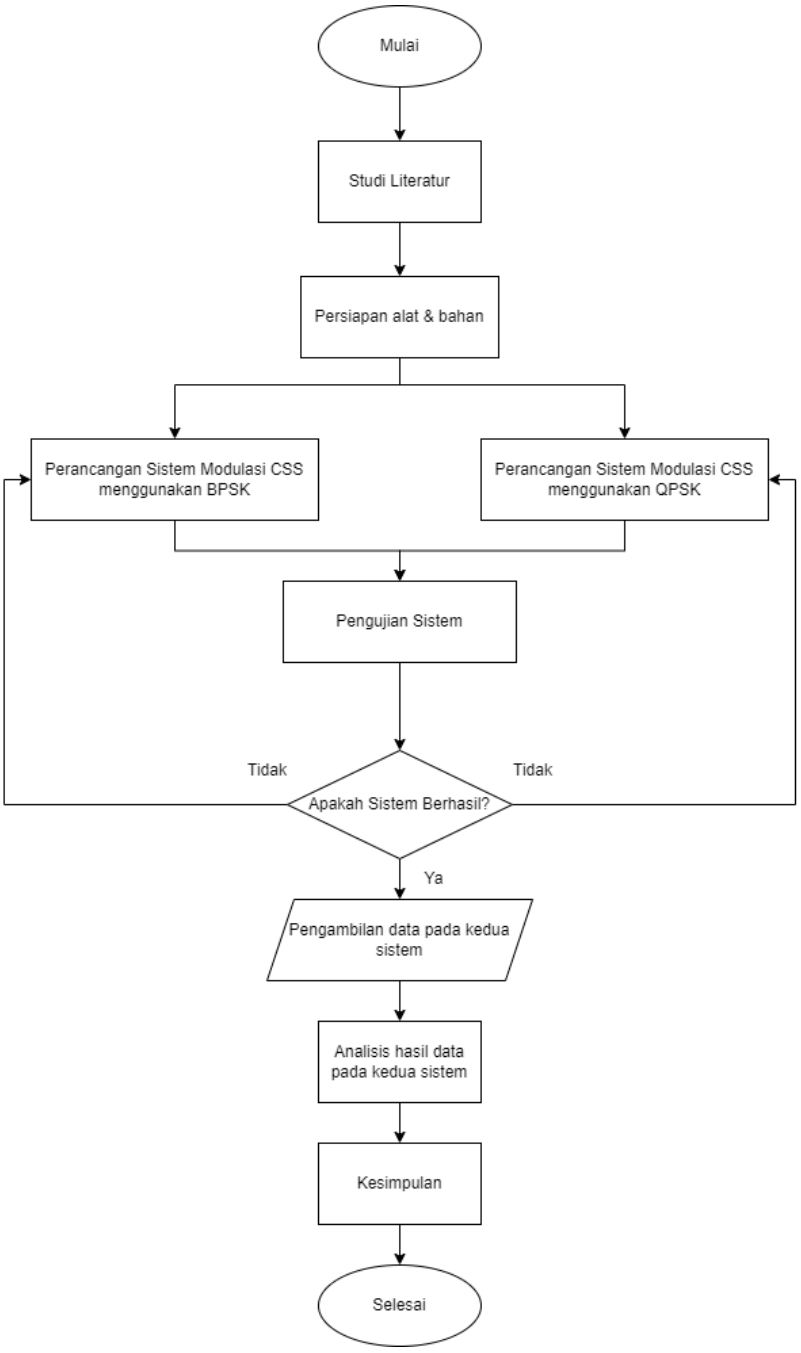
BAB III

METODE PENELITIAN

Pada bab ini menjelaskan tentang alur penelitian, alat dan bahan, perancangan sistem, dan pengujian sistem

3.1 ALUR PENELITIAN

Tahapan-tahapan kerja disusun dalam bentuk *flowchart* seperti pada gambar 3.1 dibawah ini.



Gambar 3.1 Diagram Alur Penelitian

1. Pada tahapan pertama ini penulis mencari berbagai sumber referensi untuk menunjang penelitian dan juga mempelajari tentang bagan dasar sistem komunikasi yang dilanjutkan denan memahami tentang sistem komunikasi LoRa. Pada bagan sistem komunikasi LoRa sedikit berbeda dari sistem komunikasi.
2. Mempersiapkan alat dan bahan yang akan digunakan. Penelitian ini menggunakan sebuah laptop dengan *software* matlab 2021a sebagai *simulator*.
3. Pada tahapan perancangan sistem ini dibagi menjadi 2 yaitu perancangan sistem modulasi CSS menggunakan BPSK dan perancangan sistem modulasi CSS menggunakan QPSK.
4. Tahapan pengujian sistem yang bertujuan untuk menguji sistem yang sudah dirancang sebelumnya. Ketika sistem tidak sesuai dengan parameter yang telah ditetapkan maka akan kembali ke tahap perancangan. Sistem dikatakan berhasil apabila hasil modulasi CSS BPSK dan CSS QPSK telah sesuai dengan teori.
5. Setelah mendapatkan data dari hasil pengujian, tahap selanjutnya adalah menganalisa parameter dari data yang sudah didapatkan, parameter yang akan dianalisa adalah BER dan *data rate*.
6. Tahap terakhir pada penelitian ini adalah membuat kesimpulan dari hasil analisa parameter yang didapatkan.

1.2 ALAT DAN BAHAN

Tabel 3.1 merupakan spesifikasi alat dan bahan yang akan digunakan. Pada penelitian ini menggunakan sebuah laptop yang telah di install *software* Matlab 2021a. *Software* Matlab tersebut berfungsi sebagai *simulator* dalam penelitian ini.

Pada penelitian ini membutuhkan beberapa alat dan bahan untuk menganalisa modulasi CSS pada BPSK dan QPSK, sehingga dari hal tersebut akan dijelaskan beberapa cara kerja dari alat dan bahan yang digunakan pada penelitian ini. Tabel 3.1 menunjukkan alat yang akan digunakan.

Tabel 3.1 Alat dan Bahan

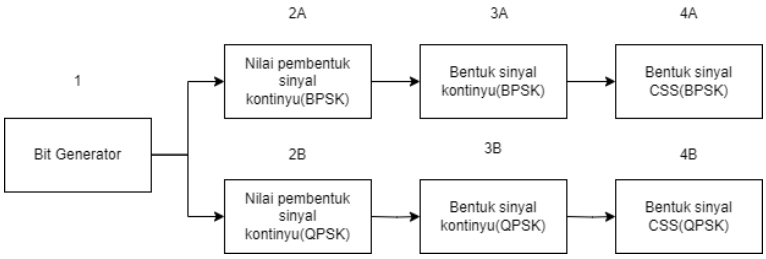
No	Alat dan Bahan	Spesifikasi	Jumlah
1.	Laptop	Prosesor AMD Ryzen 3 3250U, RAM 4 GB	1
2.	*Software* Matlab	Versi 2021a	1

1.3 PERANCANGAN SISTEM

Pada penelitian ini akan dibagi menjadi dua sistem yaitu modulasi CSS dengan BPSK dan modulasi CSS dengan QPSK. Pada kedua sistem tersebut terdapat beberapa parameter simulasi seperti *bit generator* sebagai data yang akan dikirimkan melalui modulasi CSS, *Spreading Factor*(SF) yang merupakan parameter yang mempengaruhi performa CSS, *bandwidth*, dan juga frekuensi yang digunakan. *Bit generator* yang digunakan yaitu 1 *byte*, 2 *byte*, 128 *byte*, dan 256 *byte* sedangkan untuk *bandwidth* dan frekuensi yang digunakan merupakan ketentuan standar LoRa yaitu *bandwidth* 500KHz dengan frekuensi 920MHz. Ketentuan parameter simulasi dapat dilihat pada tabel 3.2 berikut.

Tabel 3.2 Parameter simulasi

Parameter Simulasi	**Keterangan**
Bit Generator	1 *byte*, 2 *byte*, 128 *byte*, dan 256 *byte*
bandwidth	500KHz
frekuensi	920MHz



Gambar 3.2 Alur sistem modulasi CSS

Berdasarkan gambar 3.2 alur sistem modulasi CSS yang digunakan pada penelitian ini berawal dari *bit generator* yang membangkitkan rangkaian bit data yang akan dikirimkan. Selanjutnya setelah pembangkitan data, data tersebut akan melalui proses perhitungan menggunakan rumus – rumus pembentuk sinyal kontinyu pada kedua modulasi CSS. Setelah nilai – nilai pembentuk sinyal kontinyu didapatkan maka dibuktikan dengan hasil *plotting* dari matlab, apakah sesuai dengan bentuk sinyal yang akan diteliti yaitu *upchirp* atau *downchirp*. Setelah bentuk sinyal sesuai dengan tujuan penelitian maka selanjutnya membentuk sinyal CSS pada kedua modulasi menggunakan *spectrogram* pada program matlab.

3.3.1 BIT GENERATOR

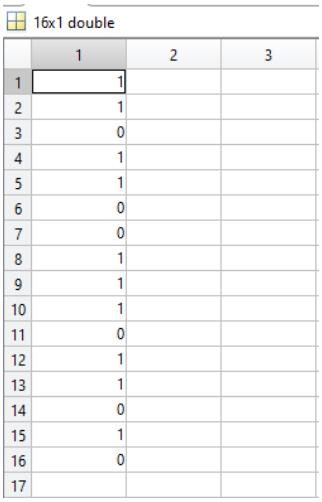
Pada tahap ini, data akan dibangkitkan menggunakan bit generator pada matlab yang berisi data “0” dan “1” atau dapat dikatakan sebagai data biner. Data input disesuaikan dengan banyaknya data yang sudah direncanakan seperti penjelasan diatas. Pada penelitian ini dirancang untuk data input 1 *byte*, 2 *byte*, 128 *byte*, dan 256 *byte*. Proses input dapat dilihat pada gambar.

8x1 double

	1	2	3
1	1		
2	1		
3	0		
4	1		
5	1		
6	0		
7	0		
8	1		
9			
10			
11			
12			
13			
14			
15			
16			
17			

Gambar 3.3 Bit Generator pada 1 *byte*

Pada gambar 3.3 terlihat data yang menjadi input sebesar 1 *byte* yang terdiri dari 8 bit input yaitu “11011001”, data tersebut dimulai dari kolom 1 hingga kolom 8.

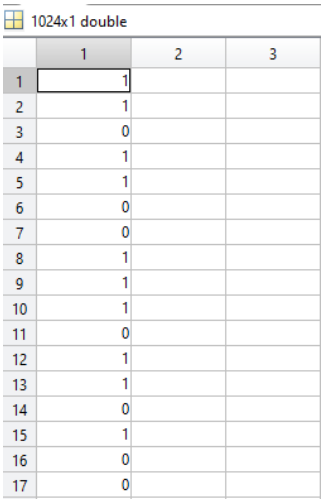


16x1 double

	1	2	3
1	1		
2	1		
3	0		
4	1		
5	1		
6	0		
7	0		
8	1		
9	1		
10	1		
11	0		
12	1		
13	1		
14	0		
15	1		
16	0		
17			

Gambar 3.4 Bit Generator pada 2 *byte*

Pada gambar 3.4 diatas terlihat bit generator yang menghasilkan data 2 *byte* sesuai dengan inputan data sebanyak 16 bit. Begitu pula pada data input bit generator 128 *byte* dan 256 *byte*.



1024x1 double

	1	2	3
1	1		
2	1		
3	0		
4	1		
5	1		
6	0		
7	0		
8	1		
9	1		
10	1		
11	0		
12	1		
13	1		
14	0		
15	1		
16	0		
17	0		

Gambar 3.5 Bit Generator pada 128 *byte*

Pada gambar 3.5 diatas merupakan bit generator dengan inputan 128 atau sebesar 1024 bit.

2048x1 double

	1	2	3
1	1		
2	1		
3	0		
4	1		
5	1		
6	0		
7	0		
8	1		
9	1		
10	1		
11	0		
12	1		
13	1		
14	0		
15	1		
16	0		
17	0		

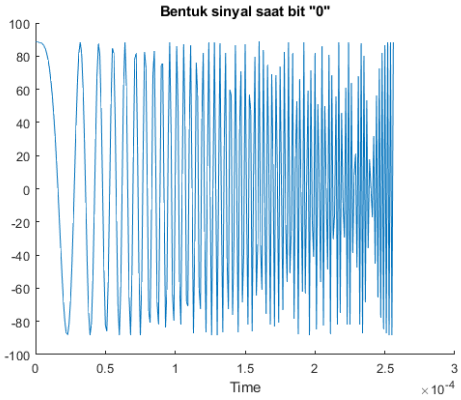
Gambar 3.6 Bit Generator pada 256 *byte*

Pada gambar 3.6 diatas juga merupakan hasil bit generator dengan inputan 256 *byte* atau sebesar 2048 bit.

3.3.2 PEMBENTUKAN SINYAL KONTINYU

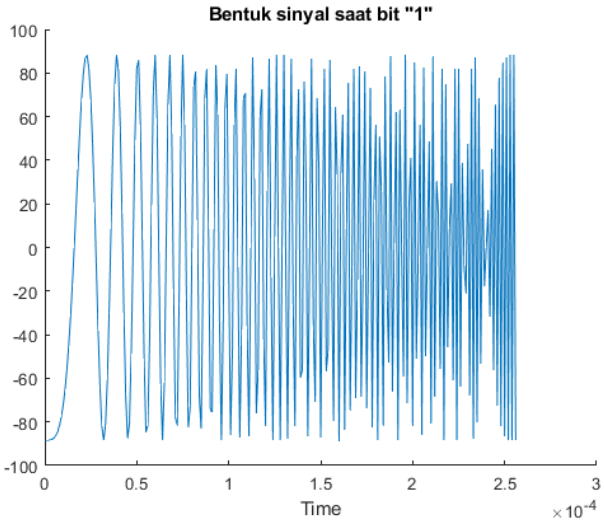
A. BENTUK SINYAL BPSK

Dalam tahap ini, data yang sudah dibangkitkan sebelumnya akan diproses untuk membentuk sinyal kontinyu. Masing – masing modulasi memiliki bentuk sinyal kontinyu yang berbeda dikarenakan pada modulasi BPSK hanya terdapat dua bit yaitu “0” dan “1”, sedangkan pada modulasi QPSK memiliki 4 pasang simbol yang masing – masing pasangan simbol memiliki dua bit dengan kemungkinan pasangan “00”, “01”, “10”, “11”. Hal ini mempengaruhi jumlah sinyal kontinyu yang terbentuk. Bentuk sinyal akan dijelaskan pada gambar 3.3 sampai dengan 3.10.



Gambar 3.7 Bentuk sinyal saat bit “0” pada BPSK

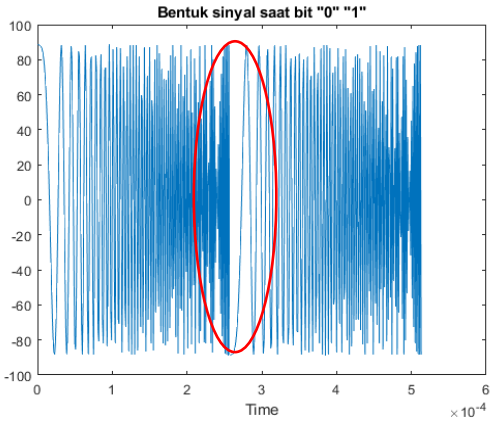
Gambar 3.7 merupakan sinyal yang dibentuk oleh rumus 2.4. Dimana sinyal diatas dimulai dari sudut 90^o karena penggunaan cos pada rumus yang menggeser sudut sebesar 90^o dari sudut awal 0^o.



Gambar 3.8 Bentuk sinyal saat bit “1” pada BPSK

Bentuk sinyal pada gambar 3.8 diperoleh berdasarkan perhitungan dengan menggunakan rumus 2.5. Sinyal tersebut memiliki perbedaan sudut sebesar 180^o dari gambar 3.7 karena penggunaan phi(π) di akhir rumus 2.5. Seperti diketahui nilai phi(π) sebesar 180^o.

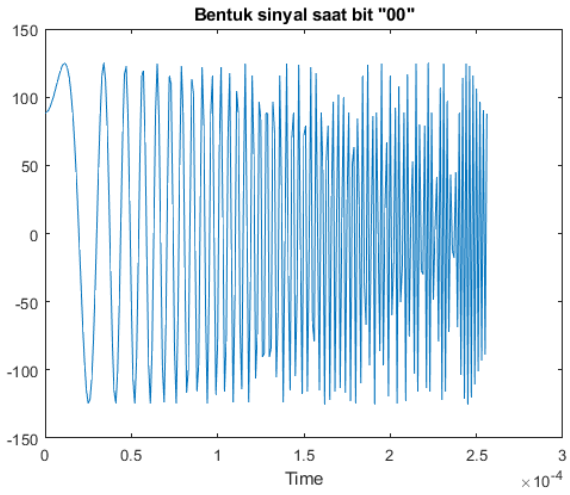
Perbedaan dari kedua sinyal diatas terdapat pada fasa yang dihasilkan karena pada perhitungan bentuk sinyal pada bit “1” ditambahkan dengan pi atau 180^o. Perapatan frekuensi pada sinyal – sinyal diatas merupakan karakteristik pada modulasi CSS yang terbentuk karena perhitungan 2.2 dan 2.3 yang membentuk sinyal *upchirp* yang ditandai dengan variabel fmin yang berarti sinyal dimulai dari frekuensi rendah, apabila variabel fmin tersebut diganti menjadi fmax maka, sinyal yang menghasilkan sinyal *downchirp* yang dimana frekuensi tersebut dimulai dari frekuensi tinggi.



Gambar 3.9 Bentuk sinyal pada saat bit “0” dan “1”

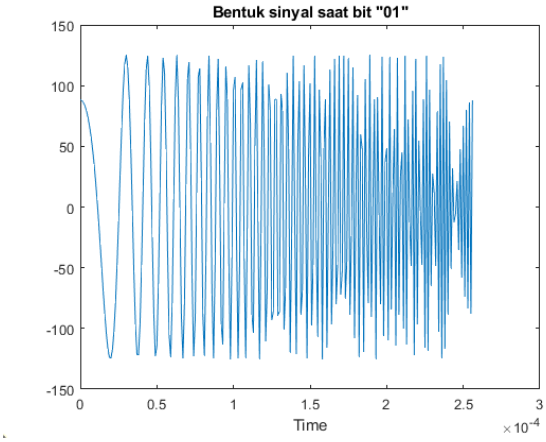
Gambar 3.9 merupakan bentuk sinyal pada saat bit “0” dan “1”, dapat dilihat bahwa terdapat perubahan frekuensi yang signifikan pada lingkaran merah. Hal ini terjadi karena pada modulasi CSS BPSK setiap bit yang dikirimkan akan memiliki frekuensi yang semakin tinggi seiring dengan bertambahnya waktu. Pada gambar diatas masing – masing bit memiliki waktu sekitar 0.0027 detik. Sehingga pada saat mencapai waktu tersebut maka akan terjadi pergantian bit yang dikirimkan dengan perubahan frekuensi rendah dan pergantian fasa sesuai dengan bit selanjutnya. Apabila menggunakan *downchirp* maka, seiring dengan berjalannya waktu frekuensi akan semakin menurun dan apabila terjadi pergantian bit yang dikirimkan maka terdapat lompatan frekuensi rendah ke frekuensi tinggi secara cepat dan perubahan fasa.

B. BENTUK SINYAL QPSK



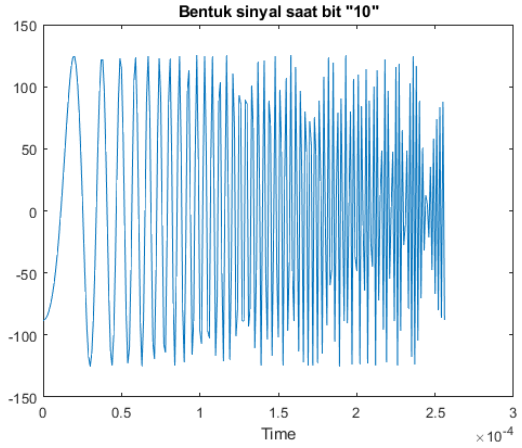
Gambar 3.10 Bentuk Sinyal pada saat pasangan bit “00” pada QPSK

Pada gambar 3.10 sinyal QPSK terbentuk karena adanya penjumlahan antara sinyal I dan Q. Sinyal diatas terbentuk di sudut +45^o karena kedua sinyal I dan Q merupakan sinyal dasar yang belum ditambahkan dengan pergeseran fasa yang apabila dijabarkan menjadi $\mathbf{s_1(t)} = \sqrt{\frac{2}{T_s}}\cos\left(2\pi f_{min}t + \pi\frac{B}{T_s}t^2\right)$ **+** $\mathbf{s_3(t)} = \sqrt{\frac{2}{T_s}}\sin\left(2\pi f_{min}t + \pi\frac{B}{T_s}t^2\right)$. Dimana s1(t) merupakan sinyal I dan s3(t) merupakan sinyal Q.



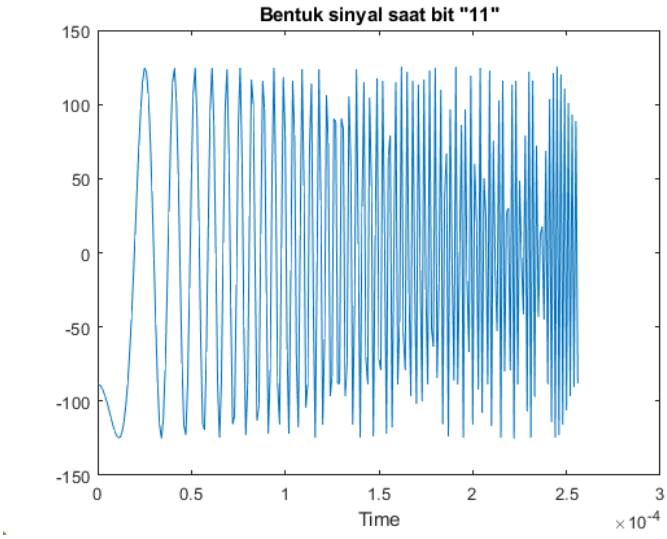
Gambar 3.11 Bentuk Sinyal pada saat pasangan bit "01" pada QPSK

Pada gambar 3.11 sinyal terbentuk karena pada sinyal Q terdapat pergeseran fasa sebesar 180^o dengan nilai I yang tetap dan tidak digeser fasanya sehingga membentuk sudut +135^o pada diagram konstelasi. Apabila dijabarkan menjadi $\mathbf{s_1(t)} = \sqrt{\frac{2}{T_s}}\cos\left(2\pi f_{min}t + \pi\frac{B}{T_s}t^2\right)$ **+** $\mathbf{s_4(t)} = \sqrt{\frac{2}{T_s}}\sin\left(2\pi f_{min}t + \pi\frac{B}{T_s}t^2 + \pi\right)$. Dimana s1(t) merupakan sinyal I dan s4(t) merupakan sinyal Q.



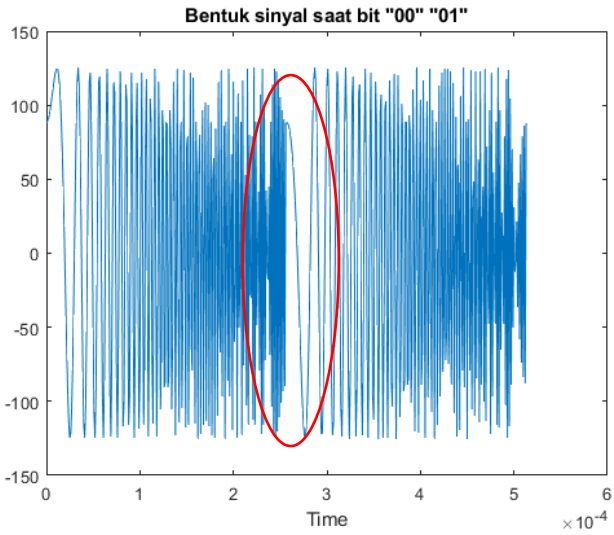
Gambar 3.12 Bentuk sinyal pada saat pasangan bit "10" pada QPSK

Pada gambar 3.12 sinyal terbentuk karena sinyal I terdapat pergeseran fasa sebesar 180^o dengan nilai Q yang tetap dan tidak digeser fasanya sehingga membentuk sudut -45^o pada diagram konstelasi. Apabila dijabarkan menjadi $\mathbf{s_2(t)} = \sqrt{\frac{2}{T_s}}\cos\left(2\pi f_{min}t + \pi\,\frac{B}{T_s}t^2 + \pi\right) + \mathbf{s_3(t)} = \sqrt{\frac{2}{T_s}}\mathbf{sin}\left(\mathbf{2\pi f_{min}t} + \boldsymbol{\pi}\frac{\boldsymbol{B}}{\boldsymbol{T_s}}\boldsymbol{t^2}\right)$. Dimana s2(t) merupakan sinyal I dan s3(t) merupakan sinyal Q.



Gambar 3.13 Bentuk sinyal pada saat pasangan bit "11" pada QPSK

Pada gambar 3.13 sinyal terbentuk karena kedua sinyal I dan Q terdapat pergeseran fasa sebesar 180^o sehingga membentuk sudut -135^o pada diagram konstelasi. Apabila dijabarkan menjadi $\mathbf{s_2(t)} = \sqrt{\frac{2}{T_s}}\cos\left(2\pi f_{min}t + \pi\,\frac{B}{T_s}t^2 + \pi\right) + \mathbf{s_4(t)} = \sqrt{\frac{2}{T_s}}\sin\left(2\pi f_{min}t + \pi\,\frac{B}{T_s}t^2 + \pi\right)$. Dimana s2(t) merupakan sinyal I dan s4(t) merupakan sinyal Q.



Gambar 3.14 Bentuk sinyal saat “00” “01”

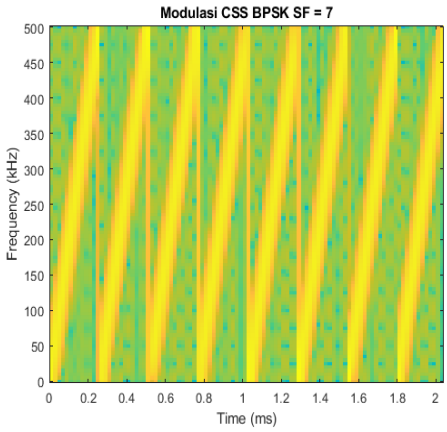
Gambar 3.14 merupakan bentuk sinyal pada saat pengiriman simbol dengan bit “00” dan “01”. Dapat dilihat bahwa terdapat lompatan frekuensi tinggi ke frekuensi rendah dan perbedaan fasa sinyal pada saat pergantian simbol yang akan dikirimkan.

Perbedaan dari BPSK dan QPSK adalah pada kemungkinan bentuk sinyal kontinyu yang dibentuk karena pada QPSK terdapat 4 titik konstelasi yang masing – masing titik berjarak 90°, yaitu +45°, +135°, -45°, dan -135°. Keempat sinyal diatas dibentuk karena perhitungan pada QPSK yang memisahkan antara urutan bit ganjil dan genap yang masing – masing memiliki perhitungan yang berbeda dengan bit genap menggunakan cos sedangkan pada bit ganjil menggunakan sin, sedangkan pada BPSK hanya menggunakan cos saja karena hanya membentuk dua buah sinyal yang memiliki perbedaan fasa sebesar 180°.

3.3.3 PEMBENTUKAN SINYAL MODULASI CSS

A. BENTUK SINYAL CSS

Pada sub bab ini memaparkan tentang sinyal modulasi CSS yang terbentuk oleh proses perhitungan dari bit data yang sudah dibangkitkan di gambar 3.3 sampai gambar 3.6.



Gambar 3.15 Bentuk sinyal modulasi CSS

Bentuk modulasi yang digunakan dalam penelitian ini adalah *upchirp* dapat dilihat dari frekuensi yang memuncak seiring berjalannya waktu sampai ke frekuensi tertinggi lalu langsung melompat ke frekuensi terendah, garis kuning pada gambar 3.15 merepresentasikan 1 buah bit yang dikirimkan pada suatu waktu pada modulasi CSS BPSK, sedangkan pada CSS QPSK garis kuning tersebut merepresentasikan 1 buah simbol yang berisi 2 bit yang dikirimkan pada suatu waktu.

B. BIT *ERROR* RATE

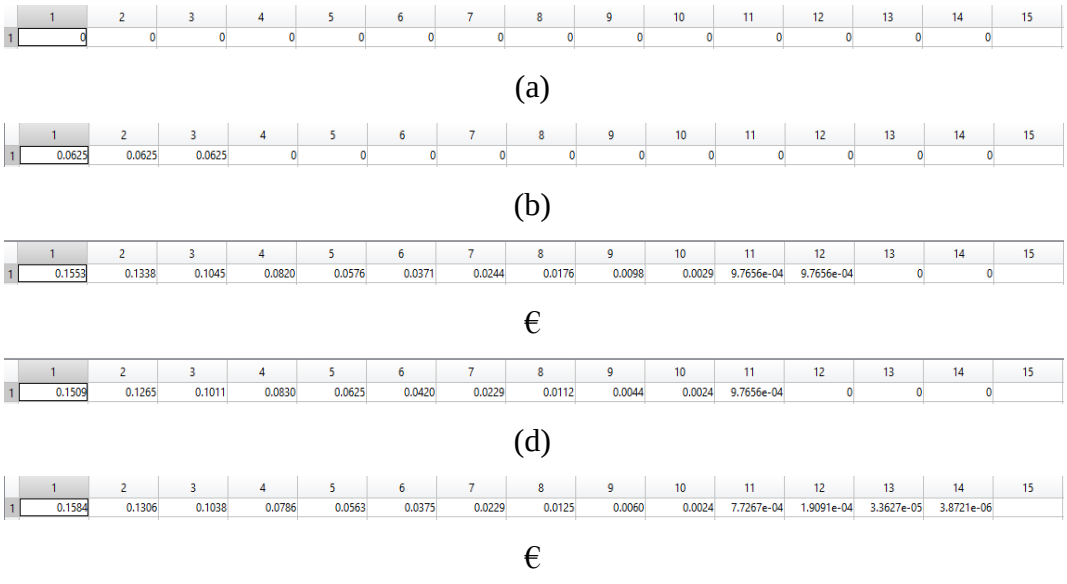
1. BIT *ERROR* RATE CSS BPSK

```
56 -    %BER
57 -    N = length(x2);
58 -    ip = x;
59 -    s = x2;
60 -    n = 1/sqrt(2)*[randn(1,N) + j*randn(1,N)];
61 -    Eb_N0_dB = [-3:10];
62 -  for ii = 1:length(Eb_N0_dB)
63          % Noise addition
64 -        y = s + 10^(-Eb_N0_dB(ii)/20)*n; % additive white gaussian noise
65
66          % receiver - hard decision decoding
67 -        ipHat = real(y)>0;
68
69          % counting the errors
70 -        nErr(ii) = size(find([ip' - ipHat]),2);
71
72 -    end
73
74 -    simBer = nErr/N; % simulated ber
75 -    theoryBer = 0.5*erfc(sqrt(10.^(Eb_N0_dB/10))); % theoretical ber
```

Gambar 3.16 Baris program untuk mensimulasikan BER CSS BPSK

Gambar 3.6 merupakan langkah – langkah untuk menghitung BER. Langkah pertama untuk mensimulasikan BER CSS BPSK adalah membuat variabel untuk membangkitkan noise yang dapat dilihat pada baris ke 60. Selanjutnya menentukan tingkat SNR sinyal pada baris 61. Setelah membuat variabel – variabel tersebut, langkah selanjutnya adalah menggabungkan sinyal BPSK dengan *noise*

yang sudah dibangkitkan, proses tersebut berada pada baris 64. Lalu mengambil nilai riil dari sinyal yang sudah digabungkan dengan *noise* yang terdapat pada baris 67. Untuk menentukan BER secara simulasi, nilai riil akan dibandingkan dengan sinyal asli yang tidak bernoise yang terdapat pada baris 70.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

(a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.0625	0.0625	0.0625	0	0	0	0	0	0	0	0	0	0	0	

(b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.1553	0.1338	0.1045	0.0820	0.0576	0.0371	0.0244	0.0176	0.0098	0.0029	9.7656e-04	9.7656e-04	0	0	

€

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.1509	0.1265	0.1011	0.0830	0.0625	0.0420	0.0229	0.0112	0.0044	0.0024	9.7656e-04	0	0	0	

(d)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.1584	0.1306	0.1038	0.0786	0.0563	0.0375	0.0229	0.0125	0.0060	0.0024	7.7267e-04	1.9091e-04	3.3627e-05	3.8721e-06	

€

Gambar 3.17 Nilai BER CSS BPSK (a)1 *byte*, (b) 2 *byte*, (c) 128 *byte*, (d) 256 *byte*, dan € Nilai BER CSS BPSK secara teori

Gambar 3.17 menunjukkan nilai – nilai pembentuk dari BER CSS BPSK yang dihasilkan oleh baris program sebelumnya, nilai diatas berlaku ke semua skenario percobaan CSS BPSK. Kolom 1 hingga kolom 14 merepresentasikan nilai BER pada SNR -3 hingga 10.

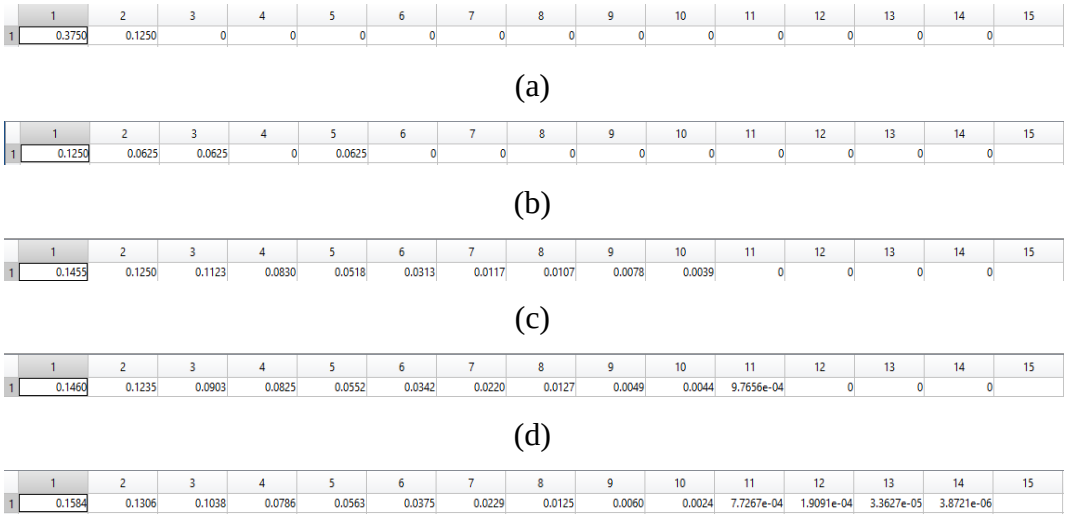
2. BIT *ERROR* RATE CSS QPSK

```
78    %%BER
79 -  L=length(x)/2;
80 -  EbNodB=-3:10;
81 -  EbNo=10.^(EbNodB/10);
82 -  for n=1:length(EbNodB)
83 -      si=even;                        %In-phase symbol generation
84 -      sq=odd;                        %Quadrature symbol generation
85 -      s=sq+j*si;                                    %Adding the two parallel symbol streams
86 -      w =(1/sqrt(2*EbNo(n)))*(randn(1,L)+j*randn(1,L)); %Random noise generation
87 -      r=s+w;                                        %Received signal
88 -      si_=sign(imag(r));                            %In-phase demodulation
89 -      sq_=sign(real(r));                            %Quadrature demodulation
90 -      ber1=(L-sum(si==si_))/L;                      %In-phase BER calculation
91 -      ber2=(L-sum(sq==sq_))/L;                     %Quadrature BER calculation
92 -      ber(n)=mean([ber1 ber2]);                     %Overall BER
93 -  end
94 -  BER_theo = qfunc(sqrt(2*EbNo)); % Theoretical BER
```

Gambar 3.18 Baris program untuk mensimulasikan BER CSS QPSK

Gambar 3.18 merupakan langkah – langkah menghitung BER QPSK. Proses simulasi BER CSS QPSK sedikit berbeda dengan proses BER CSS BPSK karena pada CSS QPSK proses membandingkan antara sinyal ber*noise* dan tidak ber*noise* dilakukan secara terpisah antara sinyal I dan Q. Langkah awal pada BER

CSS QPSK sama seperti BER CSS BPSK, yaitu menentukan batas SNR sinyal yang ditunjukkan pada baris 80. Selanjutnya terdapat proses penggabungan sinyal tak ber*noise* dengan *noise* yang dibangkitkan pada baris 86. Lalu pada baris 88 dan 89 memisahkan antara sinyal I dan Q yang berbentuk nilai riil dan imajiner. Fungsi sign() pada matlab merupakan proses signum yang memiliki proses sebagai berikut: Apabila nilai yang dimasukkan kedalam fungsi signum memiliki nilai lebih dari 0 maka akan mengeluarkan hasil 1 dan apabila nilai yang dimasukkan sebesar kurang dari 0 maka akan mengeluarkan hasil -1. Setelah nilai hasil dari baris 88 dan 89 maka selanjutnya akan dibandingkan seberapa banyak data yang berbeda pada masing – masing sinyal I dan Q bernoise dan tidak bernoise. Lalu hasilnya akan dirata – rata dan menghasilkan nilai BER.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.3750	0.1250	0	0	0	0	0	0	0	0	0	0	0	0	

(a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.1250	0.0625	0.0625	0	0.0625	0	0	0	0	0	0	0	0	0	

(b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.1455	0.1250	0.1123	0.0830	0.0518	0.0313	0.0117	0.0107	0.0078	0.0039	0	0	0	0	

(c)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.1460	0.1235	0.0903	0.0825	0.0552	0.0342	0.0220	0.0127	0.0049	0.0044	9.7656e-04	0	0	0	

(d)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.1584	0.1306	0.1038	0.0786	0.0563	0.0375	0.0229	0.0125	0.0060	0.0024	7.7267e-04	1.9091e-04	3.3627e-05	3.8721e-06	

(e)

Gambar 3.19 Nilai BER CSS QPSK (a)1 *byte*, (b) 2 *byte*, (c) 128 *byte*, (d) 256 *byte*, dan (e) Nilai BER CSS QPSK secara teori

Gambar 3.19 merupakan hasil dari baris program sebelumnya. Nilai – nilai diatas digunakan di semua skenario pengujian CSS QPSK. Kolom 1 hingga kolom 14 merepresentasikan nilai BER pada SNR -3 hingga 10.

3.4 PENGUJIAN SISTEM

Tahap pengujian sistem akan dilakukan dengan menggunakan 3 skenario yang berbeda pada setiap sistem. Ketiga skenario tersebut akan menggunakan parameter simulasi yang sama seperti *bit generator*, *bandwidth*, dan frekuensi. Perbedaan dari ketiga skenario terdapat pada parameter simulasi *Spreading Factor*(SF), dapat dilihat pada tabel 3.3 berikut ini.

Tabel. 3.3 Parameter *Spreading Factor*(SF)

Parameter Simulasi	Skenario 1	Skenario 2	Skenario 3
Spreading Factor(SF)	7	10	12

Kedua sistem ini dikatakan berhasil apabila sesuai dengan landasan teori.

3.4.1 Pengujian pada sistem modulasi BPSK

Pengujian ini meliputi pengujian BER dan data rate pada data yang sudah dibangkitkan menggunakan *bit generator*. Data tersebut akan diproses menggunakan perhitungan untuk membentuk suatu sinyal CSS.

3.4.2 Pengujian pada sistem modulasi QPSK

Pengujian pada sistem modulasi QPSK tidak berbeda jauh dengan pengujian pada sistem modulasi BPSK, hanya saja pada saat pemrosesan data bit yang sudah dibangkitkan melalui proses yang lebih panjang daripada proses CSS menggunakan BPSK. Proses yang dimaksud adalah menentukan urutan bit apakah bit tersebut pada urutan ganjil atau genap, pada masing – masing urutan ganjil genap terdapat proses perhitungan yang berbeda dari CSS menggunakan BPSK.