad (3.12)$$



Gambar 3.4. Fungsi alih penguat daya Rapp

Setelah nilai IBO ditentukan maka dapat dicari nilai  $A_{sat}$  sebagai berikut.

$$A_{sat} = \sqrt{10^{IBO/10} \times E |A|^2} \quad \dots \quad (3.13)$$

dengan  $E |A|^2$  adalah daya rata-rata.