



# PRAKTIKUM

## PENGOLAHAN SINYAL INFORMASI

### MODUL 1

## PENGENALAN ANALOG TO DIGITAL CONVERSION

### 1.1 Tujuan Praktikum

1. Mengetahui pengertian Analog to Digital Conversion
2. Mengetahui pengenalan sinyal dan sistem
3. Mengetahui jenis-jenis sinyal
4. Mengetahui representasi matematis sinyal
5. Dapat merancang dan memvisualisasi berbagai jenis sinyal

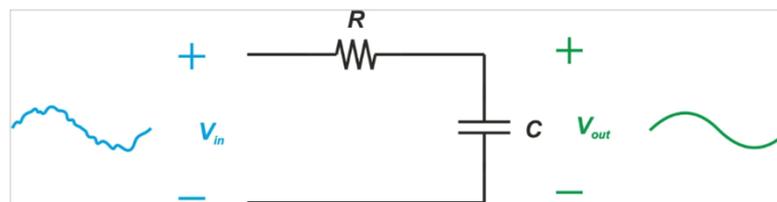
### 1.2 Alat dan Bahan

1. Laptop/PC
2. Visual Code
3. Mouse

### 1.3 Dasar Teori

#### 1.3.1 Pengenalan Analog Digital Conversion

Signal Processing (Pengolahan Sinyal) adalah bidang ilmu yang melibatkan manipulasi sinyal dari domain waktu ke domain frekuensi dan sebaliknya, menghaluskan sinyal, memisahkan noise (derau) dari sinyal (yaitu filtering atau penyaringan), serta mengekstrak informasi dari sinyal. Sinyal yang ada di alam adalah sinyal kontinu. Sinyal waktu kontinu (atau analog) ada dalam interval kontinu ( $t_1, t_2$ ) yang dapat berkisar dari  $-\infty$  hingga  $+\infty$ .



Gambar 1.

Rangkaian ini bertindak sebagai penyaring *low-pass*. Rangkaian ini membuang atau menyaring komponen frekuensi yang berada di atas frekuensi batas rangkaian dan melewatkan komponen frekuensi yang lebih rendah dengan sedikit redaman. Dalam contoh ini, tujuan pemrosesan sinyal adalah untuk menghilangkan derau frekuensi tinggi dan mengekstrak bagian sinyal yang diinginkan. Perhatikan bahwa input dan output keduanya dalam bentuk analog.



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

Ini merupakan keuntungan besar karena sinyal yang menarik dalam sains dan teknik bersifat analog. Oleh karena itu, dengan pemrosesan sinyal analog, tidak diperlukan sirkuit antarmuka (ADC dan DAC) pada input dan output blok pemrosesan sinyal.

#### **1.3.2 Kekurangan Pemrosesan Sinyal Analog**

Salah satu kelemahan utama pemrosesan sinyal analog adalah variasi nilai komponen listrik. Rangkaian analog bergantung pada ketepatan komponen aktif dan pasif (resistor, kapasitor, induktor, dan amplifier). Misalnya, frekuensi batas ( $f_C$ ) dari filter *low-pass* di atas diberikan oleh:

$$F_C = \frac{1}{2\pi RC}$$

Seperti yang dapat Anda lihat, respons filter merupakan fungsi dari nilai komponen. Karena komponen listrik tidak dapat diproduksi dengan presisi sempurna, akurasi sirkuit analog terbatas. Karena toleransi komponen, kinerjanya tidak dapat diulang 100% dan kami memperkirakan adanya beberapa variasi antarpapan dalam parameter sirkuit yang berbeda. Kerugian lainnya adalah bahwa rangkaian analog tidak fleksibel. Misalnya, untuk mengubah respons frekuensi filter di atas, kita perlu menyesuaikan nilai komponen (perangkat keras perlu dimodifikasi). Ini tidak terjadi pada pemrosesan sinyal digital. Dengan DSP, bahkan memungkinkan untuk mengubah filter *low-pass* menjadi filter *high-pass* hanya dengan mengubah beberapa koefisien yang dapat diprogram. Selain itu, rangkaian analog tidak cocok untuk mengimplementasikan fungsi matematika (perkalian, pembagian, dll.). Hal ini berbeda dengan ranah digital di mana operasi matematika yang jauh lebih canggih pun dapat diimplementasikan dengan mudah.



# PRAKTIKUM

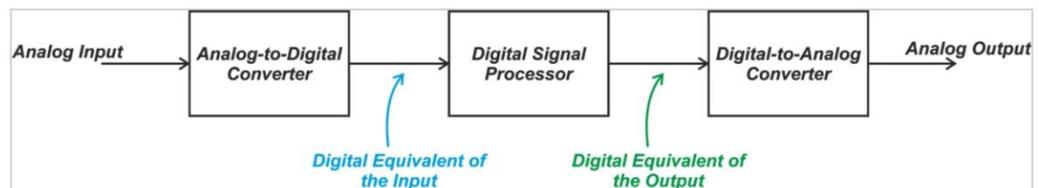
## PENGOLAHAN SINYAL INFORMASI

### 1.3.3 Pemrosesan Sinyal Digital Dapat Mengatasi Banyak Tantangan

Rangkaian digital tidak mengalami keterbatasan di atas. Misalnya, meskipun variasi nilai komponen dan parasit dapat sedikit mengubah penundaan gerbang inverter CMOS, fungsionalitas gerbang secara keseluruhan akan tetap dipertahankan. Oleh karena itu, tidak seperti rangkaian analog, rangkaian digital tidak terlalu rentan terhadap variasi komponen dan parasit. Rangkaian digital juga lebih fleksibel dan cocok untuk mengimplementasikan fungsi matematika.

Pertanyaan yang tersisa adalah, komponen dasar apa yang kita butuhkan untuk memproses sinyal dalam domain digital.

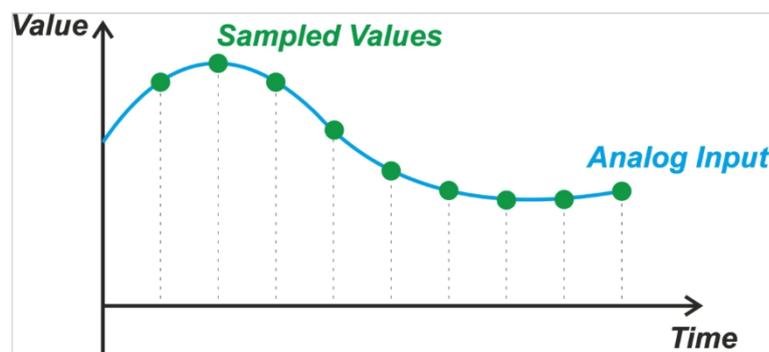
Seperti ditunjukkan dalam Gambar 2, kita memerlukan konverter analog-ke-digital (A/D) dan digital-ke-analog (D/A) pada input dan output blok pemrosesan sinyal untuk menghubungkan sirkuit digital kita dengan sinyal analog dunia nyata.



Gambar 2.

### 1.3.4 Pemrosesan Sinyal Digital Dapat Mengatasi Banyak Tantangan

Konverter A/D mengambil sampel masukan analog pada interval waktu tertentu seperti ditunjukkan pada Gambar 3.

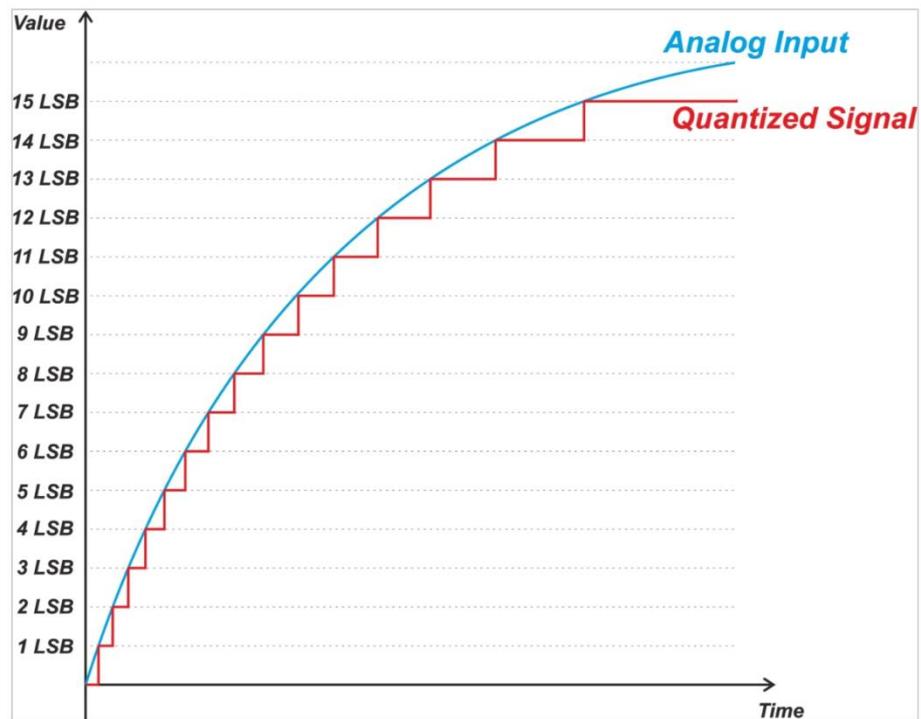


Gambar 3.

Kemudian, ia mengkuantisasi amplitudo setiap sampel. Gambar 4 menunjukkan bagaimana ADC 4-bit dapat mengkuantisasi masukan analog.



## PRAKTIKUM PENGOLAHAN SINYAL INFORMASI



Gambar 4.

Pada gambar ini, input analog (kurva biru) mengambil nilai yang berbeda dalam rentang input ADC. Jika mempertimbangkan ADC 4-bit, terdapat 16 level diskrit untuk mengkuantisasi amplitudo sinyal input. Level-level ini ditunjukkan dengan kelipatan LSB pada gambar. Oleh karena itu, LSB (bit paling signifikan) menentukan perubahan minimum pada nilai input analog yang dapat dideteksi oleh ADC. Dengan kata lain, perubahan minimum pada input inilah yang menyebabkan perubahan pada kode output ADC. Mari kita lihat bagaimana ADC menghasilkan kode biner untuk setiap sampel. ADC membandingkan amplitudo sinyal masukan analog dengan 16 level diskretnya. Berdasarkan perbandingan ini, representasi digital dari masukan dihasilkan. Misalnya, dengan kurva biru yang ditunjukkan pada Gambar 4, proses membandingkan sinyal masukan dengan 16 level diskret ADC dapat menghasilkan kurva merah yang digambarkan. Kemudian, ADC menggunakan kode biner untuk merepresentasikan setiap level dari perkiraan tangga yang diperoleh. Misalnya, ketika nilai kurva merah sama dengan 4 kali LSB, output dari ADC empat bit kita adalah 0100. Hal yang perlu diperhatikan adalah bahwa blok "prosesor sinyal digital" pada Gambar 2 menerima urutan waktu diskrit karena ADC mengambil sampel pada kelipatan interval pengambilan sampel yang telah ditentukan sebelumnya. Dan, amplitudo setiap sampel dikuantisasi. Hal ini berbeda dengan pemrosesan sinyal analog di mana inputnya adalah



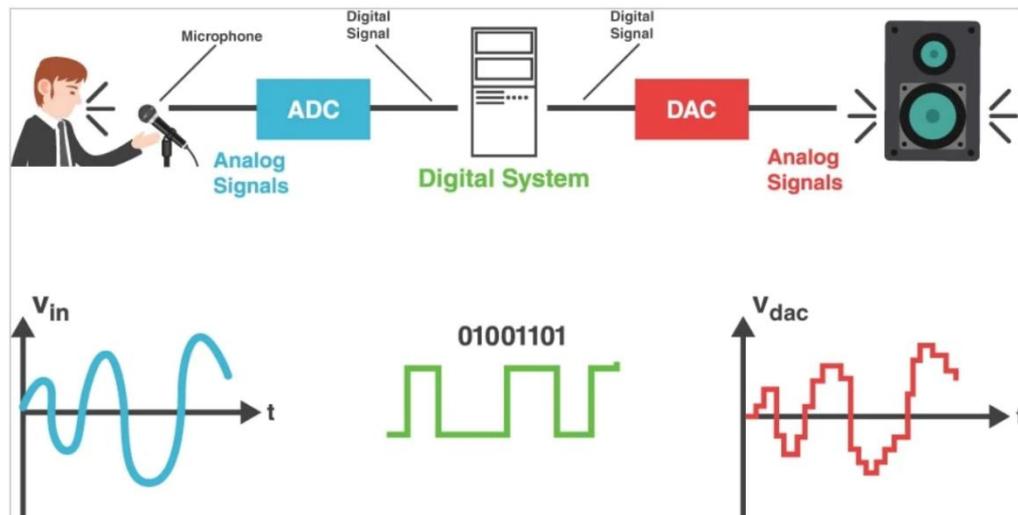
## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

sinyal waktu kontinu dan dapat mengambil nilai apa pun dalam rentang yang ditentukan.

#### 1.3.5 Peran DAC

Setelah sinyal diproses oleh blok “Digital Signal Processor”, kita biasanya perlu mengubahnya menjadi sinyal analog yang setara. Hal ini dicapai oleh konverter D/A. Gambar 5 menggambarkan aplikasi pemrosesan audio.



Gambar 5.

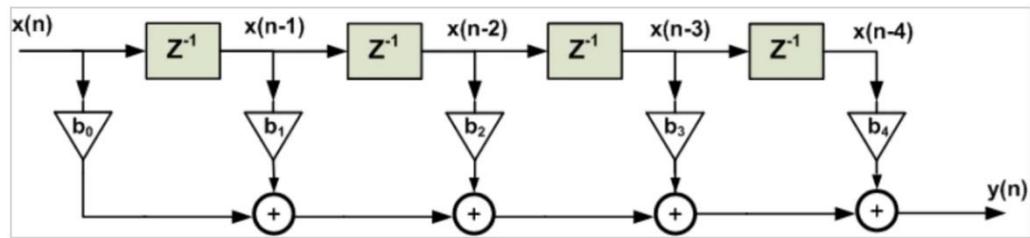
Dalam kasus ini, sistem pemrosesan sinyal digital digunakan untuk menambahkan gema atau menyesuaikan tempo dan nada suara untuk mendapatkan suara yang sempurna. Kemudian, sinyal yang diproses dikirim ke DAC untuk menghasilkan sinyal analog yang dapat dikeluarkan oleh pengeras suara. Perhatikan bahwa ada aplikasi DSP yang tidak memerlukan DAC. Misalnya, algoritma pemrosesan sinyal digital yang digunakan dalam radar dapat memberi kita posisi dan kecepatan pesawat terbang. Informasi ini dapat dicetak di atas kertas.

#### 1.3.6 Blok “Prosesor Sinyal Digital”

Algoritma DSP terdiri dari sejumlah operasi matematika. Misalnya, filter respons impuls terbatas (FIR) orde ke-4 memerlukan lima pengali digital, empat penjumlahan, dan beberapa elemen tunda seperti yang ditunjukkan di bawah ini.



## PRAKTIKUM *PENGOLAHAN SINYAL INFORMASI*



Oleh karena itu, prosesor sinyal digital sebenarnya adalah mesin komputasi. Mesin komputasi ini dapat berupa prosesor serbaguna, FPGA, atau bahkan chip DSP yang dibuat khusus. Setiap opsi memiliki kelebihan dan kekurangannya sendiri dalam hal fleksibilitas, kecepatan, kemudahan pemrograman, dan konsumsi daya. Karena sumber daya komputasi sangat berharga, pemrosesan sinyal digital berupaya menyediakan alat dan teknik yang memungkinkan algoritma yang cepat dan efisien secara komputasi. Misalnya, ada beberapa struktur berbeda yang dapat digunakan untuk mengimplementasikan filter FIR tertentu.

### 1.3.7 DSP dapat digunakan dalam berbagai Aplikasi

Konsep dan alat DSP digunakan dalam aplikasi apa pun yang perlu memanipulasi sinyal input dalam domain digital. Ini termasuk tetapi tidak terbatas pada kompresi audio dan video, pemrosesan dan pengenalan suara, pemrosesan gambar digital, dan aplikasi radar. Mengejar karier di masing-masing bidang ini memerlukan penguasaan berbagai algoritme, matematika, dan teknik DSP yang terspesialisasi. Bahkan, tampaknya sangat tidak mungkin bagi satu orang untuk menguasai semua teknologi DSP yang telah dikembangkan. Namun, beberapa konsep DSP umum seperti penyaringan, korelasi, dan analisis spektral digunakan di hampir semua aplikasi DSP. Oleh karena itu, langkah pertama dalam pendidikan DSP adalah menguasai konsep dasar dan kemudian fokus pada teknik khusus yang dibutuhkan bidang minat tertentu. Beberapa konsep dasar DSP serta beberapa teknik khusus yang terkait dengan pemrosesan gambar digital dibahas dalam artikel saya sebelumnya . Saya juga memiliki seri tentang FPGA dan implementasi algoritma DSP berbasis FPGA yang dapat membantu Anda memulai topik yang relatif sulit ini.

### 1.3.8 Proses Analog to Digital Conversion of Signals

Jika kita memiliki sinyal analog seperti yang dibuat oleh mikrofon atau kamera. Untuk mengubah sinyal analog menjadi data digital, menggunakan dua teknik, modulasi kode pulsa dan modulasi delta. Setelah data digital dibuat (digitalisasi) maka kita konversi data digital menjadi sinyal digital.

#### 1. Pulse Code Modulation (PCM)

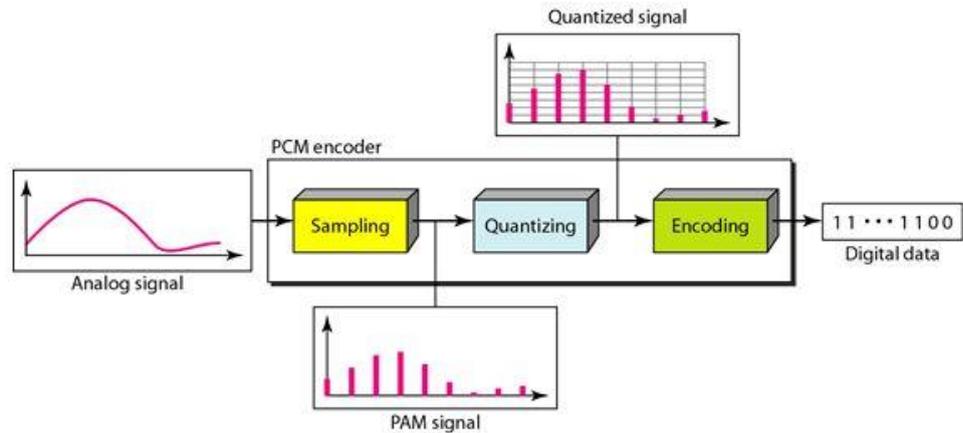


## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

adalah teknik paling umum yang digunakan untuk mengubah sinyal analog menjadi data digital (digitalisasi). Encoder PCM memiliki tiga proses seperti yang ditunjukkan pada Gambar berikut.

- Sinyal analog diambil sampel.
- Sinyal sampel dikuantisasi.
- Nilai terkuantisasi dikodekan sebagai aliran bit.



#### **A. Sampling**

Langkah pertama, dalam PCM adalah pengambilan sampel. Sinyal analog diambil sampelnya setiap  $T_s$  s, di mana  $T_s$  adalah interval atau periode sampel. Kebalikan dari interval pengambilan sampel disebut laju pengambilan sampel atau frekuensi pengambilan sampel dan dilambangkan dengan  $f_s$ , Di mana  $f_s = 1/T_s$ .

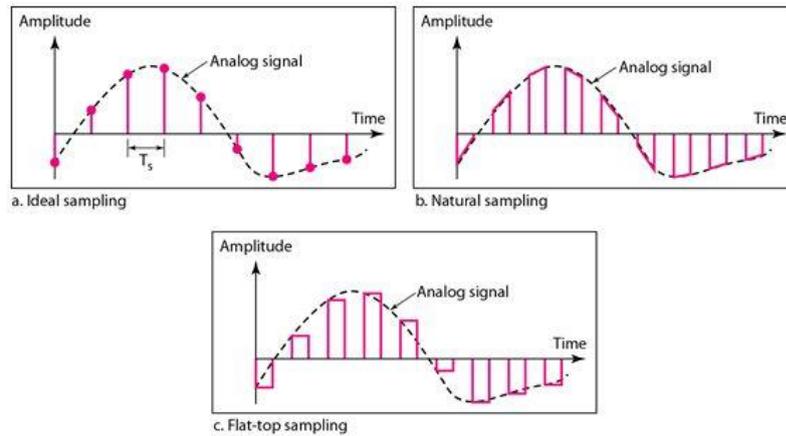
Ada tiga metode pengambilan sampel-ideal, alami, dan datar. Dalam pengambilan sampel yang ideal, pulsa dari sinyal analog diambil sampel. Ini adalah metode pengambilan sampel yang ideal dan tidak dapat dengan mudah diterapkan. Dalam pengambilan sampel alami, sakelar berkecepatan tinggi dihidupkan hanya untuk jangka waktu yang singkat ketika pengambilan sampel terjadi.

Hasilnya adalah urutan sampel yang mempertahankan bentuk sinyal analog. Metode pengambilan sampel yang paling umum, yang disebut sample and hold, bagaimanapun, membuat sampel datar dengan menggunakan sirkuit. Proses pengambilan sampel kadang-kadang disebut sebagai modulasi amplitudo pulsa (PAM). Metode pengambilan sampel yang berbeda adalah seperti yang ditunjukkan pada gambar berikut.



## PRAKTIKUM

### PENGOLAHAN SINYAL INFORMASI



Salah satu pertimbangan penting adalah laju atau frekuensi pengambilan sampel. Apa batasan pada  $T_s$ ? Menurut teorema Nyquist, untuk mereproduksi sinyal analog asli, salah satu syarat yang diperlukan adalah laju pengambilan sampel setidaknya dua kali frekuensi tertinggi dalam sinyal asli.

Adapun Teorema ini, Pertama, kita dapat mengambil sampel sinyal hanya jika sinyal tersebut terbatas pita yaitu sinyal dengan bandwidth tak terbatas tidak dapat diambil sampelnya. Kedua, laju pengambilan sampel harus setidaknya 2 kali frekuensi tertinggi, bukan bandwidth. Jika sinyal analog low-pass, bandwidth dan frekuensi tertinggi adalah nilai yang sama. Jika sinyal analog adalah bandpass, nilai bandwidth lebih rendah dari nilai frekuensi maksimum

#### B. Kuantisasi

Hasil pengambilan sampel adalah serangkaian pulsa dengan nilai amplitudo antara amplitudo maksimum dan minimum sinyal. Himpunan amplitudo bisa tak terbatas dengan nilai non-integral antara dua batas. Nilai-nilai ini tidak dapat digunakan dalam proses pengkodean. Berikut ini adalah langkah-langkah dalam kuantisasi

1. Kita berasumsi bahwa sinyal analog asli memiliki amplitudo sesaat antara  $V_{min}$  dan  $V_{max}$ .
2. Kita membagi rentang menjadi zona  $L$ , masing-masing tinggi  $\Delta$  (delta).

$$\Delta = (V_{max} - V_{min}) / L$$

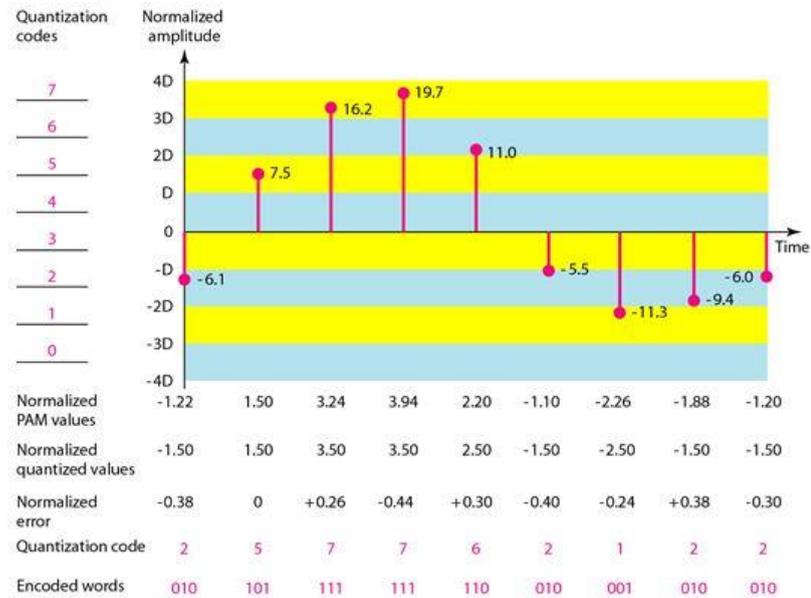
3. Kami menetapkan nilai terkuantisasi 0 ke  $L - 1$  ke titik tengah setiap zona.



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

4. Kami memperkirakan nilai amplitudo sampel ke nilai kuantisasi. Sebagai contoh sederhana.



Asumsikan bahwa kita memiliki sinyal sampel dan amplitudo sampel antara -20 dan +20 V. Kita memutuskan untuk memiliki delapan level ( $L = 8$ ). Ini berarti bahwa  $\Delta = 5$  V. Kami hanya menunjukkan sembilan sampel menggunakan pengambilan sampel ideal (untuk kesederhanaan). Nilai di bagian atas setiap sampel dalam grafik menunjukkan amplitudo aktual. Dalam bagan, baris pertama adalah nilai yang dinormalisasi untuk setiap sampel (amplitudo/ $\Delta$  aktual).

Proses kuantisasi memilih nilai kuantisasi dari tengah setiap zona. Ini berarti bahwa nilai terkuantisasi yang dinormalisasi (baris kedua) berbeda dari amplitudo yang dinormalisasi. Perbedaannya disebut kesalahan yang dinormalisasi (baris ketiga). Baris keempat adalah kode kuantisasi untuk setiap sampel berdasarkan tingkat kuantisasi di sebelah kiri grafik. Kata-kata yang dikodekan (baris kelima) adalah produk akhir dari konversi.

Salah satu masalah penting adalah kesalahan yang dibuat dalam proses kuantisasi. Kuantisasi adalah proses perkiraan. Nilai input ke quantizer adalah nilai nyata; Nilai keluaran adalah nilai perkiraan. Nilai keluaran dipilih untuk menjadi nilai tengah di zona. Jika nilai input juga berada di tengah zona, tidak ada kesalahan kuantisasi; jika tidak, ada kesalahan. Pada contoh sebelumnya, amplitudo yang dinormalisasi dari sampel ketiga adalah 3,24, tetapi nilai terkuantisasi yang dinormalisasi adalah 3,50. Ini berarti ada kesalahan +0,26. Nilai kesalahan untuk



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

---

sampel apa pun kurang dari  $\Delta/2$ . Dengan kata lain, kita memiliki  $\Delta/2 \leq \text{error} \leq \Delta/2$ .

#### **C. Encoding**

Langkah terakhir dalam PCM adalah pengkodean. Setelah setiap sampel dikuantisasi dan jumlah bit per sampel diputuskan, setiap sampel dapat diubah menjadi kata kode nb-bit. Pada gambar di atas, kata-kata yang dikodekan ditampilkan di baris terakhir. Kode kuantisasi 2 dikodekan sebagai 010; 5 dikodekan sebagai 101; dan seterusnya. Perhatikan bahwa jumlah bit untuk setiap sampel ditentukan dari jumlah tingkat kuantisasi. Jika jumlah tingkat kuantisasi adalah  $L$ , jumlah bit adalah  $n_b = \log_2 L$ . Dalam contoh kita  $L$  adalah 8 dan  $n_b$  oleh karena itu adalah 3. Kecepatan bit dapat ditemukan dari rumus.

$$\text{Bitrate} = \text{Sampling rate} \times \text{Jumlah gigitan per sampel} = f_s \times$$

