

III. METODE PENELITIAN

A. Waktu dan Tempat Penelitian

Waktu : Nopember 2009 - Maret 2010

Tempat : Laboratorium Teknik Telekomunikasi Jurusan Teknik Elektro
Universitas Lampung.

B. Metode Penelitian

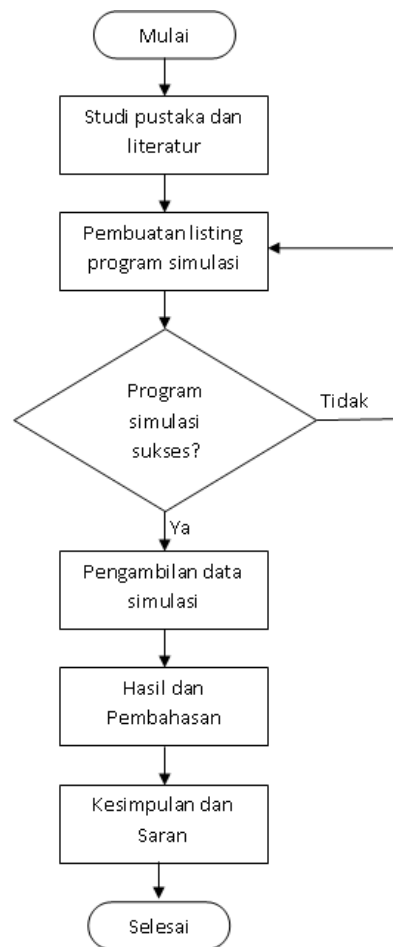
Metode yang dilakukan dalam tugas akhir ini adalah:

1. Studi Pustaka dan Literatur: dilakukan untuk menambah teori dasar dan pendukung dalam menganalisa masalah yang terjadi. Literatur diperoleh dari buku-buku, jurnal dan artikel ilmiah serta bahan-bahan dari internet.
2. Perancangan Perangkat Lunak: dilakukan perancangan perangkat lunak untuk mengukur nilai BER pada simulasi jaringan WiMAX dengan menggunakan MATLAB 7.0.
3. Simulasi dan Pengujian: dilakukan simulasi pentransmisian data pada jaringan WiMAX dengan menggunakan MATLAB 7.0 dan pengujian untuk mengetahui besarnya nilai *Bit Error Rate* dari jaringan WiMAX terhadap modulasi yang digunakan.
4. Hasil dan Pembahasan: dilakukan analisis terhadap hasil simulasi dan pengujian.

C. Alat dan Bahan

Adapun alat dan bahan yang akan digunakan pada penelitian ini yaitu *Personal Computer* (PC) Intel Pentium Dual Core E 2160 dengan RAM 1 GB yang terinstall software Matlab 7.0 sebagai program untuk mensimulasikan penelitian yang dilaksanakan.

D. Diagram Alir Penelitian



Gambar 3. Bagan alir penelitian

E. Performansi Sistem dalam *Bit Error Rate* (BER)

Melihat performansi sistem dengan menggunakan parameter *Bit Error Rate*, tidak boleh mengabaikan peranan subsistem di tingkat *baseband* yaitu berupa pengacak bit, koding *Reed Solomon*, koding konvolusi dan *interleaver*. Keempat subsistem tersebut sangat berpengaruh dalam proses deteksi dan koreksi kesalahan bit. Sebagai gambaran, jika suatu simbol diterima salah maka belum tentu bit yang dikandung oleh simbol tersebut menjadi salah juga. Seberapa besar kemampuan subsistem *baseband* dalam hal deteksi dan koreksi kesalahan bit, tergantung pada bagaimana subsistem *baseband* tersebut dirancang.

F. Perancangan Model dan Simulasi Sistem

Tugas akhir ini akan meneliti pengaruh jenis modulasi dan SNR terhadap nilai *Bit Error Rate* yang dihasilkan pada jaringan telekomunikasi WiMAX untuk masing-masing modulasi. Ada beberapa jenis modulasi, namun yang akan dibahas dalam tugas akhir ini adalah BPSK, QPSK, 16-QAM dan 64-QAM.

Perancangan model simulasi WiMAX sebagai platform dari penelitian mengacu pada beberapa pilihan yang dikeluarkan oleh organisasi dan beberapa peneliti sebelumnya, namun acuan utama adalah standart IEEE 802.16. Standart IEEE 802.16 merupakan standar akses wireless pita lebar yang memiliki laju data yang bervariasi tergantung dari laju keluaran pengkodean kanal, jenis pemetaan dan penambahan *guard interval* yang digunakan.

G. Parameter Model Layer Fisik[15]

Adapun parameter model layer fisik yang digunakan pada simulasi ini yaitu:

- Jumlah *subcarrier* : 256 *subcarrier*
- *Subcarrier* untuk data : 192 *subcarrier*
- *Subcarrier* untuk pilot : 8 *subcarrier*
- Periode *guard interval* : 1/4, 1/8, 1/16 dan 1/32
- Mapping : BPSK, QPSK dan QAM
- *Code rate* : 1/2, 2/3 dan 3/4

H. Algoritma dan *Flowchart* Program Simulasi

Pada laporan tugas akhir ini digunakan software program Matlab 7.0 untuk mensimulasikan pentransmision data pada jaringan WiMAX. Dalam mensimulasikan pentransmision data pada jaringan WiMAX ini, sebelumnya dibuat terlebih dahulu algoritma dan flowchart untuk memberikan gambaran yang jelas mengenai prinsip kerja dari pentransmision data pada jaringan WiMAX agar bisa disimulasikan dengan menggunakan program Matlab 7.0. Algoritma dan flowchart tersebut antara lain :

Algoritma

1. Menentukan jenis modulasi, *code rate*, *transmitter*, *receiver* dan *guard interval* yang akan digunakan dalam proses pentransmision
2. Inisialisai variabel-variabel yang berkenaan dengan besaran-besaran yang digunakan dalam proses pentransmision dan penghitungan BER

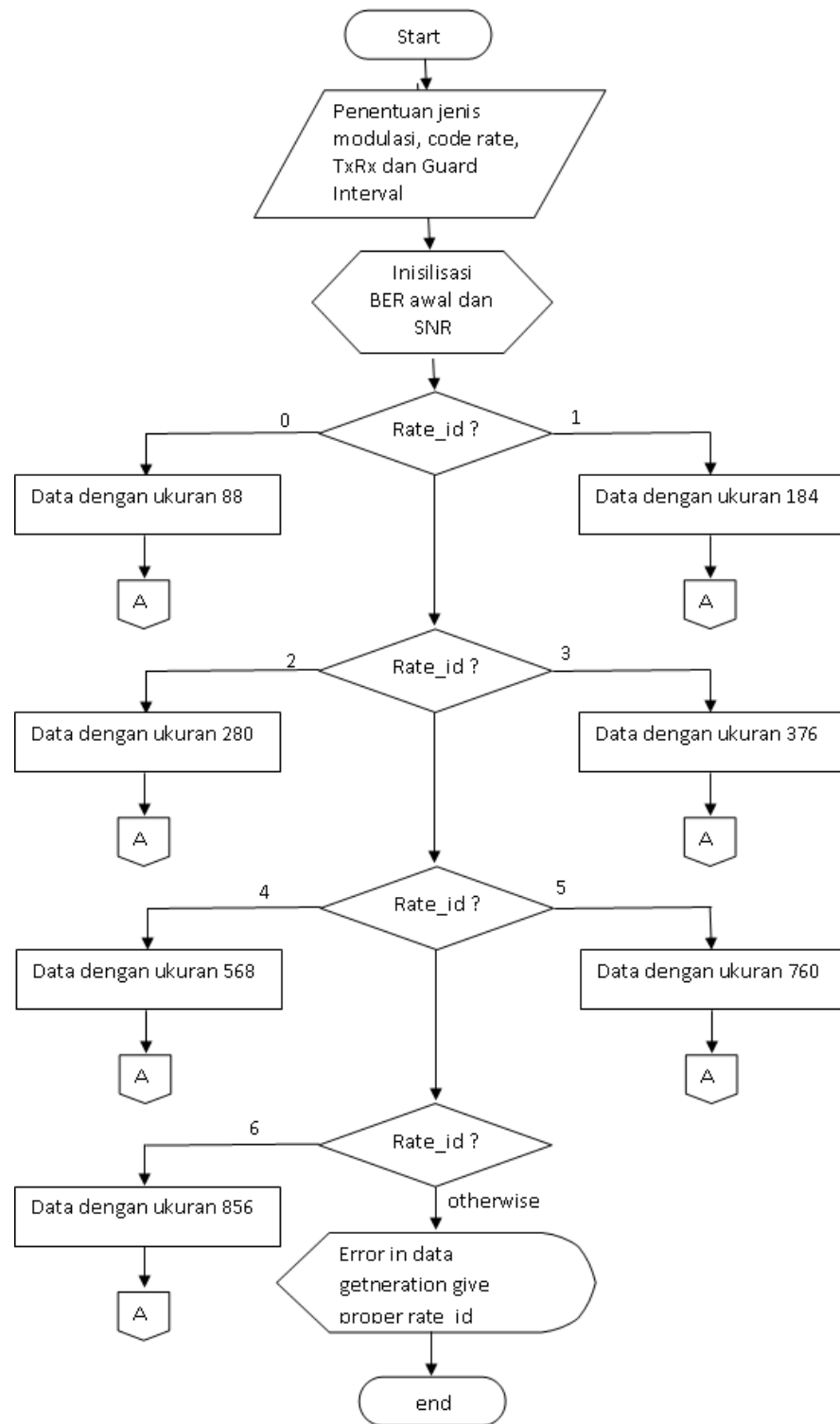
3. Tentukan *rate_id* yang dipakai untuk inialisasi modulasi dan *code rate* yang dipakai
4. Jika *rate_id* yang dimasukan tidak sesuai dengan yang diminta maka ditampilkan keterangan error dan simulasi berakhir
5. Bangkitkan data yang akan ditransmisikan
6. Dilakukan pengacakan data
7. Inialisasi panjang dari *codeword* dan jumlah dari simbol informasi yang digunakan untuk proses RS *encoder* berdasarkan jenis modulasi dan *code rate* yang dipakai
8. Untuk *rate id* 0 tentukan nilai dari TxRx
9. Untuk TxRx 10 dilakukan proses RS *encoder*
10. Untuk TxRx 01 dilakukan penghapusan zero
11. Inialisasi *Bit Parity* (*m*) dari RS *encoder* dan *decoder*
12. Menghitung koreksi kesalahan dari setiap kode RS
13. Tentukan nilai dari TxRx
14. Jika TxRx 01, Bit dirubah kedalam bentuk desimal
15. Dilakukan proses RS *decoder* dan data dirubah kembali menjadi bentuk biner
16. Untuk TxRx 10, Bit dirubah kedalam bentuk desimal dan ditambahkan 0 pada akhir vektor
17. Dilakukan pengkodean RS dengan menggunakan *Galois field* 2^8 dan data dirubah menjadi bentuk desimal
18. Membangun trellis dengan *constraint length* 7, $G_1=171_{\text{oct}}$ dan $G_2=131_{\text{oct}}$
19. Tentukan nilai dari *rate_id*

20. Tentukan nilai TxRx
21. Untuk `rate_id` 0 dan TxRx 01 dilakukan *deconvolution* dengan *Traceback length* 12
22. Jika TxRx 10 dilakukan proses *convolution*
23. Untuk `rate_id` 1 sampai dengan 6 pada nilai TxRx 10 dilakukan proses *convolution*
24. Dilakukan proses *puncturing*
25. Untuk nilai TxRx 01 dilakukan proses membangkitkan *decoded length zero* vektor sesuai dengan masing-masing modulasi dan *code rate*
26. *Deconvolution* dengan *Traceback length* sesuai dengan masing-masing *code rate* yang dipakai
27. Penentuan `rate_id`
28. Jika `rate_id` tidak bernilai 0 sampai dengan 6 maka akan ditampilkan error
29. Hentikan operasi jika `rate_id` tidak bernilai 0 sampai dengan 6
30. Untuk `rate_id` bernilai 0 sampai dengan 6 inisialisasi jumlah bit yang dialokasikan untuk *subcarrier* dan jumlah bit per *carrier* (N_{cbps} dan N_{cpc}) yang digunakan pada saat proses *interleaving* sesuai dengan modulasi yang digunakan
31. Lakukan proses *interleaving*
32. Penentuan nilai dari `rate_id`
33. Untuk `rate_id` bernilai selain 0 sampai dengan 6 maka akan ditampilkan error
34. Untuk `rate_id` 0 sampai dengan 6 dilakukan inisialisasi *constellation* dari masing-masing modulasi

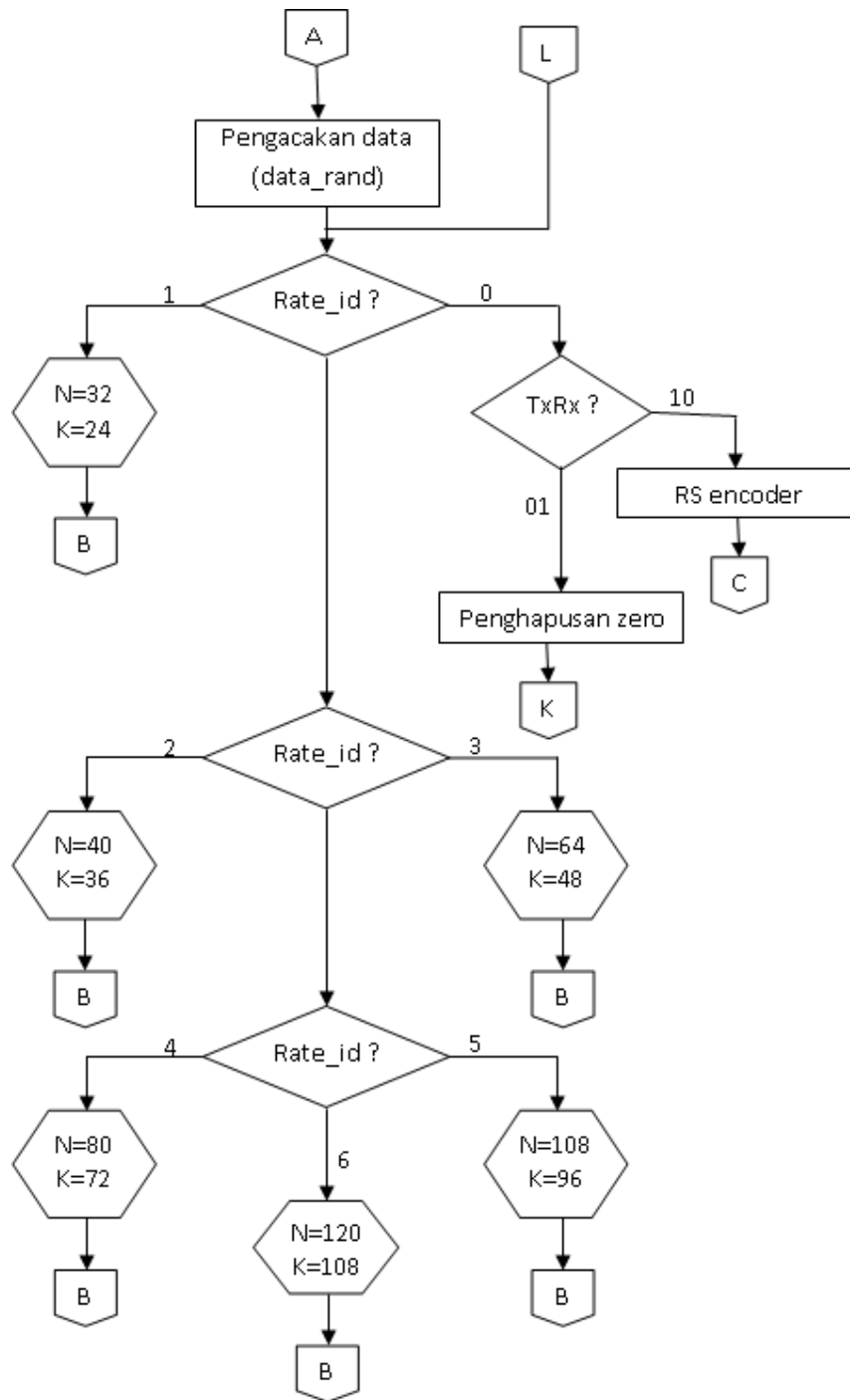
35. Hentikan operasi jika `rate_id` tidak bernilai 0 sampai dengan 6
36. Menghitung jumlah bit per simbol $k=\log_2(M)$ dan data diubah menjadi bentuk desimal
37. Tentukan `rate_id` yang dipakai
38. Untuk `rate_id` selain 0 sampai dengan 6 maka akan ditampilkan error
39. Untuk `rate_id` selain 0 sampai dengan 6 proses akan dihentikan
40. Untuk `rate_id` 0 sampai dengan 2 dilakukan modulasi PSK
41. Untuk `rate_id` 3 sampai dengan 6 dilakukan modulasi QAM
42. Inisialisasi besarnya FFT dan IFFT yang digunakan (`Nfft`)
43. Menentukan TxRx
44. Jika TxRx 10 maka dibangkitkan pilot
45. Menyusun OFDM symbol
46. Menambahkan *cyclic prefix*
47. Menghitung besarnya SNR kanal di *receiver*
48. Mengirim sinyal pada AWGN channel
49. Jika TxRx 01 dilakukan penghapusan *cyclic prefix*
50. Memisahkan pilot dari OFDM symbol
51. Memisahkan pilot
52. Mengekstrak data FFT
53. Tentukan nilai dari `rate_id`
54. Jika `rate_id` selain 0 sampai dengan 6 ditampilkan error dan proses berakhir
55. Untuk `rate_id` 0 sampai dengan 6 inisialisasi *constellation* (M) untuk masing-masing modulasi
56. Inisialisasi jumlah bit per symbol

57. Tentukan `rate_id`
58. Untuk `rate_id` selain 0 sampai dengan 6 ditampilkan error
59. Proses berakhir
60. Untuk `rate_id` 0 sampai dengan 2 dilakukan demodulasi PSK
61. Untuk `rate_id` 3 sampai dengan 6 dilakukan demodulasi QAM
62. Data dirubah kembali menjadi biner
63. Tentukan `rate_id` yang dimasukan
64. Jika `rate_id` yang dimasukan selain 0 sampai dengan 6 maka ditampilkan error
65. Proses akan berakhir
66. Untuk `rate_id` 0 sampai dengan 6 inialisasi variabel-variabel yang digunakan dalam proses *deinterleaving*
67. Dilakukan permutasi pertama
68. Permutasi kedua
69. Penyusunan kembali data
70. Dihasilkan data yang diterima di user
71. Dilakukan perhitungan BER dengan membandingkan data yang diterima dengan data yang dikirim
72. Dihasilkan data berupa grafik perbandingan BER terhadap SNR.

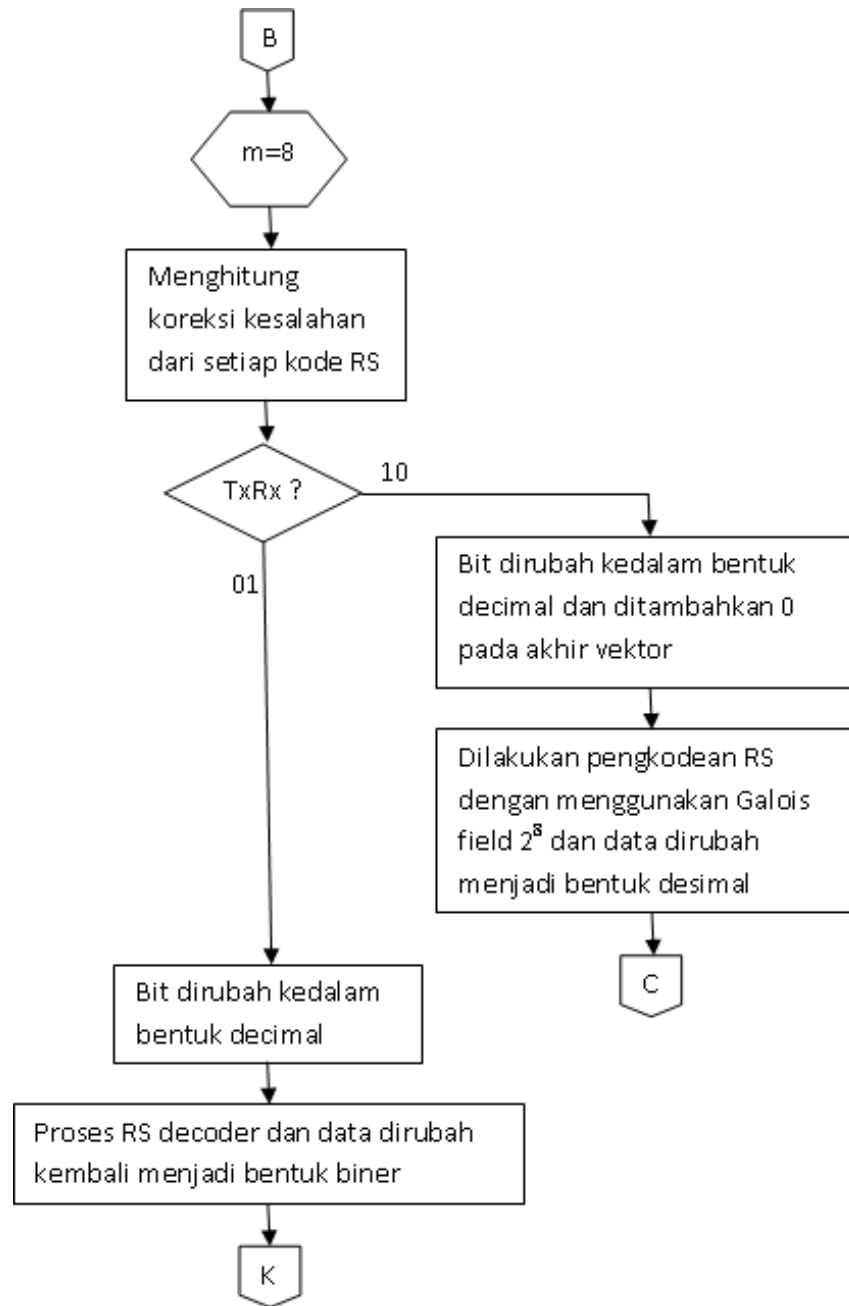
Flowchart



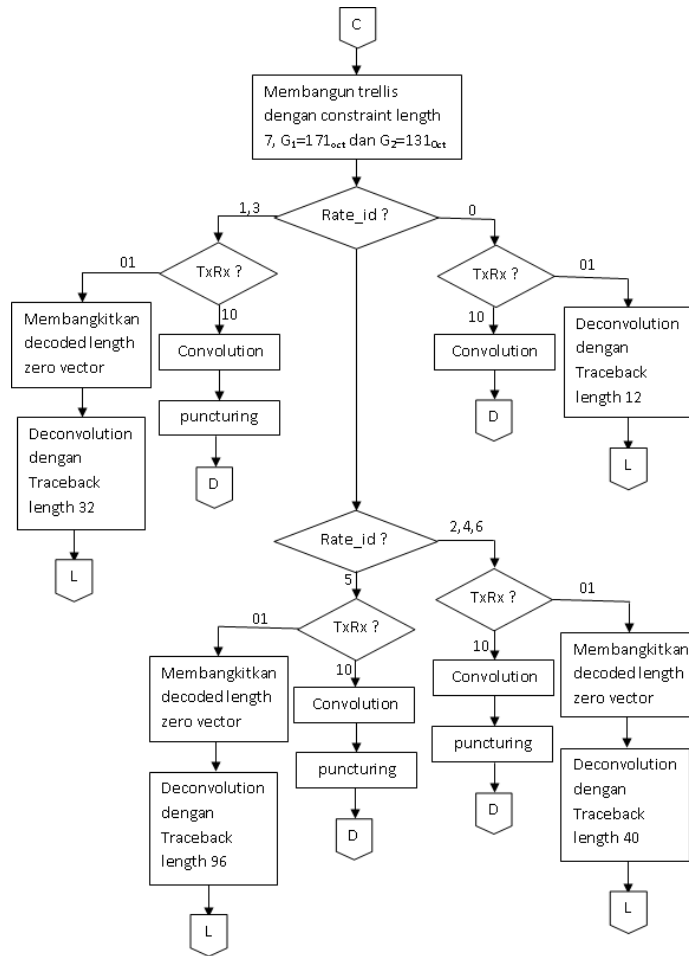
(a)



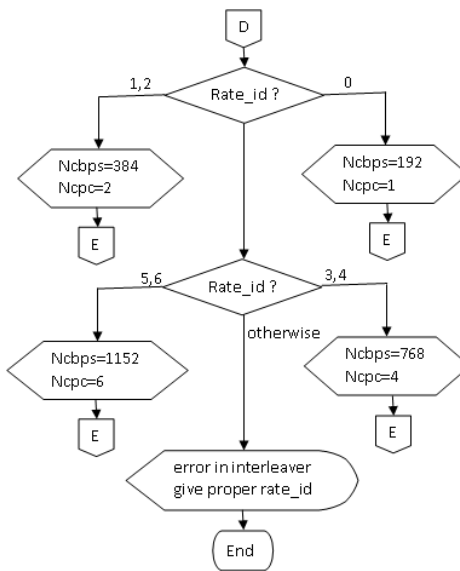
(b)



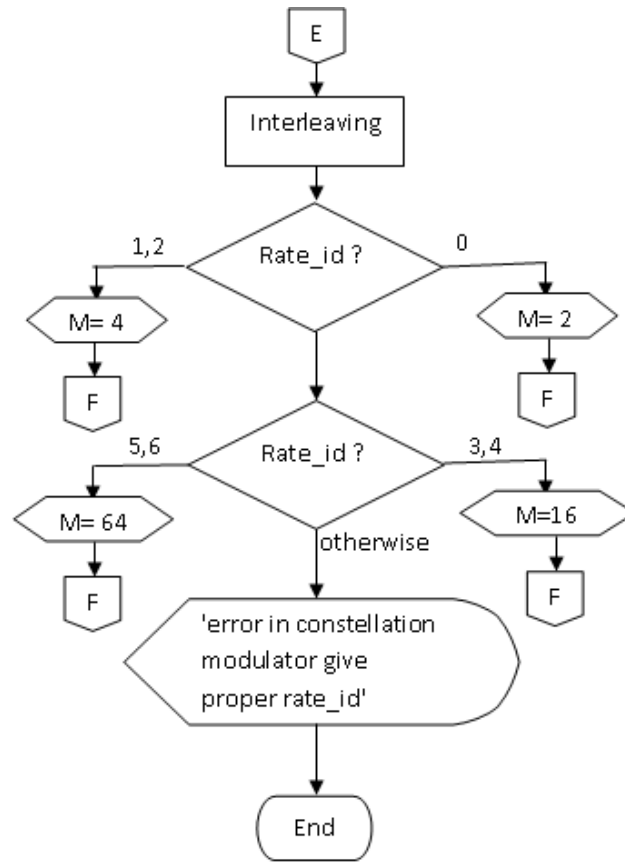
(c)



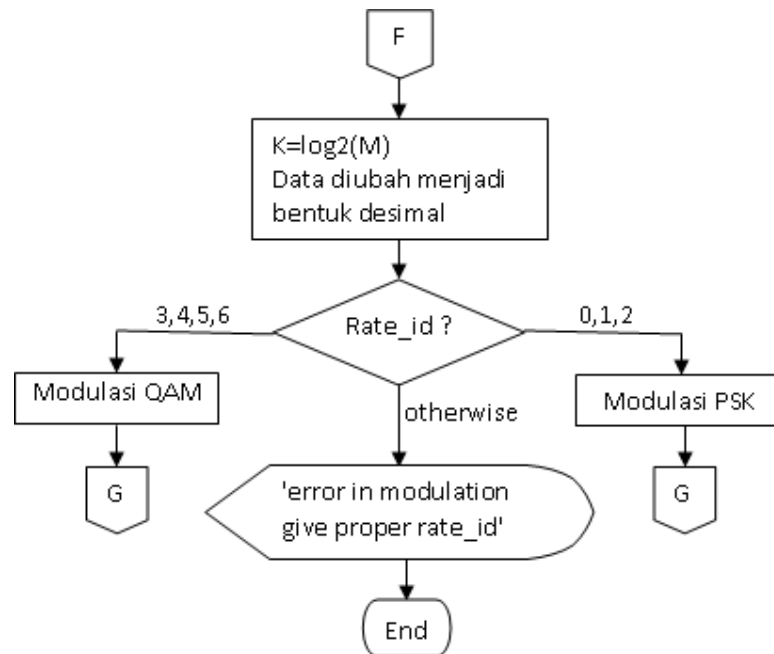
(d)



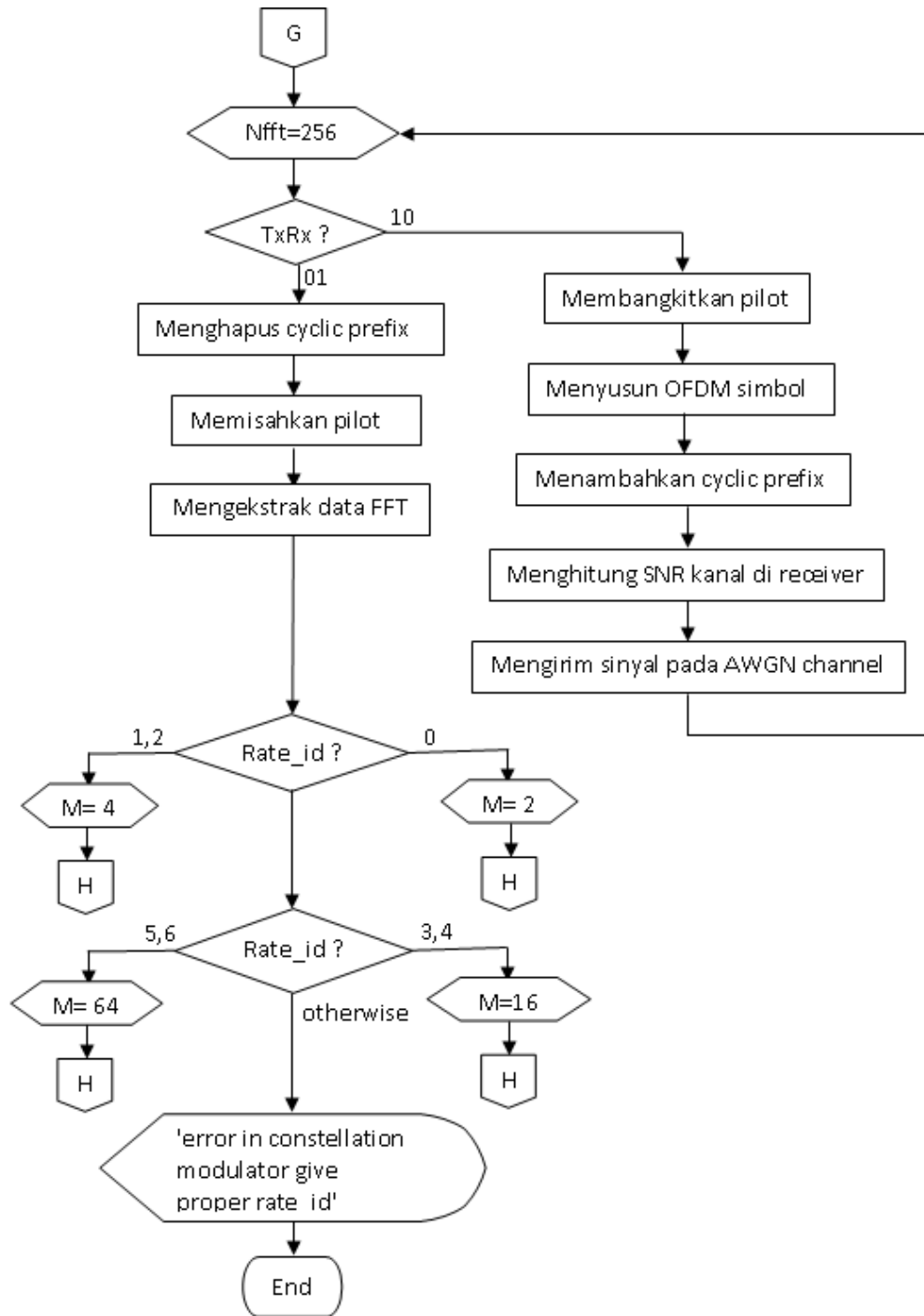
(e)



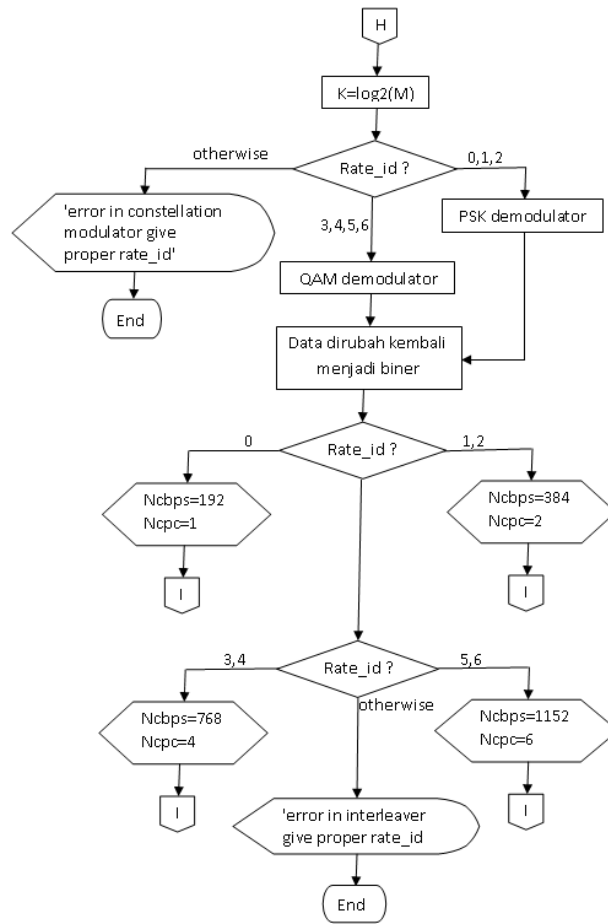
(f)



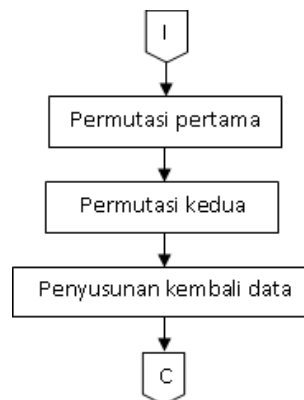
(g)



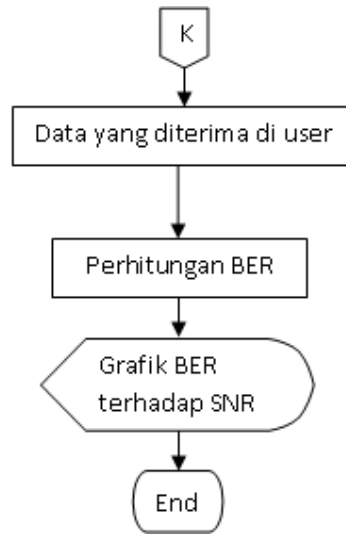
(h)



(i)



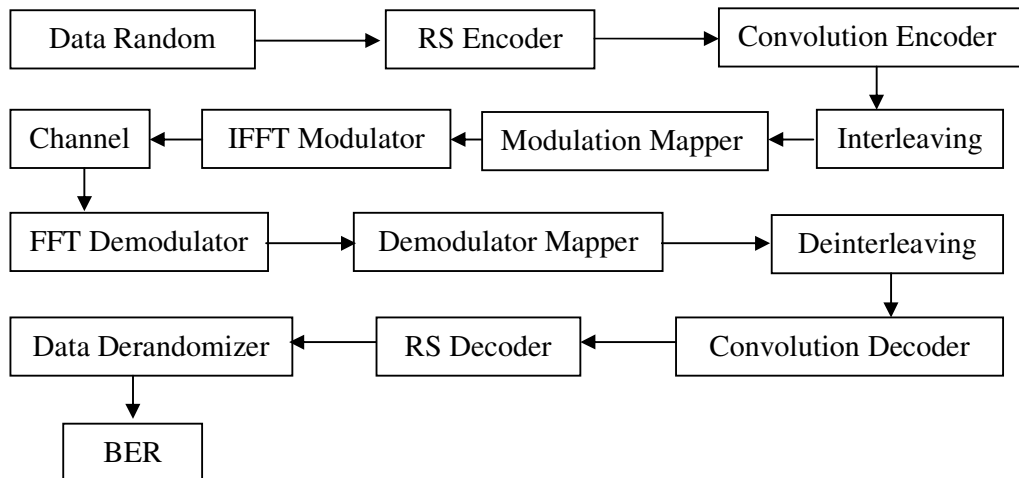
(j)



(k)

Gambar 4. (a),(b),(c),(d),(e),(f),(g),(h),(i),(j),(k) Flowchart program simulasi pentransmisian data pada jaringan WiMAX

I. Bagan Alir Simulasi Jaringan WiMAX



Gambar 5. Bagan alir simulasi WiMAX

Bagan alir di atas menggambarkan simulasi untuk Wireless MAN-OFDM *physical layer* WiMAX. Lapisan PHY ini menggunakan *Orthogonal Frequency Division Multiplexing* (OFDM) dengan 256 *subcarriers*. Setiap simbol OFDM terdiri dari 192 data *subcarriers*, 1 zero DC *subcarrier*, 8 pilot *subcarriers*, dan pada sumber dibangkitkan bit acak. Besarnya bit acak yang dibangkitkan yaitu sebesar *RS code* yang digunakan dikurangi 1. Sumber yang berupa bit acak tersebut kemudian dikodekan dengan menggunakan *RS encoder*, *convolution encoder*, *interleaving* dan *modulation mapper*. Setelah proses pengkodean dilakukan, sinyal dikonversi ke domain waktu dengan menggunakan *Inverse Fast Fourier Transform*, dan dalam proses ini juga ditambahkan *cyclic prefix* (CP) dengan tujuan mencegah interferensi antar simbol. Pada channel ditambahkan *Additif Gaussian White Noise*.

Di *receiver* pada dasarnya bekerja dengan pembalikan operasi yang sama pada *transmitter*. Pertama-tama, CP dihilangkan dan sinyal yang diterima dikonversi menjadi daerah frekuensi menggunakan algoritma FFT. Karena suatu simbol OFDM disusun dari data, pilot-pilot, *zero DC subcarrier*, dan beberapa *guard band*. Jadi dengan demikian diperlukan suatu proses untuk memisahkan semua *subcarriers* ini. Pertama-tama, *guard band* dihilangkan, kemudian dilakukan proses disassembling untuk memperoleh pilot-pilot dan data. Data-data yang diperoleh kemudian digunakan dalam proses *decoder* untuk mendapatkan kembali sinyal yang dikirim oleh sumber. Setelah mendapatkan data asli yang dikirim oleh sumber, baru dapat dihitung besarnya nilai BER yang dihasilkan dengan cara membandingkan data yang diterima dan data yang dikirim.

1. Sumber

Dalam standard WiMAX sebelum informasi ditransmisikan, perlu dilakukan pengacakan terhadap bit informasi tersebut. Proses pengacakan digunakan untuk meminimalkan kemungkinan transmisi modulasi *non-subcarriers*. Proses dari pengacakan dilakukan pada setiap *burst* data *downlink* dan *uplink*, dan di setiap alokasi blok data (subchannels domain frekuensi dan simbol-simbol OFDM pada domain waktu). Jumlah bit yang dihasilkan ditentukan untuk menjadi *frame-based* dan dihitung dari ukuran paket yang diperlukan dalam setiap situasi. Ukuran paket tergantung dari jumlah simbol OFDM yang ditransmisikan dan keseluruhan tingkat pengkodean sistem.

2. RS Encoder [12]

Proses pengkodean terdiri dari Rangkaian *Reed Solomon* (RS) dan *convolutional code* (CC) sebagai FEC skema. Kode *Reed Solomon* (RS) adalah salah teknik pengkodean kanal yang bertujuan untuk melindungi sinyal informasi dari setiap ancaman yang ada pada kanal. Kode *Reed Solomon* merupakan kode blok, yang berarti pesan yang akan ditransmisikan dibagi menjadi blok-blok data yang terpisah. Kode ini disebut juga kode sistematik yang artinya proses encoding tidak merubah simbol-simbol pesan dan simbol proteksi ditambahkan pada tempat yang terpisah pada blok data tersebut. RS disebut juga kode linear (dengan menjumlahkan dua *codeword* akan menghasilkan *codeword* yang lain), dan juga *cyclic* (dengan menggeser sebuah *codeword* secara *cyclic* akan menghasilkan *codeword* lain). RS termasuk dalam keluarga pengkodean

Bose-Choundhuri-Hocquenghem (BCH) non-biner. RS (n,k) artinya dengan masukan sejumlah k simbol informasi maka akan menghasilkan *blok kode* sebanyak n simbol. Dimana, $n=2^m - 1$ dengan m adalah jumlah bit pada setiap simbol. Kemampuan deteksi dan koreksi errornya adalah $t < n-k$.

Proses perhitungan dalam kode RS menggunakan aturan *field* terbatas *Galois Field* $GF(2^m)$ artinya setiap simbol dalam RS merupakan anggota dari $GF(2^m)$ tersebut. Encoder membentuk blok kode

$$C(x) = X^{n-k}m(x) + r(x) \dots\dots\dots (8)$$

dengan $r(x)$ adalah sisa pembagian dari polinom simbol masukan $X^{n-k}m(x)$ oleh polinom generator $g(x)$. $G(x)$ adalah polinom generator berderajat $n-k$ dan merupakan faktor dari $X^n + 1$,

$$g(x)=(x-\alpha^1)(x-\alpha^2)\dots(x-\alpha^{n-k}) \dots\dots\dots (9)$$

Aplikasi encoding ini dapat diterapkan pada encoder RS dengan menggunakan Linear Feedback Shift Register (LFSR) dengan internal feedback yang bersesuaian dengan $g(x)$. Operasi yang digunakan adalah penjumlahan dan perkalian Galois Field (2^m).

Sifat dari kode *Reed Solomon* cocok untuk aplikasi kesalahan yang terjadi di *burst*. *Reed Solomon error correction* adalah *coding* skema yang bekerja dengan terlebih dahulu membangun sebuah polinom dari simbol data yang akan dikirim, dan kemudian mengirimkan versi *oversampled* polinomial bukan simbol-simbol asli sendiri. Kemampuan koreksi kesalahan dari setiap kode RS ditentukan oleh $(n-k)$, dari ukuran redundansi di blok.

Di dalam *blok codes*, suatu urutan dari informasi simbol K disandikan di suatu blok dari N symbol ($N>K$), untuk ditransmisikan pada suatu saluran.

Pada suatu sumber data yang akan dikirim, bit informasi dengan rate B bps setiap T seconds, encoder menerima suatu urutan dari $K=BT$ bits yang merupakan suatu pesan. Setelah informasi bit K dimasukkan ke encoder, dihasilkan suatu urutan dari *code symbol* yang ditransmisikan pada saluran. Dalam pentransmisian data atau *codeword*, N harus lebih besar atau sama dengan K untuk menjamin suatu hubungan yang unik antara masing-masing *codeword* dan masing-masing dari pesan-pesan 2^K . Kode-kode tersebut yang memetakan suatu block dari K informasi symbol ke dalam suatu block N *coded symbol* yang disebut (N,K) *block code*.

Di dalam blok-blok ini, data tersebut lebih lanjut dibagi lagi ke dalam suatu nomor dari suatu symbol, yang secara umum mempunyai ukuran 6 sampai 10 bit. Data asli, adalah suatu blok terdiri dari symbol $N-R$, yang mengalami proses enkoder RS dan R *check symbol*, yang ditambahkan untuk membentuk suatu *code word* dari *length* N .

Pengkodean RS bisa dilakukan di setiap panjangnya pesan dan dapat ditambahkan di setiap nomor dari *check symbol*, kode RS dinyatakan sebagai RS (N, N, R) code. Dengan N adalah jumlah keseluruhan nomor dari simbol per *code word*, R adalah banyaknya *check symbol* per *code word*, dan $N-R$ adalah banyaknya symbol informasi actual per *code word*.

RS *encoding* terdiri dari pembangkitan *check symbol* data asli. Variabel-variabel tersebut dipakai untuk menghasilkan kode RS tertentu yang akan dimasukkan dalam *field polinomial* dan generator polinomial. *Field polinomial* ini digunakan untuk menentukan order *finite field*.

Primitive Polynomial:

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1 \dots\dots\dots (10)$$

Tabel. 2 Skema modulasi dan *coding* WiMAX

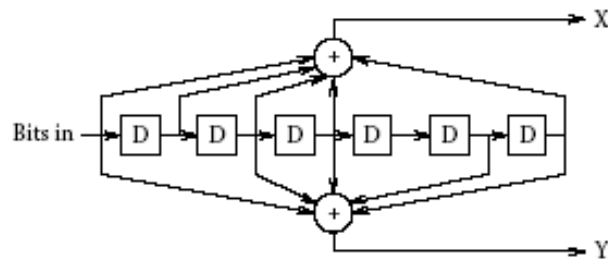
AMC	Modulation	RS Code	CC code rate	Overall code rate
1	2-PAM	(12,12,0)	1/2	1/2
2	4-QAM	(32,24,4)	2/3	1/2
3	4-QAM	(40,36,2)	5/6	3/4
4	16-QAM	(64,48,4)	2/3	1/2
5	16-QAM	(80,72,4)	5/6	3/4
6	64-QAM	(108,96,6)	3/4	2/3
7	64-QAM	(120,108,6)	5/6	3/4

3. Convolutional Encoder

Setelah proses *encoding* RS, data bit dikodekan lebih lanjut oleh *binary convolutional encoder*, yang memiliki *native rate* 1/2 dan *constraint length* 7. Generator polinomial digunakan untuk memperoleh dua kode output bit, dilambangkan X dan Y, yaitu sebagai berikut berikut:

$$G_1 = 171_{\text{OCT}} \text{ untuk X} \dots\dots\dots (11)$$

$$G_2 = 133_{\text{OCT}} \text{ untuk Y} \dots\dots\dots (12)$$



Gambar 6. Convolutional encoder pada WiMAX

Pada blok ini juga terjadi proses *puncturing*. *Puncturing* adalah proses sistematis menghapus bit dari output *stream* pada suatu *low rate encoder*

untuk mengurangi jumlah data yang akan dikirim, sehingga membentuk kode tingkat tinggi. Proses *puncturing* digunakan untuk menciptakan *variable coding rate* yang diperlukan untuk memberikan perlindungan error berbagai level untuk para pengguna sistem.

Rate yang berbeda yang dapat digunakan adalah *rate 1/2*, *rate 2/3*, *rate 3/4*, dan *rate 5/6*. *Puncturing* vektor untuk *rate* ini ditunjukkan dalam Tabel 3.

Tabel. 3 *Puncture vector* untuk berbagai *convolutional coding rate*

Rate	Puncture vector
1/2	[1]
2/3	[1 1 1 0]
3/4	[1 1 0 1 1 0]
5/6	[1 1 0 1 1 0 0 1 1 0]

4. *Interleaver* [13]

Interleaving data umumnya digunakan untuk *scatter error burst* sehingga mengurangi konsentrasi error dengan tujuan meningkatkan efisiensi FEC. *Interleaver* bekerja dengan cara mengacak terjadinya error sehingga mirip dengan *error random*. *Interleaver* dirancang untuk menyebar pola urutan bit-bit yang sebenarnya, agar pengaruh dari *error burst* tidak menyebabkan pola error yang berurutan sebelum masuk decoder. Pada pengirim, deretan bit diatur sedemikian rupa untuk memastikan agar bit-bit yang bersebelahan terpisah sejauh beberapa bit setelah *interleaving*. Proses ini memuat pesan yang akan dikirimkan ke dalam sebuah matrik baris demi baris dan kemudian dibaca kolom per kolom.

WiMAX menggunakan *interleaver* yang bekerja dengan menggabungkan data menggunakan 12 *interleaving level*. Efek dari proses berupa

penyebaran bit simbol yang berbeda, yang digabungkan untuk mendapatkan simbol baru, dengan ukuran yang sama tetapi dengan bit diatur ulang.

Contoh peroses *interleaving*[14]:

Original message:

ARE YOU SURE THAT THEY ARE COMING TO LUNCH WITH US

Interleave matrix:

```
A R E Y O U S U R E
T H A T T H E Y A R
E C O M I N G T O L
U N C H W I T H U S
```

Interleaved message:

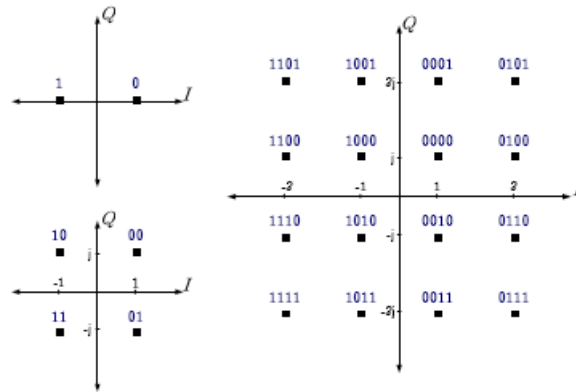
ATEU RHCN EAOC YTMH OTIW UHNI SEGT UYTH RAOU ERLS

5. Modulation Mapper

Setelah sinyal dikodekan, sinyal tersebut memasuki modulasi blok. Semua sistem komunikasi wireless menggunakan skema modulasi untuk memetakan data-data biner menjadi simbol-simbol yang sesuai dengan konstelasi simbol tersebut, yang efektif untuk ditransmisikan melalui saluran komunikasi.

Jadi, bit dipetakan ke *subcarrier* amplitudo dan fase, yang diwakili oleh *complex in-phase* dan *quadrature-phase (IQ) vector*. IQ plot dari suatu skema modulasi berfungsi menunjukkan vektor transmisi untuk semua data dari kombinasi kata. *Gray coding* adalah metode untuk alokasi ini sehingga poin yang berdekatan di konstelasi hanya berbeda satu bit. *Coding* tersebut membantu untuk memperkecil tingkat kesalahan bit.

Modulasi BPSK, 4-QAM, 16-QAM, dan 64-QAM merupakan modulasi yang didukung oleh sistem komunikasi WiMAX. Modulasi yang terakhir, yaitu modulasi 64-QAM, adalah opsional untuk band bebas lisensi. Konstelasi peta untuk modulasi BPSK, 4-QAM, dan 16-QAM ditunjukkan pada Gambar 7.

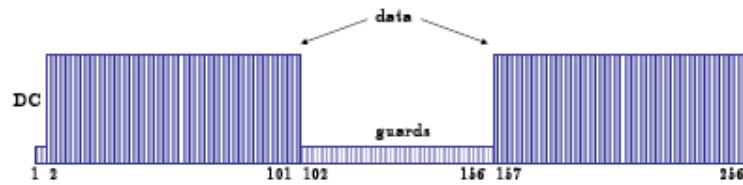


Gambar 7. Konstelasi peta untuk modulasi BPSK, 4-QAM, dan 16-QAM

6. Inverse Fast Fourier Transform

IFFT digunakan untuk menghasilkan sinyal dalam domain waktu, simbol-simbol yang diperoleh setelah modulasi dapat dianggap amplitudo dari range tertentu dari suatu sinyal sinusoid. Selain memastikan *orthogonality* dari OFDM *subcarriers*, IFFT juga mewakili *modulasi subcarriers* secara paralel, sehingga penggunaan berbagai *modulator* dan *demodulator* yang menghabiskan banyak waktu dan sumber daya untuk melakukan operasi ini dapat dihindari.

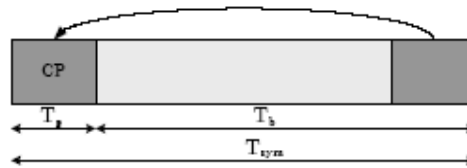
Gambar 8 menunjukkan struktur *subcarrier* yang memasuki IFFT blok setelah melakukan penataan ulang. Seperti yang terlihat pada gambar di bawah ini, *Zero subcarriers* disimpan di tengah-tengah struktur.



Gambar 8. Penyusunan kembali *subcarrier* sebelum melakukan operasi IFFT.

Pada blok IFFT ini juga dibentuk pilot dan *cyclic prefix*. Pilot simbol dapat digunakan untuk melakukan kompensasi frekuensi offset di penerima. Selain itu juga dapat digunakan untuk *fast time-varying channels*. Pilot simbol mengalokasikan *subcarriers* khusus di semua data OFDM simbol. Pilot ini diperoleh dari *Pseudo-Random Binary Sequence* (PRBS) generator yang didasarkan pada polinomial $x^{11} + x^9 + 1$. Semacam pemetaan yang dihasilkan oleh operasi $1 - 2wk$ dan $1 - 2\overline{wk}$, di mana wk adalah urutan PRBS dihasilkan oleh generator, dan \overline{wk} menunjukkan *biner inverse*.

Ketahanan transmisi OFDM terhadap *multipath delay spread* dapat dicapai jika memiliki periode simbol yang panjang dengan tujuan untuk meminimalkan *inter symbol interference*. Pada gambar 9 ditunjukkan satu cara untuk melakukan pengkutipan periode simbol panjang, yang menciptakan *extended guard interval* di mana setiap simbol OFDM didahului oleh perluasan periodik sinyal itu sendiri. *Guard interval* ini, yang sebenarnya merupakan salinan dari bagian terakhir data simbol, yang dikenal sebagai *cyclic prefix* (CP).



Gambar 9. OFDM symbol dengan *cyclic prefix*

Dengan demikian, total panjang simbol

$$T_{\text{sym}} = T_b + T_g, \dots \dots \dots (13)$$

Dimana:

- T_{sym} adalah OFDM simbol waktu,
- T_b adalah simbol waktu yang digunakan, dan
- T_g mewakili waktu CP.

Parameter G mendefinisikan rasio dari panjang CP untuk simbol waktu yang digunakan. Ketika menghilangkan ISI, harus diperhitungkan bahwa CP harus lebih panjang daripada dispersi dari saluran.

7. Channel

Karakteristik propagasi pada kanal transmisi mobile wireless memiliki beberapa gangguan yang sangat merusak. Perusakan ini bisa menyebabkan sinyal yang diterima berbeda dengan sinyal yang dikirim, sehingga dapat menyebabkan terjadinya kesalahan informasi yang diterima.

Additive White Gaussian Noise (AWGN) merupakan gangguan (*noise*) yang diberikan kepada sinyal yang dikirim pada kanal transmisi dengan menggunakan distribusi gaussian. Keadaan *multipath* dalam sistem komunikasi menyebabkan penghamburan dan pemantulan sinyal transmisi menjadi beberapa sinyal lain. Beberapa sinyal lain ini melalui lintasan yang

berbeda-beda. Sehingga kuat sinyal dan waktu tiba di penerima berbeda-beda. Hal ini menyebabkan masalah *multipath fading* dan *delay spread*.

Panjang lintasan dan perlakuan perlambatan gelombang yang berbeda-beda mengakibatkan sinyal-sinyal *multipath* sampai pada penerima dengan waktu tunda yang bervariasi. Sebuah impuls yang dikirimkan oleh pemancar akan diterima oleh penerima bukan lagi sebagai impuls melainkan pulsa dengan lebar penyebaran yang disebut *delay spread*. *Delay spread* ini dapat menimbulkan interferensi antar simbol (ISI), karena setiap simbol akan saling bertumbukan dengan simbol sebelum dan sesudahnya.

8. *Fast Fourier Transform*

Proses FFT perlu dilakukan untuk mengkonversi sinyal kembali kepada domain frekuensi. Dalam proses ini juga dilakukan penghapusan *guard band* atau *cyclic prefix*. Ketika menghapus *subcarriers* agar cocok dengan *guard band*, struktur frekuensi harus diperhitungkan. Meski *zero padding* bertindak sebagai *guard band* yang ditambahkan pada akhir struktur *subcarrier* di *transmitter*, suatu penyusunan kembali dari *subcarriers* harus dilaksanakan ketika melakukan operasi IFFT ini, seperti yang ditunjukkan pada gambar 8. Jadi dengan demikian, *guard band* dipindahkan dari pusat simbol OFDM, dimana dialokasikan setelah proses pengaturan. Dalam proses ini dilakukan juga pemisahan sinyal, dalam domain waktu atau di dalam daerah frekuensi, untuk mendapat data, dan pilot-pilot yang berguna dalam estimasi sinyal.

9. Demodulator Mapper

Demapper berfungsi menyediakan *interface* antara kanal transmisi, fungsi-fungsi perhitungan dan beroperasi pada bentuk gelombang yang diterima pada setiap pemisahan interval simbol transmisi sehingga menghasilkan suatu nomor atau satu set angka-angka yang mewakili suatu perkiraan dari suatu biner yang dipancarkan atau M-ary simbol. Jadi dengan demikian, metoda-metoda *demapping* merupakan ilmu tentang matrik keputusan dengan tujuan bagaimana membuat suatu keputusan tentang bit, "nol" atau "satu", yang dipancarkan.

10. Deinterleaving

Blok ini bekerja sama seperti yang digunakan dalam *interleaver*. Blok *deinterleaving* melakukan proses mengisi matriks dengan simbol-simbol input ke dalam kolom-kolom, dan kemudian membacanya baris demi baris. Parameter yang digunakan dalam kedua blok adalah sama dengan yang yang digunakan dalam proses *interleaving*.

11. Convolution Decoder

Pada blok *convolution decoder* merupakan kebalikan dari proses *puncturing*. Seperti telah dijelaskan sebelumnya, dalam proses *puncturing* terjadi proses penghapusan bit. Penerima tidak tahu nilai dari bit yang dihapus akan tetapi dapat mengetahui posisi bit-bit tersebut dari *puncturing vector*. Angka nol yang dimasukkan dapat dilihat sebagai *erasures* dari

saluran. Mereka tidak memiliki pengaruh pada perhitungan metrik Viterbi decoder.

Algoritma Viterbi mengurangi beban komputasi dengan memanfaatkan struktur khusus kode trellis. Keuntungan lain yang didapat adalah kompleksitas, yang bukan merupakan fungsi dari jumlah simbol yang membentuk urutan *codeword*. Algoritma Viterbi melakukan perkiraan kemungkinan maksimum *decoding*.

12. RS Decoder

Pada RS decoder ini dilakukan operasi yang diperlukan untuk memecahkan kode sinyal, dan mendapatkan pesan asli yang dikirim oleh sumber. Seperti pada semua blok *receiver*, RS decoder membalikkan langkah-langkah yang dilakukan pada blok di *transmitter*. Dengan demikian, RS decoder mengambil *codewords* dengan panjang N , dan setelah *decoding* sinyal pesan itu dikembalikan dengan panjang K , nilai N dan K yang digunakan pada RS decoder ini harus sama seperti yang digunakan di RS encoder.

13. Data Derandomizer

Pada blok ini dihasilkan data yang sesuai dengan data yang dikirimkan pada sumber di transmitter. Yang menggambarkan data yang akan diperoleh oleh penerima pesan.

14. Perhitungan BER

Pada blok ini dihasilkan nilai BER yang diperoleh berdasarkan besarnya SNR dan modulasi yang digunakan. Perhitungan BER ini dilakukan untuk melihat kualitas dari suatu sistem komunikasi. Metoda perhitungan BER pada blok ini dilakukan dengan membandingkan data yang dikirim terhadap data yang diterima, dan dilakukan perhitungan kesalahan bit, akumulasi total kesalahan kemudian dibagi dengan total data bit yang terkirim.

$$BER = \frac{\textit{jumlah_bit_salah}}{\textit{jumlah_bit_diterima}} \dots\dots\dots (14)$$