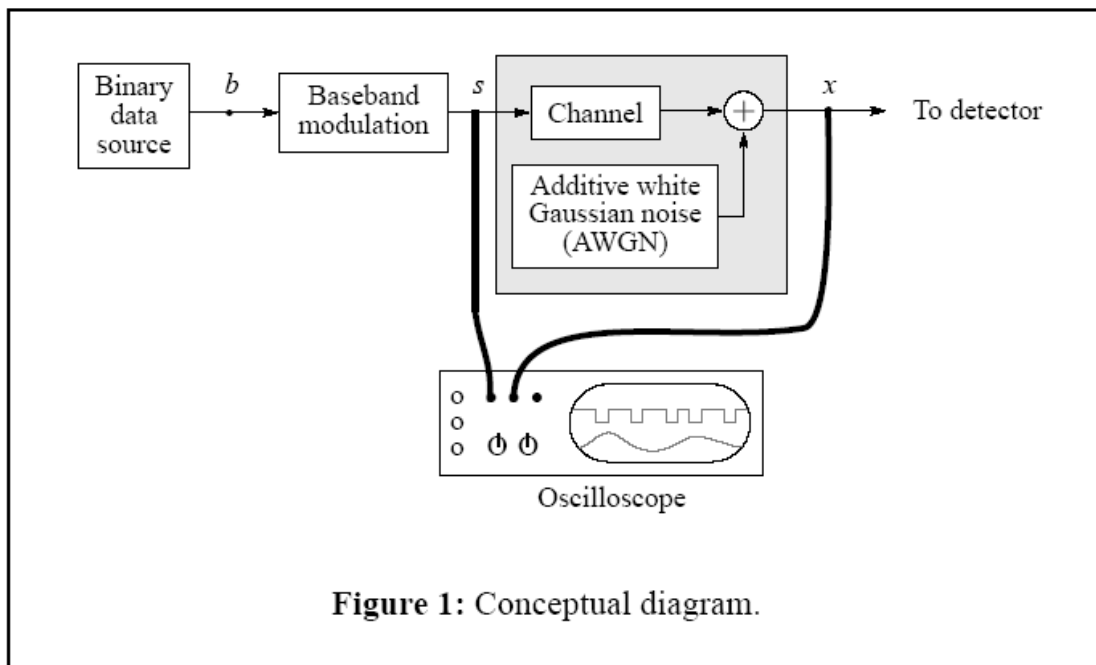


Percobaan 1

Transmisi Digital Baseband



Gambar 1: Diagram konseptual

TUJUAN

Percobaan ini akan mengkaji bagaimana sebuah deretan data digital dapat dikodekan menjadi urutan pulsa-pulsa untuk dikirimkan melalui kanal analog baseband. Secara khusus, anda akan mengamati:

- Berbagai metode pengkodean saluran (*line coding*) yang digunakan untuk modulasi baseband digital pada aplikasi komunikasi data;
- Sifat-sifat spektral dari kode-kode saluran tersebut, mis. kerapatan spektral data (*power spectral density*);
- Dua penyebab utama distorsi sinyal pada kanal komunikasi data, yaitu: derau Gaussian aditif dan pengaruh pemfilteran yang disebabkan oleh bandwidth kanal;
- Pengaruh dari interferensi antar simbol (*intersymbol interference* atau ISI) dan derau aditif yang dapat diamati dengan bantuan pola mata.

REFERENSI

- [1] S. Haykin, *Communication Systems*, 4th ed., John Wiley, 2001. pp. 201-209, 259-264.
[2] B. Sklar, *Digital Communications*, 2nd ed., Prentice Hall, 2001, pp. 78-82.

TUGAS PRA-PERCOBAAN:

Dengan menggunakan referensi di atas atau buku-buku teks lain yang berkaitan dengan topik percobaan ini, selesaikan pertanyaan-pertanyaan berikut.

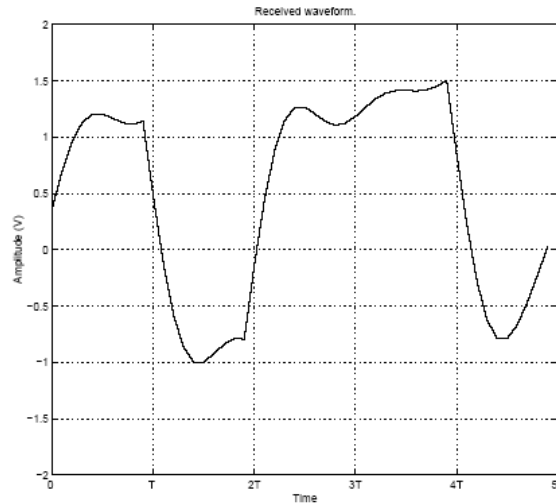
1. Misalkan sebuah pulsa dengan amplitudo satu dan laju data biner $R_b = 1$ kb/dtk, gambarkan grafik gelombang yang merepresentasikan urutan biner berikut $b = \{0,1,0,1,0,0,0,0,1,0,0,1,1\}$ dengan menggunakan kode-kode saluran berikut:
 - a. NRZ (non-return o zero) unipolar (pensinyalan on-off);
 - b. Pensinyalan antipodal biner (NRZ bipolar);
 - c. Alternate Mark Inversion (AMI);
 - d. Manchester (*split-phase*);
 - e. Nyquist (transmisi bipolar dengan bentuk pulsa berupa fungsi sinc).
2. Mari kita amati sifat-sifat spektral dari kode-kode saluran.
 - a. Tentukan secara analitik dan sket fungsi kerapatan spektral daya (*power spectral density* atau PSD) untuk setiap kode saluran di atas. Gunakan laju bit $F_d = 1$ kb/dtk.
 - b. Misalkan f_1 menyatakan nilai nol spektral (*spectral null*) yang pertama (selain $f = 0$, yaitu $f_1 > 0$) pada fungsi PSD. Jika bandwidth transmisi W dari sebuah kode saluran ditentukan oleh f_1 , tentukan W untuk setiap kode saluran pada Pertanyaan 1 sebagai fungsi F_d .
3. Misal $x(t)$ adalah sebuah gelombang yang diamati pada output sebuah kanal komunikasi (lihat Gambar 2). Gelombang tersebut dikodekan dengan menggunakan kode saluran NRZ bipolar. Untuk menampilkan pola mata (*eye diagram*) yang berkaitan, $x(t)$ dihubungkan ke sebuah osiloskop dengan waktu *horizontal sweep* diatur pada T . Sket pola mata yang berkaitan.

LANGKAH-LANGKAH PERCOBAAN:

1. Kode-kode saluran untuk pensinyalan biner pada kanal baseband

Digit-digit biner yang dibangkitkan oleh sebuah sumber data dapat dikodekan secara serial dengan menggunakan berbagai format pensinyalan yang disebut kode saluran (*line code*) untuk pengiriman pada kanal baseband analog. Struktur dan sifat-sifat spektral mereka dapat diamati pada bagian percobaan ini. Anda akan menggunakan perintah khusus Matlab modul dengan penggunaan sebagai berikut:

```
output = modul(<binary_sequence>,<line_code_name>,<Fd>,<Fs>);
```



Gambar 2. Gelombang $x(t)$ pada output sebuah kanal komunikasi

di mana F_d adalah laju data dalam simbol per detik (baud) dan F_s adalah frekuensi sampling simulasi. Perhatikan bahwa dalam percobaan ini kita hanya membatasi pada urutan biner dan skema modulasi biner. Oleh karena itu, F_d berkaitan dengan laju data biner dalam bit per detik. Yang dilakukan perintah di atas adalah mengambil sebuah vektor `binary_sequence` dan memodulasinya sesuai dengan kode saluran yang diberikan. Kode-kode berikut ini dapat digunakan: *'unipolar_nrz'*, *'bipolar_nrz'*, *'ami'*, *'manchester'*, *'miller'*, *'unipolar_nyquist'*, *'bipolar_nyquist'*, *'raised_cosine'*, dan *'duobinary'*.

Beberapa perintah khusus Matlab (`scope`, `modul`, `dll`) membutuhkan parameter F_s dan F_d . Tetapi jika mereka didefinisikan sebagai variabel global sebelum perintah-perintah tersebut diberikan, maka F_d dan F_s tidak perlu diberikan sebagai parameter (baca bagian 1.1 di bawah sebagai contoh).

1.1 Pertama kali, buatlah urutan biner berikut:

```
>>b = [1 0 1 0 0 0 1 1 1 1];
```

Kemudian, bangkitkan sinyal yang terkode yang merepresentasikan `b` dengan menggunakan perintah `modul` dengan laju data $F_d = 1$ kb/dtk, laju sampling $F_s = 10$ kb/dtk dan menggunakan kode saluran NRZ unipolar:

```
>>global Fd Fs;
>>Fs = 10000;
>>Fd = 1000;
>>T = 1/Fd;
>>s = modul(b, 'unipolar_nrz');
```

Tampilkan sinyal yang diperoleh:

```
>>scope(s, 'one-shot', 10*T);
```

Ulangi langkah-langkah ini untuk kode-kode saluran berikut:

- Bipolar NRZ;
- Bipolar RZ;
- Manchester;
- AMI.

Selain itu, ulangi juga untuk kode saluran Nyquist bipolar dengan perbedaan bahwa perintah yang anda berikan perlu sedikit diubah menjadi sbb.:

```
>>scope(s(100:300), 'one-shot', 10*T);
```

Modifikasi ini perlu untuk membuang “ekor” yang panjang dari pulsa Nyquist.

Bandingkan gelombang-gelombang yang anda peroleh dengan jawaban anda untuk pertanyaan no. 1 dari tugas pra-percobaan.

Sekarang kita akan mengamati PSD dari kode-kode saluran. Bangkitkan urutan-urutan biner acak sejumlah 1000 bit:

```
>>b = round(rand(1,1000));
```

Tampilkan fungsi PSD dari urutan ini setelah dikodekan dengan menggunakan NRZ unipolar:

```
>>[Ss,f] = psd(modul(b, 'unipolar_nrz'), 10000, Fs,  
'mean');  
>>plot(f,Ss);
```

Catatan: angka 10000 di atas menyatakan jumlah sampel, dan dalam hal ini diperoleh dari $1000 \text{ bit} \times F_s / F_d$.

Grafik yang diperoleh adalah apa yang akan anda amati dengan menghubungkan *spectrum analyzer* yang beroperasi pada skala linier pada titik s pada Gambar 1.

Untuk menentukan puncak-puncak dan nilai-nilai nol, akan lebih baik jika PSD diamati pada skala logaritmik:

```
>>plot(f, 10*log10(Ss));
```

Dengan mengamati plot dari PSD, temukan lokasi pada sumbu frekuensi dari puncak spektral pertama dan kedua (yang dinyatakan dengan f_{p1} dan f_{p2}) dan nol spektral pertama dan kedua (f_{n1} dan f_{n2}). Ingat bahwa meskipun puncak spektral pertama dapat

terletak pada $f = 0$, kita hanya mencari nol spektral untuk $f > 0$, jadi nol pada $f = 0$ tidak termasuk. Lokasi dari puncak-puncak dan nol-nol anda masukkan pada Tabel 1 di bawah. Ulangi langkah-langkah di atas untuk semua kode-kode saluran yang dipelajari di Bagian 1.1 dan catat hasilnya di Tabel 1.

Catatan: untuk menentukan nol spektral dari kode saluran Nyquist, anda harus menampilkan PSD pada skala linier.

Jika nilai minimum dari bandwidth kanal W yang dibutuhkan untuk kode saluran tertentu ditentukan oleh letak dari nol spektral pertama, tentukan W dari setiap kode saluran.

Tabel 1: Karakteristik f_{p1} spektral dari kode saluran pada laju data 1 kb/dtk.

Kode saluran	f_{p1} (Hz)	f_{p2} (Hz)	f_{n1} (Hz)	f_{n2} (Hz)	W (Hz)
NRZ unipolar					
NRZ bipolar					
RZ bipolar					
Manchester					
AMI					
Nyquist bipolar					

Sekarang marilah kita mengamati hubungan antara PSD dan laju data F_d . Gunakan kode Manchester dan hasilkan PSD untuk nilai F_d yang berbeda-beda:

```
>>[Ss,f] = psd(modul(b,'manchester',<data_rate>),
10000*Fs/<data_rate>,Fs,'mean');
>>plot(f,Ss);
```

di mana `<data_rate>` diisi untuk 250, 500 dan 1000 bit/dtk. Perhatikan perpindahan dari nol dan puncak spektral, dan kaitkan dengan laju data biner.

2. Mensimulasikan kanal baseband dengan derau

Sekarang anda akan mensimulasikan sebuah kanal komunikasi baseband ideal termasuk di dalamnya derau Gaussian putih aditif (*additive white Gaussian noise* atau disingkat AWGN). Kanal tersebut dimodelkan sebagai filter *low-pass* ideal yang outputnya dijumlahkan dengan output dari sebuah pembangkit derau Gaussian putih (lihat Gambar 1). Model kanal ini diimplementasikan dengan fungsi Matlab khusus `bbchannel`:

```
out = bbchannel(<input>,<gain>,<noise power>,<bandwidth>,<Fs>);
```

Sekali lagi perlu anda ingat, jika frekuensi sampling simulasi F_s telah didefinisikan sebelumnya sebagai variabel global, maka tidak perlu dimasukkan sebagai parameter pada perintah `bbchannel` di atas.

- 2.1 Buat urutan biner b terdiri dari 10 bit dan kodekan ke gelombang analog dengan NRZ bipolar.

```
>>b = [0 1 0 0 1 0 0 1 1 0];  
>>s = modul(b, 'bipolar_nrz');
```

Dari pengamatan anda pada Bagian 1, tentukan bandwidth transmisi W dari s :

$W = \dots [\text{Hz}]$

- 2.2 Kirimkan gelombang s pada kanal baseband dengan gain satu dan derau Gaussian putih aditif (AWGN) di daya derau adalah $0.01 W$ dan bandwidth kanal adalah 4.9 kHz. Tampilkan input dan output kanal:

```
>>x = bbchannel(s,1,0.01,4900);  
>>subplot(211), scope(s,'one-shot',0.01);  
>>subplot(212), scope(x,'one-shot',0.01);
```

Apakah informasi yang dikirimkan masih dipertahankan keberadaannya oleh kanal ini? Dengan kata lain, apakah anda bisa memperkirakan secara visual urutan yang dikirimkan dengan hanya mengetahui outputnya saja?

YA atau TIDAK

- 2.3 Untuk mengamati pengaruh derau pada sinyal yang dikirimkan, anda akan memperbesar sedikit demi sedikit daya derau pada kanal sambil tetap menjaga bandwidth tetap pada 4.9 kHz.

```
>>clf; subplot(321), scope(s,'one-shot',0.01);  
>>title('Sinyal asli');  
>>subplot(32<sub_win>),  
scope(bbchannel(s,1,<noise_power>,4900),'one-shot',0.01);  
>>title('daya derau = <noise_power> W');
```

Untuk parameter `noise_power` di atas anda isikan berganti-ganti $\{0.1, 0.5, 1, 2, 5\}$ dan akan membuat grafik masing-masing sinyal pada window yang berbeda-beda. Oleh karena itu `<sub_win>` anda isikan nilai-nilai berikut $\{2, 3, 4, 5, 6\}$.

CETAK grafik yang anda peroleh.

Pada level daya derau berapakah output kanal tidak bisa dibedakan terhadap derau?

$\sigma_n^2 = \dots [W]$

Kita juga bisa mengamati pengaruh dari peningkatan daya derau dengan membuat grafik PSD dari output kanal. Pertama kali, buatlah plot dari PSD sinyal termodulasi s:

```
>>b = round(rand(1,1000));
>>s = modul(b,'bipolar_nrz');
>>[Ss,f] = psd(s,length(s),Fs,'mean');
>>clf, subplot(221), plot(f,Ss), a = axis;
>>title('PSD dari sinyal asli');
```

Setelah itu, buatlah sejumlah PSD dari output kanal yang berbeda sesuai dengan level daya derau {0.01, 1, 5} dan buat grafiknya pada *subwindow* yang berbeda. Variabel <sub_win> mempunyai harga-harga berikut: {2, 3, 4}.

```
>>[Sx,f] = psd(bbchannel(s,1,<noise_power>,4900),
length(s),Fs,'mean');
>>subplot(22<sub_win>), plot(f,Sx); axis(a);
>>title('PSD dari sinyal terima; daya derau =
<noise_power> W');
```

CETAK grafik yang anda peroleh.

- 2.4 Sekarang kita akan melihat pengaruh dari bandwidth kanal terhadap sinyal yang dikirimkan. Kanal yang kita pelajari pada percobaan ini adalah filter *low-pass* ideal dan ia membangkitkan distorsi dengan memotong bagian atas dari spektrum sinyal. Untuk mempelajari pengaruh ini, atur daya derau ke nol dan bangkitkan gelombang output dari kanal untuk bandwidth {4900, 3000, 2000, 1000, 500} Hz:

```
>>clf;
>>b = [0 1 0 0 1 0 0 1 1 0];
>>s = modul(b, 'bipolar_nrz');
>>subplot(611), scope(s,'one-shot', 10*T);
>>subplot(612), scope(bbchannel(s,1,0,4900),'one-
shot',10*T);
>>subplot(613), scope(bbchannel(s,1,0,3000),'one-
shot',10*T);
```

dan seterusnya untuk nilai-nilai bandwidth yang lain ...

Sekarang anda mempunyai 6 plot dengan grafik paling atas adalah gelombang asli yang dikirim dan grafik-grafik berikutnya menunjukkan tingkat distorsi yang semakin besar.

CETAK grafik yang anda peroleh.

3. Pola mata

Pengaruh distorsi yang diakibatkan oleh kanal dan derau dapat diamati secara visual dengan melihat 'pola mata' (*eye pattern* atau juga disebut diagram mata) dari gelombang output. Pola mata dapat diamati dengan sebuah osiloskop di mana beberapa sweep ditampilkan bertumpukan satu sama lain. Masing-masing sweep dipicu oleh sinyal clock yang disinkronkan dengan laju data yang dapat diperoleh dari luar atau diturunkan dari output kanal itu sendiri. Karena terdapat banyak sweep yang ditampilkan, pola mata adalah alat bantu visual yang baik untuk mengamati perilaku rata-rata dan kondisi terjelek dari gelombang output. Secara umum bentuknya mendekati lingkaran atau kadang-kadang seperti pembukaan mata antara setiap *trace*. Mata yang menutup mengindikasikan kanal yang terganggu kesalahan karena hal itu menunjukkan tidak ada margin kesalahan dalam mendeteksi simbol yang terkirim. Demikian juga, jika mata yang diperoleh terbuka lebar dan pola tersebut diamati untuk deretan simbol yang cukup panjang, hal ini mengindikasikan probabilitas kesalahan yang sangat kecil.

Pembangkitan pola mata

Mulailah dengan membuat urutan biner pendek dan bangkitkan pola mata dengan perintah `scope`:

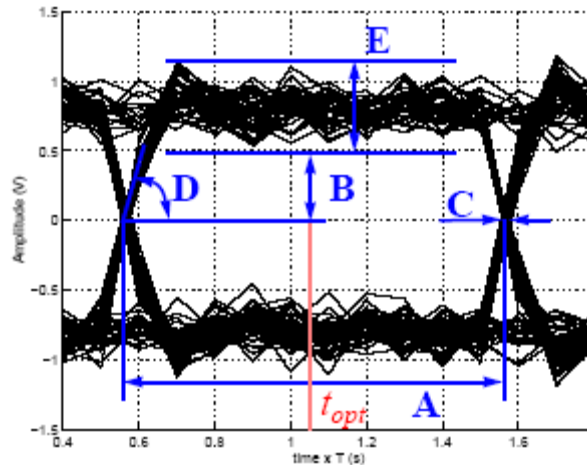
```
>>b = [1 0 0 1 0 1 1 0 0 1];  
>>s = modul(b,'bipolar_nrz');  
>>clf;  
>>subplot(221), scope(s,'one-shot',10*T);  
>>subplot(223), scope(s,'continuous',2*T);
```

Pola mata untuk gelombang *s* merepresentasikan kondisi ideal, yaitu gelombang yang tidak terdistorsi. Sekarang untuk membangkitkan pola mata dari urutan tersebut setelah melewati kanal yang terbatas bandwidth-nya (*band-limited*);

```
>>x = bbchannel(s,1,0,4000);  
>>subplot(222), scope(x,'one-shot',10*T);  
>>subplot(224), scope(x,'continuous',-2*T);
```

Nilai waktu *sweep* yang negatif akan mengakibatkan pola mata ditampilkan setiap *trace* sekali. Anda harus menekan tombol RETURN setiap saat agar *trace* berikutnya ditampilkan. Dengan menekan tombol RETURN berkali-kali anda dapat mengamati pembentukan pola mata.

Sekarang anda akan mempelajari bagaimana caranya untuk mengekstraksi informasi yang bermanfaat dan terukur dari sebuah pola mata. Besaran-besaran yang dapat diukur dari sebuah pola mata ditunjukkan pada Gambar 3.



Gambar 3. Pola mata

A: periode sampling;

B: margin terhadap noise (setengah dari “bukaan mata”);

C: distorsi dari pelintasan nol;

D: kemiringan – menunjukkan kepekaan terhadap kesalahan pewaktuan (*timing*) ;

E: distorsi maksimum

t_{opt} : waktu sampling optimal yang diukur terhadap titik waktu nol. Hal ini adalah waktu terbaik yang mungkin untuk menyampel gelombang pada setiap durasi simbol T untuk mengekstraksi pesan yang dikirimkan dengan kesalahan sesedikit mungkin. Hal itu harus dipilih di tengah dari mata sedemikian hingga diperoleh B yang maksimal (dan juga akan meminimisasi kesalahan akibat derau) dan memaksimalkan jarak terhadap pelintasan nol terdekat (untuk memperkecil kesalahan terhadap *timing jitter*).

Sebuah urutan 1000-bit dikirimkan sebagai bipolar NRZ melalui kanal dengan bandwidth bw dan daya derau np . Bangkitkan urutan biner ini dan modulasikan dengan perintah-perintah yang sudah anda pelajari sebelumnya. Bangkitkan pola mata yang berkaitan untuk semua harga bw dan np yang tercantum pada Tabel 2. Catat t_{opt} , A dan B .

Tabel 2: Data pola mata untuk kode saluran NRZ bipolar.

np [W]	bw [Hz]	t_{opt} [dtk]	A [dtk]	B [V]
0.01	3000			
	2000			
	1000			
0.02	4000			
0.08				
0.10				

Ulangi langkah percobaan pada Bagian 3.2 untuk kode saluran Manchester dan catat hasilnya pada Tabel 3.

Tabel 3: Data pola mata untuk kode saluran Manchester.

np [W]	bw [Hz]	t_{opt} [dtk]	A [dtk]	B [V]
0.01	3000			
	2000			
	1000			
0.02	4000			
0.08				
0.10				

4. Pembentukan pulsa dan ISI

Kanal-kanal dengan bandwidth yang lebih kecil daripada bandwidth W nominal, seperti halnya juga kanal-kanal dengan spektrum yang tidak datar (misal: sebuah lekukan di tengah-tengah band yang bermanfaat) dapat menyebabkan penyebaran waktu (*time spreading*) dari sinyal. Jika penyebaran waktu tersebut lebih besar daripada durasi simbol, maka akan menghasilkan interferensi antar simbol (*intersymbol interference* atau ISI). Jika kita ambil sebuah simbol yang dikirim (pulsa) tersendiri, ISI berkaitan dengan distorsi yang disebabkan oleh sebagian energi dari dari pulsa-pulsa lain di dekatnya yang telah diproyeksikan ke simbol yang diinginkan oleh penyebaran waktu dari kanal. Pada bagian ini akan ditunjukkan bagaimana pembentukan pulsa sebelum pengiriman pada kanal dengan filter kirim (*transmit filter*) dapat mempengaruhi besarnya ISI pada penerima.

Fungsi Matlab khusus `isichannel` mendefinisikan jenis kanal yang akan mengakibatkan ISI yang cukup besar, yaitu sebuah tanggapan frekuensi datar dengan sebuah lekukan di tengahnya:

```
output =
isichannel(<input>,<gain>,<noise_power>,<bandwidth>,<width_of_notch>,<Fs>);
```

di mana lebar dari lekukan (*notch*) dinyatakan sebagai pecahan terhadap bandwidth total.

4.1 Pertama kali, bangkitkan sebuah pulsa NRZ unipolar yang diikuti dengan sederetan nol:

```
>>b = [1 zeros(1,20)];
>>s1 = modul(b,'unipolar_nrz');
```

Kirimkan urutan ini melalui sebuah kanal ISI tanpa derau dan dengan bandwidth 1500 Hz:

```
>>x1 = isichannel(s1,1,0,1500,0.2);
```

Kita hanya tertarik pada energi ISI yang terdapat pada waktu-waktu sampling. Jika kita mengirimkan sebuah pulsa yang disampel pada $T/2$, maka energi yang terdapat di $3T/2$, $5T/2$, $7T/2$, dst adalah ISI dan jumlah dari energi ini memberikan ukuran dari distorsi yang dibangkitkan oleh sebuah pulsa melalui kanal yang kita pelajari. Kita dapat menghitung besaran ini dengan fungsi khusus `computeisi`:

```
isi = computeisi(<signal>,<sampling_instant>,<origin>);
```

di mana `sampling_instant` dan `origin` dinyatakan sebagai kelipatan dari periode bit T . Titik asal biasanya diambil nol kecuali untuk pulsa yang tidak kausal. Kita dapat memeriksa dengan mudah bahwa sebelum pengiriman tidak ada ISI :

```
>>computeisi(s1,0.6,0)
```

Sebagai hasil dari perintah ini, Matlab akan memberikan hasil 0 (perhatikan bahwa tanda semicolon tidak perlu dituliskan pada akhir perintah agar hasilnya tercetak di layar). Sekarang hitung ISI pada penerima :

```
>>computeisi(x1,0.7,0)
```

```
ISI tanpa pembentukan pulsa = .....
```

- 4.2 Jika pulsa-pulsa yang dikirim dibentuk oleh filter pancar ideal untuk mengatasi ISI, hasilnya berupa pulsa Nyquist yang mempunyai durasi tak hingga tetapi harganya sama dengan 1 pada waktu penyampelan utama dan berharga nol pada waktu-waktu sampling yang lain.

```
>>s2 = modul(b,'unipolar_nyquist');
>>x2 = isichannel(s2,1,0,1500,0.2);
>>computeisi(s2,0.6,10)
>>computeisi(x2,0.7,10)
```

Sekali lagi, anda akan menjumpai bahwa tidak ada ISI sebelum pengiriman.

```
ISI tanpa pembentukan pulsa = .....
```

```
Persentase dari pengurangan ISI yang diperoleh dengan pembentukan pulsa = ..... %
```

TUGAS PASCA-PERCOBAAN

1. Kode-kode saluran apa yang akan membangkitkan gelombang tanpa komponen DC untuk urutan biner apapun ? Mengapa kadang-kadang pada prakteknya penting untuk mengkodekan sinyal yang akan dikirim sedemikian hingga tidak mengandung komponen DC ?
2. Pada output sebuah kanal, kita perlu untuk dapat mengekstraksi waktu laju-simbol (*symbol-rate clock*) yang sinkron dengan sinyanya yang diterima. Sinyal clock tersebut menunjukkan kapan sebuah simbol mulai dan oleh karena itu dapat digunakan untuk memicu penyampelan terhadap setiap simbol yang diterima. Hal ini disebut penemuan waktu (*timing recovery*), yang dipermudah oleh beberapa kode saluran. Sebutkan kode saluran yang mana saja. Jelaskan jawaban anda. Petunjuk: coba anda pelajari apa yang terjadi jika pesan yang dikirim berupa satu saja atau nol saja.
3. Untuk sebuah kanal komunikasi data baseband tertentu dengan bandwidth yang dapat dipakai 10 kHz, berapakah laju data maksimal yang dapat dicapai untuk setiap kode saluran yang dipelajari pada percobaan ini? (Lihat Bagian 1 dari langkah percobaan).
4. Mengingat bahwa derau kanal adalah aditif dan tidak berkorelasi dengan input kanal, turunkan persamaan dari PSD output kanal sebagai fungsi dari PSD input dan derau.
5. Pada Tabel 2 dan 3, bandingkan pola-pola mata yang diperoleh dengan daya derau $0.01 W$ dan bandwidth 1000 Hz. Untuk kode saluran yang mana anda amati pola mata yang terbaik? Jelaskan hasil ini berkaitan dengan sifat-sifat kode saluran dan kanal.
6. Berdasarkan pengamatan anda terhadap pola mata, apakah waktu sampling optimal t_{opt} tergantung pada daya dari derau aditif? Jelaskan.
7. Untuk NRZ bipolar pada 1000 bit/dtk, filter pembentuk pulsa terbaik untuk mengatasi ISI adalah sebuah filter *low-pass* ideal dengan frekuensi *cutoff* 1 kHz. Filter pembentuk pulsa apakah yang terbaik untuk kode saluran Manchester pada laju bit yang sama?