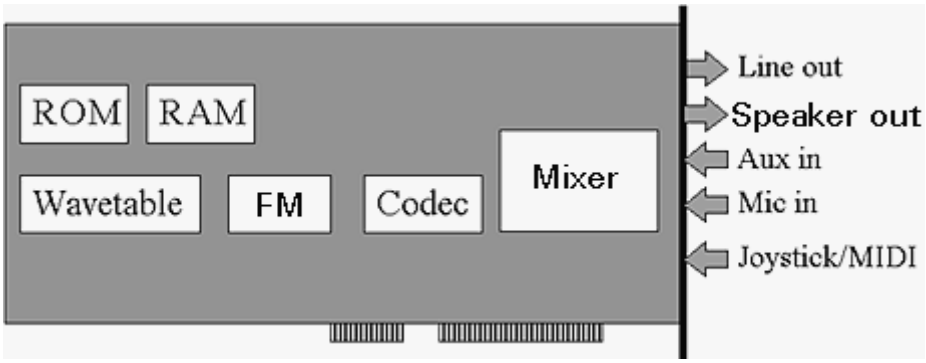
2. LANDASAN TEORI

2.1. *Soundcard*

Gambar berikut menunjukan bagian-bagian yang ada pada tipe-tipe *modern wavetable soundcard* (contoh: AWE32, AWE64):

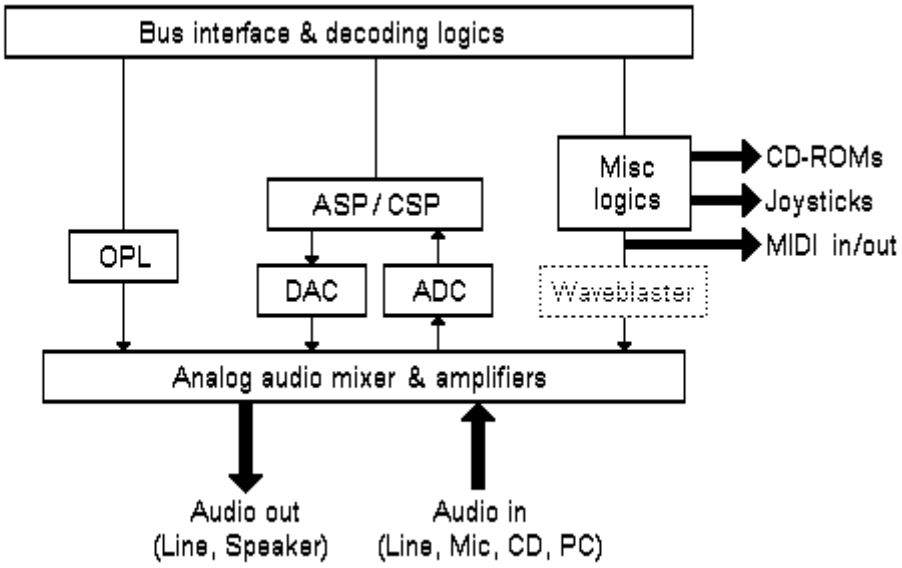


Gambar 2.1. Bagian-bagian dari Tipe *Modern Wavetable Soundcard*
Engdahl, Tomi. "Soundcard Tips and Facts." ELH Communication Ltd. 28 May. 2002 <http://www.epanorama.net/documents/pc/soundcard_tips.html>.

Bagian-bagian dari gambar 2.1 adalah:

- ROM berisi *wavetable syntetizer samples* yang bisa diatur
- RAM untuk peralatan *wavetable* yang bisa di-*download*
- *Wavetable syntetizer* membuat suara di luar *sample* ROM dan RAM
- CODEC melakukan perubahan A/D atau D/A untuk sinyal-sinyal *audio*
- FM *syntetizer* memainkan suara-suara FM (untuk *original Sound Blaster/Adlib compatibility*)
- MIXER adalah sebuah IC *mixer* analog yang menggabungkan suara dari berbagai *input*-an (*microphone, aux input, wavetable syntetizer*, FM syntetizer, CD-ROM *audio*) yang kemudian dikirimkan ke *line level* dan *speaker output.*

Berikut adalah blok diagram dari *soundcard* 16-bit:



Gambar 2.2. Blok Diagram *Soundcard* 16-bit
Engdahl, Tomi. "Soundcard Tips and Facts." *ELH Communication Ltd.* 28 May. 2002 <http://www.epanorama.net/documents/pc/soundcard_tips.html>.

Sistem perekaman dibedakan menjadi:

- *Full Duplex*

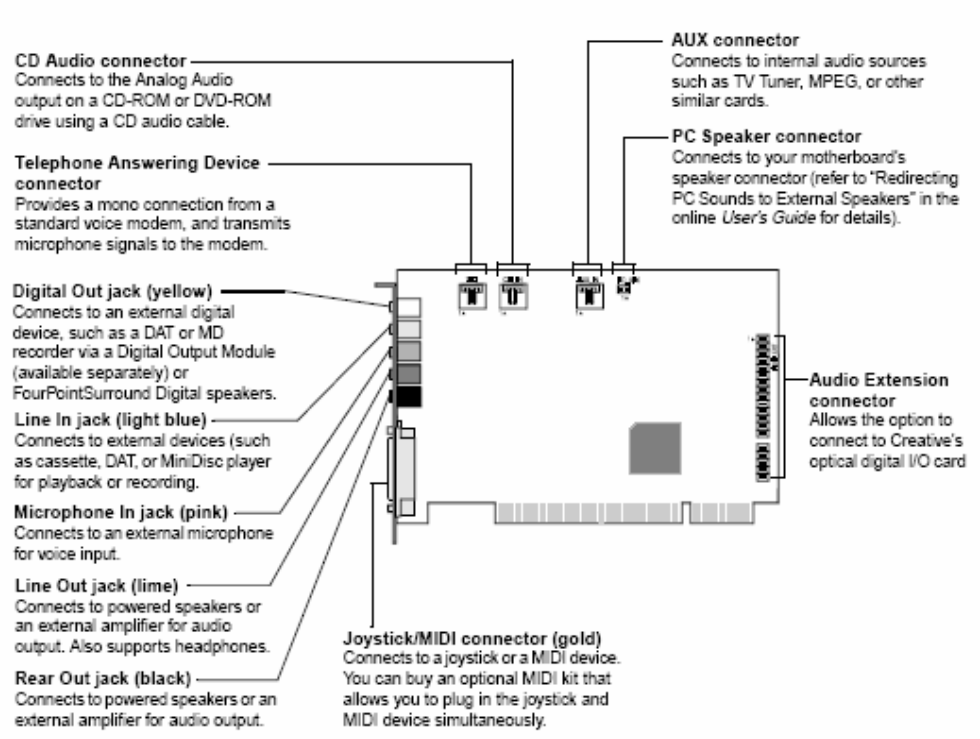
  Soundcard dapat merekam (*recording*) suara dan memainkan (*playback*) suara pada saat yang sama. *Sample rate playback* dan *recording* harus sama.

- *Enhanced Full Duplex*

  Merekam suara pada satu *sampling rate*, dan memainkan suara pada *sampling rate* yang berbeda secara bergantian.

- *Half Duplex*

  Dapat merekam suara atau *playback* suara, tetapi tidak dapat melakukan kedua hal tersebut pada waktu yang sama. *Game voice* tidak dapat bekerja dengan baik dengan *half duplex sound card.*



Gambar 2.3. *Jack* dan Konektor pada *Soundcard Sound Blaster Live!*
Sound Blaster Live! Support Datasheet. Creative Technology Ltd. America. 2002.

Pada gambar di atas digambarkan jenis-jenis konektor yang ada pada *soundcard Sound-Blaster Live!* yang pada umumnya kurang lebih sama dengan tipe-tipe lain yang sekelasnya. Beberapa konektor yang terdapat pada *soundcard* yaitu:

- *Digital out jack* (*yellow*)

  Koneksi ke sebuah peratalan digital eksternal, seperti *MD recorder* melalui sebuah *digital output module atau four point surround digital speakers*

- *Line in jack / aux in* (*light blue*)

  Koneksi ke peralatan eksternal seperti *cassete*, DAT, atau *minidisc player* untuk *playback* atau *recording.* Line in adalah stereo *input* (*left* and *right channel*).

- *Microphone in jack* (*pink*)

  Koneksi ke sebuah *microphone* eksternal untuk *input*-an suara. *Microphone input* adalah mono *input* dan tidak dapat digunakan untuk *stereo input.*

- *Line out jack* (*lime*)

  Koneksi ke *powered speakers* atau sebuah eksternal *amplifier* untuk *audio output.* Juga men-*support headphones.*

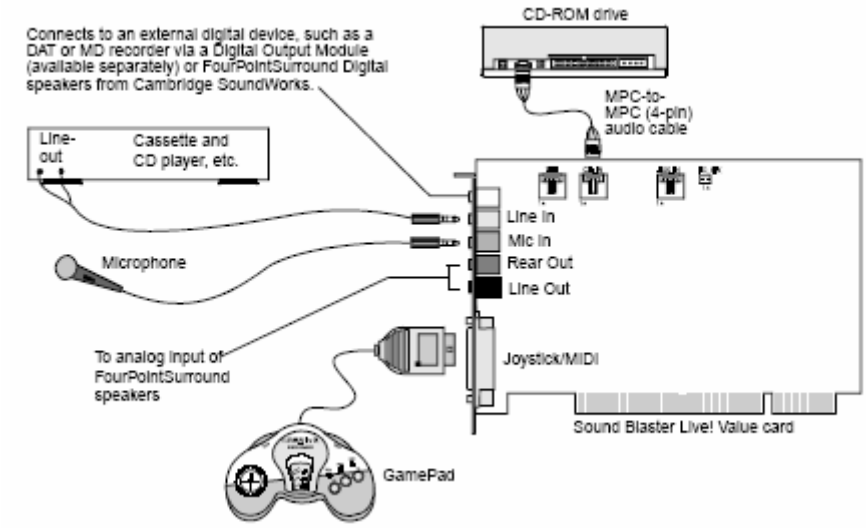
- *Rear out jack* (*black*)

  Koneksi ke *powered speakers* atau sebuah eksternal *amplifier* untuk *audio output.*

- *Joystcik*/MIDI (*gold*)

  Koneksi ke *joystick* atau *midi device.*

Penghubungan konektor-konektor pada *soundcard* dapat dilihat ilustrasinya pada gambar berikut:



Gambar 2.4. *Soundcard* Terhubung pada Digital *Device* Ekternal
Sound Blaster Live! Support Datasheet. Creative Technology Ltd. America. 2002.

2.2. *Function Generator*

Function generator juga sering dikenal sebagai *signal generator* adalah suatu modul yang berguna untuk menghasilkan *output* berupa sinyal meliputi gelombang sinus, segitiga, kotak, ataupun campuran atau acak. Sinyal tersebut mempunyai parameter-parameternya seperti; tegangan, frekuensi, amplitudo, dan fasa. *Output* ini digunakan oleh peralatan elektronik lain sebagai *input*-an tegangan atau sinyal.

2.3 LM555

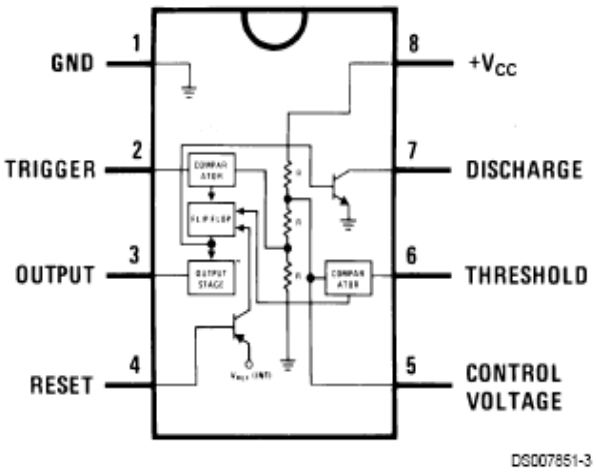
LM555 adalah peralatan untuk menghasilkan *time delays* atau osilasi yang akurat. Beberapa terminalnya berfungsi untuk *trigger* atau *reset* (jika diperlukan). Untuk *mode* operasi *time delay*, waktu dikontrol oleh satu resistor *external* dan kapasitor. Untuk operasi *astable* sebagai *oscillator*, *free running frequency* dan *duty cycle* dikontrol secara akurat dengan dua eksternal resistor dan satu kapasitor. Beberapa kemampuan dari LM555, yaitu:

- *Timing* dari *microseconds* sampai jam
- Mengoperasikan *mode astable* dan *monostable*
- *Duty cycle* dapat diatur
- *Output* dapat menghasilkan 200mA
- Kompatibel sebagai *output* dan *supply* TTL
- Kestabilan temperatur lebih baik dari 0.005% per C
- *Outout normally on* dan *normally off*
- Tersedia 8-pin paket MSOP

Beberapa aplikasi yang menggunakan LM555, yaitu:

- *Precision timing*
- *Pulse generation*
- *Sequential timing*
- *Time delay generation*
- *Pulse width modulation*
- *Pulse position modulation*
- *Linier ramp generator*

Konfigurasi pin LM555 dapat dilihat pada gambar 2.5. berikut.

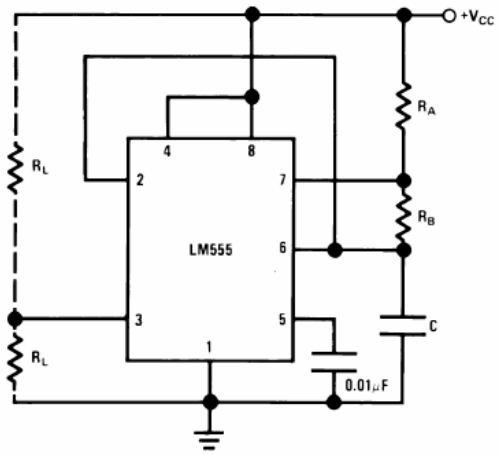


Gambar 2.5. Konfigurasi Pin LM555

National Semiconductor LM555 Timer Datasheet. National Semiconductor Corporation. America. February 2000.

2.3.1. *Astable Operation*

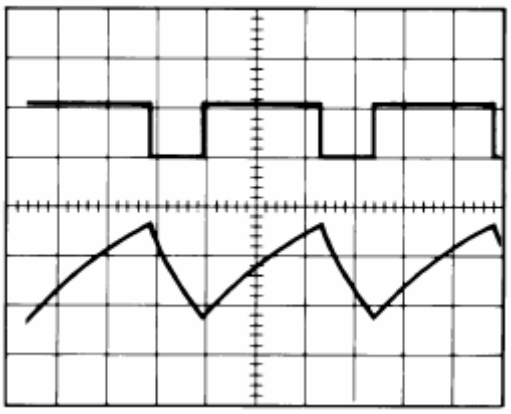
Salah satu aplikasi dari IC LM555 adalah *astable mutivibrator* yaitu dengan rangkaian dihubungkan seperti pada gambar 2.6 (port-2 dan port-6 terhubung). Rangkaian tersebut akan men-*trigger* secara sendirinya dan bekerja sebagai *multivibrator*. Kapasitor eksternal *charges* melalui Ra + Rb dan *discharge* melalui Rb. *Duty cycle* di-*set* dengan rasio dari dua buah resistor tersebut.



Gambar 2.6. *Astable Multivibrator Circuit*

National Semiconductor LM555 Timer Datasheet. National Semiconductor Corporation. America. February 2000.

Pada *mode* operasi seperti ini , kapasitor *charges* dan *discharges* antara 1/3Vcc dan 2/3 Vcc. Pada *mode trigger*, waktu *charge* dan *discharge*, dan frekuensi tidak berhubungan dengan tegangan *supply*. Pada gambar 2.7 ditunjukan hasil *waveform* pada *astable mode* ini.



Gambar 2.7. *Astable Waveform*
National Semiconductor LM555 Timer Datasheet. National Semiconductor Corporation. America. February 2000.

Perhitungan rangkaian *astable*:

- Waktu *charge* (*output high*):

$$t_1 = 0.693 (R_A + R_B) C \quad (2.1)$$

- Waktu *discharge* (*output low*):

$$t_2 = 0.693 (R_B) C \quad (2.2)$$

- Periode total:

$$T = t_1 + t_2 = 0.693 (R_A + R_B) C \quad (2.3)$$

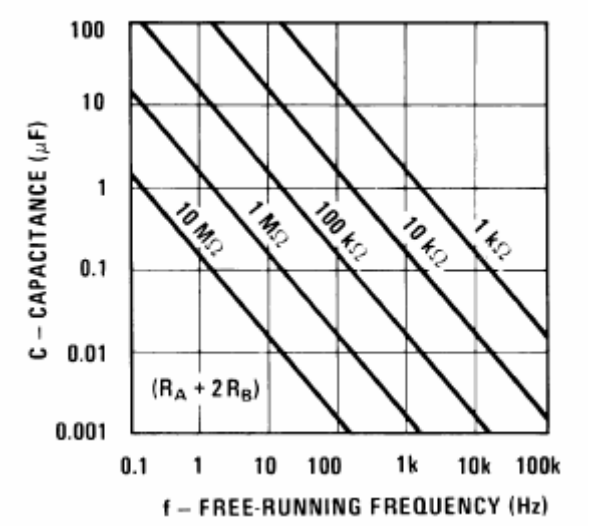
- Frekuensi osilasi:

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B)C} \quad (2.4)$$

Berikut ditampilkan Gambar 2.8 yang digunakan untuk menentukan nilai dari RC.

Perhitungan *duty cyle* adalah :

$$D = R_B / (R_A + 2R_B) \quad (2.5)$$



Gambar 2.8. Frekuensi *Free Running*
National Semiconductor LM555 Timer Datasheet. National Semiconductor Corporation. America. February 2000.

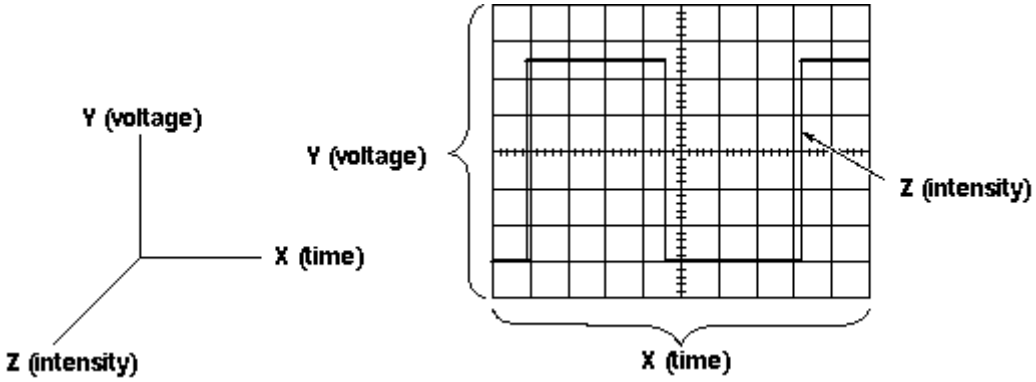
2.4. *Oscilloscope*

Oscilloscope adalah suatu peralatan yang sangat berguna dalam *instrument electronic. Oscilloscope* tersebut digunakan untuk mengukur signal dengan tegangan terhadap waktu. Tidak seperti *voltmeter* yang hanya menampilkan angka saja, o*scilloscope* menampilkan bentuk sinyal, dalam fungsi waktu. *Oscilloscope* dapat mengukur sinyal dengan parameter frekuensi, tegangan *peak-to-peak*, dan lain-lain.

Oscilloscope pada dasarnya adalah sebuah alat untuk menampilkan sebuah grafik, yaitu grafik dari sinyal elektrik. Perubahan sinyal ditunjukkan berdasarkan fungsi waktu dengan sumbu vertikal (Y) mewakilkan tegangan dan sedangkan sumbu horizontal (X) mewakilkan waktu. Intensitas atau terangnya dalam menampilan gambar grafiknya sering disebut sebagai sumbu Z (lihat gambar 2.9). Dengan o*scilloscope* dapat dilihat beberapa hal mengenai sinyal yaitu:

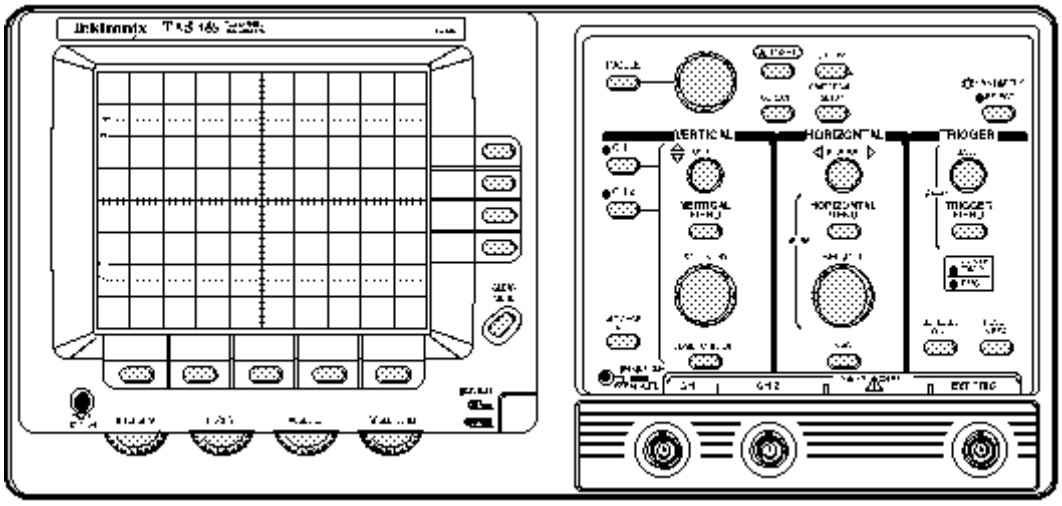
- Penentuan nilai waktu dan tegangan sinyal
- Kalkulasi frekuensi sebuah sinyal
- Melihat sebuah peralatan elektronik / rangkaian elektronik yang bekerja yang digambarkan sebagai sebuah sinyal.

- Dapat melihat kinerja sebuah komponen yang tidak baik atau tidak baik dari sebuah sinyal yang rusak.
- Menentukan seberapa banyak sebuah sinyal tersebut mempunyai *noise* / gangguan sinyal tidak diinginkan seiring perubahan waktu.

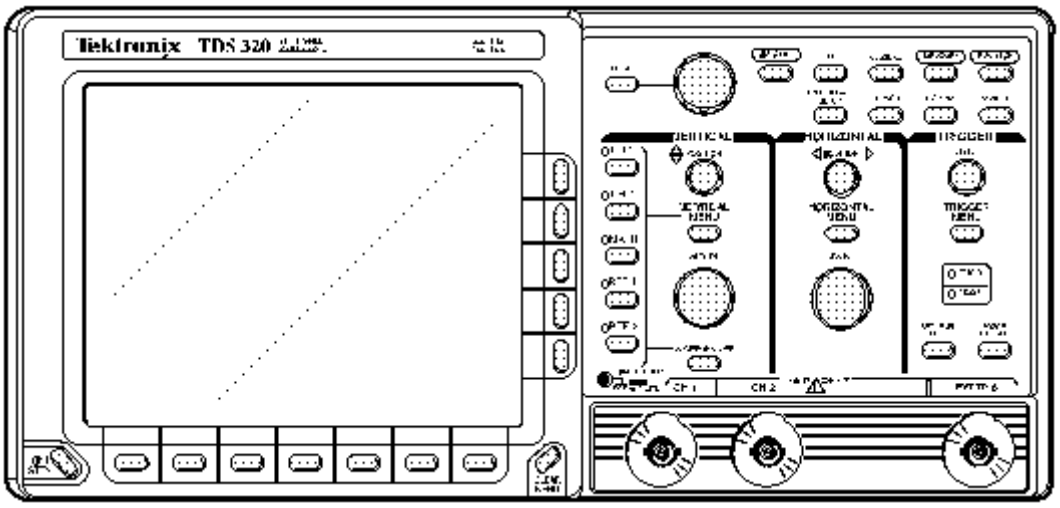


Gambar 2.9. X, Y, *and* Z *Components of a Displayed Waveform*
"Oscilloscope Tutorial." Department of Computer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/oscilloscope.html>.

Sebuah *oscilloscope* terlihat seperti sebuah televisi kecil, kecuali beda karena adanya tampilan garis-garis sumbu pada layar dan adanya tombol-tombol kontrol. Berikut contoh tampilan depan panel *oscilloscope*;



Gambar 2.10. TAS 465 *Analog Oscilloscope Front Panel*
"Oscilloscope Tutorial." Department of Computer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/oscilloscope.html>.

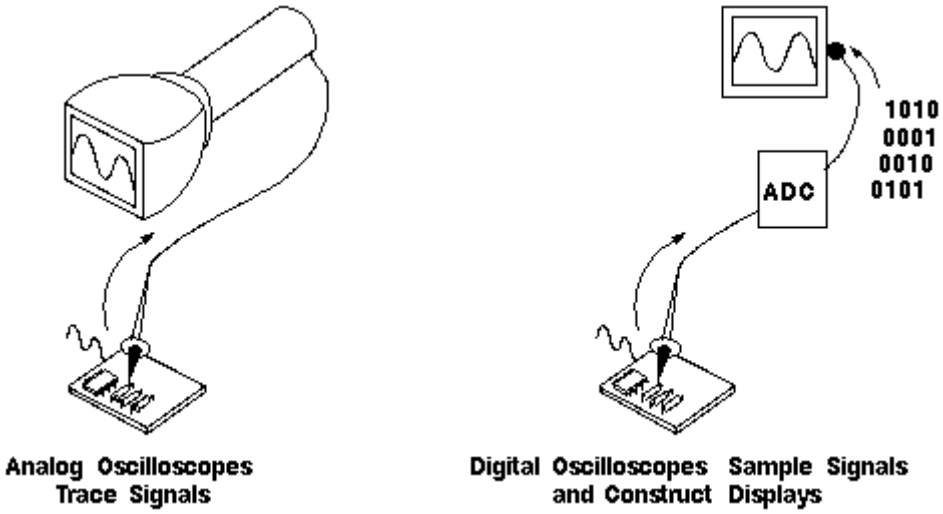


Gambar 2.11. TDS 320 *Digital Oscilloscope Front Panel*
"*Oscilloscope Tutorial*." Department of Computer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/oscilloscope.html>.

Peralatan elektronik dapat dibedakan menjadi dua tipe yaitu analog dan digital. Peralatan analog bekerja dengan *variable* tegangan yang berkesinambungan, sedangkan peralatan digital bekerja dengan data biner yang *discrete* yang mewakilkan *voltage samples*.

Oscilloscope juga tersedia tipe analog atau digital. Tipe analog bekerja menerima langsung sebuah tegangan dan digambarkan *waveform* pada layar. Cara tersebut memberikan tampilan *waveform* secara cepat.

Sebuah *oscilloscope* digital, *sample waveform* menggunakan sebuah *analog-to-digital converter* (ADC) untuk mengubah tegangan yang diukur menjadi informasi digital. Informasi digital tersebut digunakan untuk mengkonstruksikan *waveform* pada *layar*.



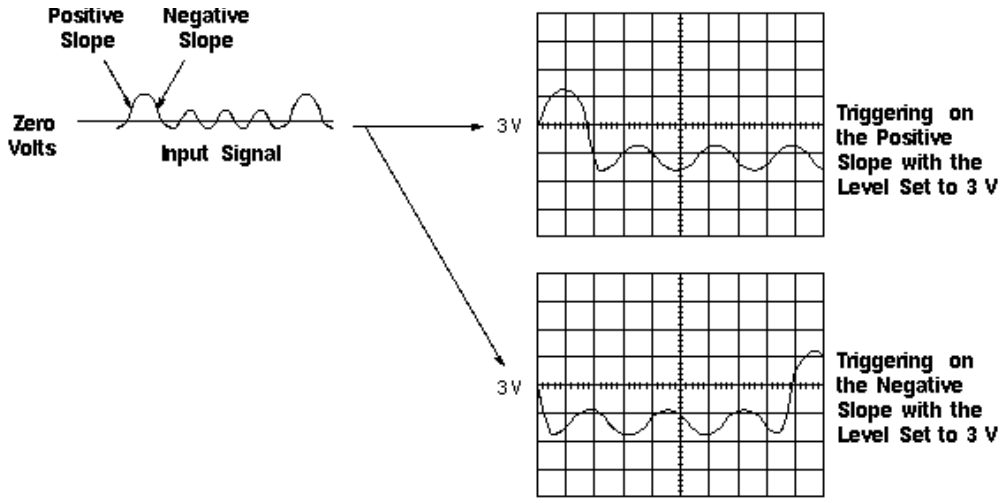
Gambar 2.12. *Digital and Analog Oscilloscopes Display Waveforms* "*Oscilloscope Tutorial*." Department of Computer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/ oscilloscope.html>.

Sedangkan *oscilloscope* digital dapat dilakukan *capture* dan melihat suatu peristiwa yang mungkin terjadi sekali saja. *Oscilloscope* digital dapat memproses digital *waveform* data atau mengirim data ke komputer untuk diproses dan juga dapat menyimpan digital *waveform* data untuk dicetak (*printing*).

2.4.1. *Trigger Controls*

Trigger controls membuat *waveform* yang berulang tampil *static* pada *oscilloscope display*. Pada *oscilloscope* terdapat berbagai macam tipe *trigger*, salah satunya adalah *edge trigger*.

Rangkaian *trigger* bekerja sebagai sebuah komparator. Dipilih *slope* dan *voltage level* dari sebuah sisi komparator, kemudian jika sinyal *trigger* sesuai dengan setting, *oscilloscope* menghasilkan *trigger. Slope control* menentukan *trigger point* pada saat *rising* atau *falling edge* dari sinyal. *Rising edge* adalah slope positif dan *falling edge* adalah *slope* negatif. *Level control* menentukan dimana *trigger point* berada pada *edge*. Gambar 2.13 menunjukkan bagaimana *trigger slope* dan *level setting* menentukan sebuah *waveform* ditampilkan.



Gambar 2.13. *Positive* dan *Negative Slope Triggering*
"*Oscilloscope Tutorial*." Department of Computer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/controls.html>.

Sumber dari *trigger* dapat berasal *input channel*, sinyal luar selain *input channel, power source signal,* sinyal secara *internal* diberikan dari *oscilloscope.*

2.4.2. *Trigger Modes*

Trigger mode menentukan terjadinya proses atau tidaknya sebuah *oscilloscope* menggambarkan *waveform*. *Trigger mode* meliputi *normal* dan *auto.* Pada *normal mode, oscilloscope sweeps* jika sinyal input mencapai *set trigger point*. Hal tersebut pada analog *oscilliscope* ditampilkan *blank* pada layar dan sedangkan untuk digital *oscilloscope* berhenti pada *waveform* terakhir yang tertampilkan. *Auto mode* menyebabkan *oscilloscope sweeps* meskipun tidak ada *trigger*. Jika tidak ada sinyal, *display* tidak menghilang meskipun sinyal mengalami *drop* tegangan. Hal ini adalah *mode* terbaik untuk digunakan jika ingin menampilkan banyak sinyal dan tidak mau terganggu dengan *setting trigger* setiap saat.

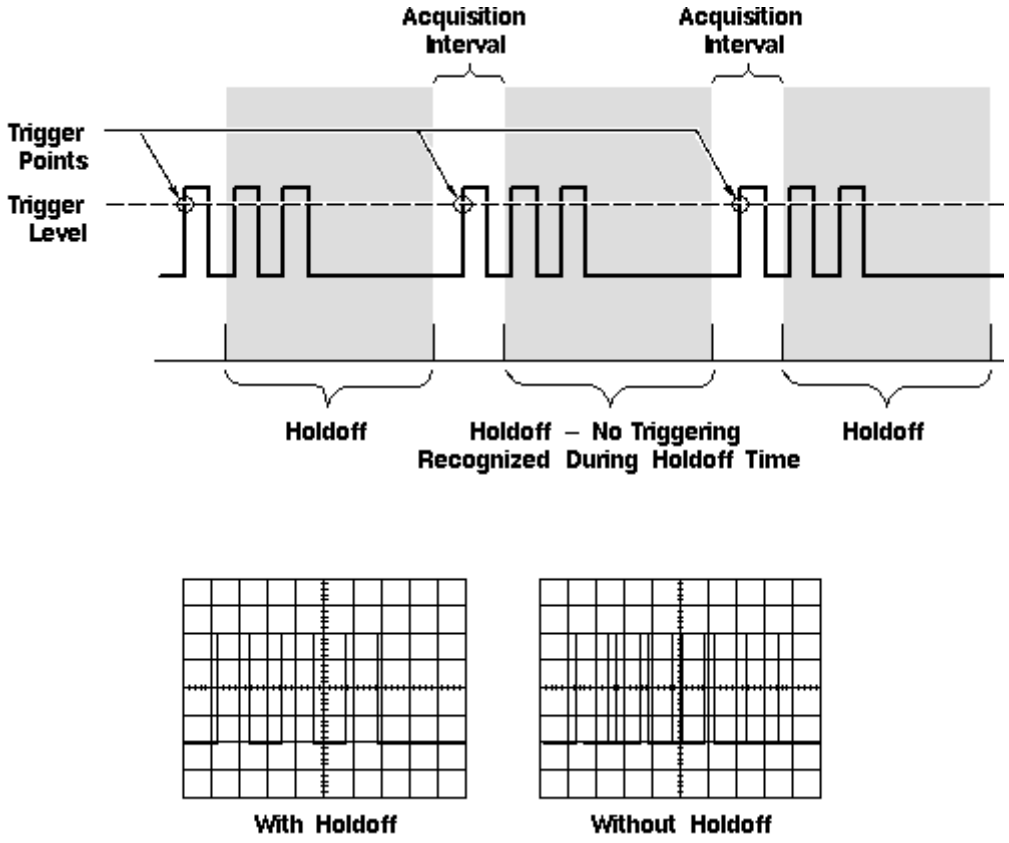
2.4.3. *Trigger Coupling*

Trigger coupling terdapat pilihan AC atau DC *coupling* untuk *vertical system*. Selain AC dan DC *coupling, oscilloscope* memiliki *high frequency*

rejection, low frequency rejection, dan *noise rejection trigger coupling*. *Setting* spesial ini berguna untuk menghilangkan *noise* dari sinyal *trigger*.

2.4.4. *Trigger Holdoff*

Trigger holdoff adalah pengaturan periode waktu pada saat *oscilloscope* tidak dapat men-*trigger*. *Trigger holdoff* tersebut berguna ketika men-*trigger waveform* yang kompleks, sehingga *oscilloscope* hanya men-*trigger* pada *trigger point* pertama yang dipilih. Hal tersebut digambarkan pada gambar 2.14.

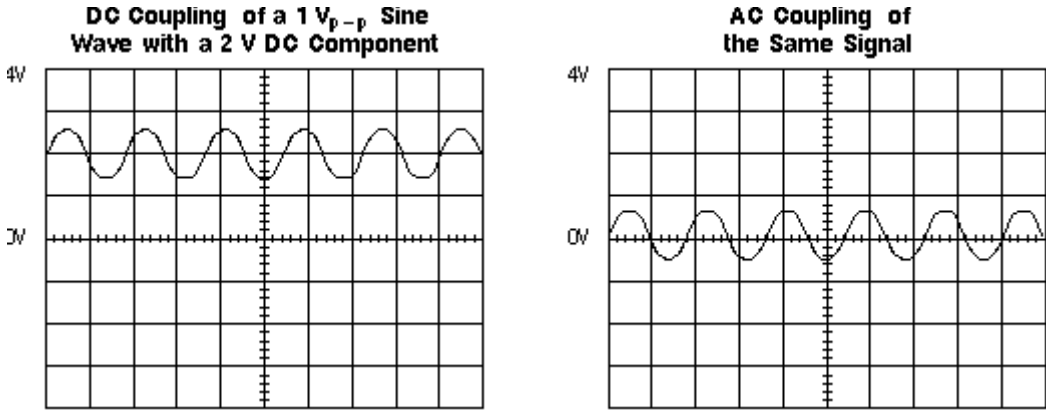


Gambar 2.14. *Trigger Holdoff*
"*Oscilloscope Tutorial*." Department of Computer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/controls.html>.

2.4.5. *Input Coupling*

Coupling adalah motode yang digunakan untuk menghubungkan sebuah sinyal elektrik dari satu rangkaian ke rangkaian lain. Dalam kasus ini, *input*

coupling terhubung dari rangkaian yang akan di *test* ke *oscilloscope*. *Coupling* dapat di-*set* DC, AC, atau *ground*. DC *coupling* menunjukan semua *input sinyal*. AC *coupling* menahan DC *component* sebuah sinyal sehingga *waveform* tampil di tengah-tengah pada *zero volts*. Kedua hal terebut diperlihatkan pada gambar 2.15.

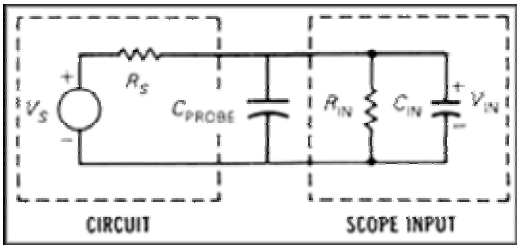


Gambar 2.15. AC dan DC *Input Coupling*
"Oscilloscope Tutorial." Department of Computer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/controls.html>.

Ground setting memisahkan sinyal input dari *vertical system* agar terlihat posisi *zero volts* pada layar. Dengan *grounded input coupling* dan *auto trigger mode,* dapat dilihat sebuah garis horizontal pada layar yang mewakili *zero volts*.

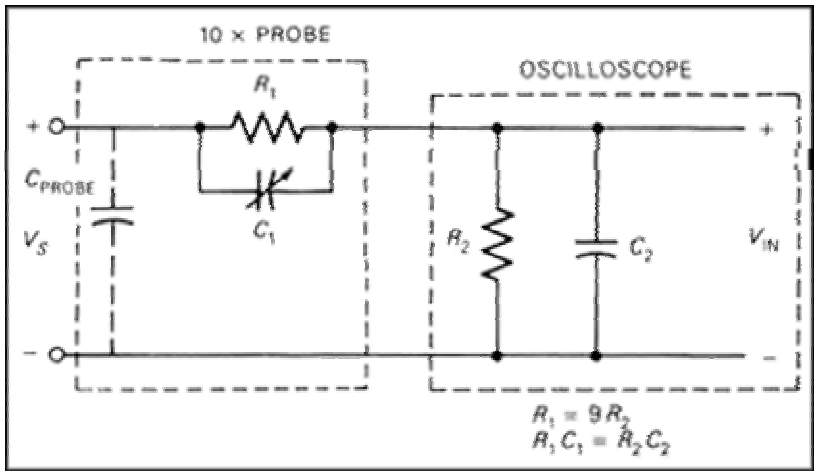
2.4.6 . *Probe*

1X *probes* juga dikenal sebagai 1:1 (*one-to-one*) *probes*, menghubungkan *high impedance input* sebuah *oscilloscop*e ke rangakaian yang akan diukur. *Probe* didesain untuk kerusakan minimum dan kemudahan koneksi. Gambar 2.16 menunjukkan diagram rangkaian untuk *high impedance scope input* dengan sebuah resistor seri. 1X *probe* (atau kabel) akan memberikan sebuah nilai signifikan akan sebuah kapasitansi yang muncul pada paralel *input scope*. 1X *probe* memiliki kapasitansi 40-60 pF, yang mana lebih pada umumnya lebih besar dari kapasitansi *input oscilloscope.*



Gambar 2.16 *High-Impedance Input* Terhubung ke Rangkaian Menggunakan 1X *Probe.*
Witte, Robert A. *Attenuating Probes*. Electronic Test Instruments, Prentice Hall PTR. National Instruments Corporation. <http://zone.ni.com/devzone/devzoneweb.nsf/Opendoc?openagent&3313443A44AE7CAB8625684600556380>.

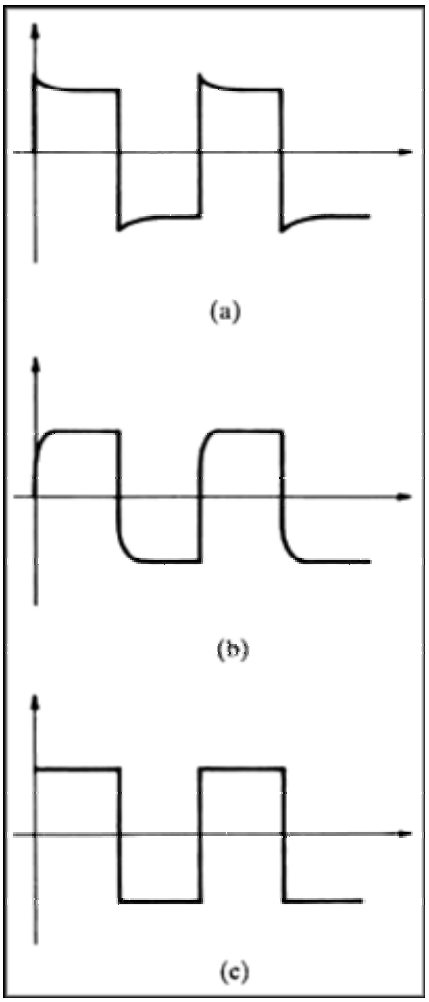
10X *Probes* (disebut juga 10:1 *probes, devider probes*, atau *attenuating probes*) mempunyai sebuah resistor dan kapasitor (paralel) yang dimasukkan ke dalam *probe*. Gambar 2.17 menunjukkan rangkaian untuk 10X *probe* terhubung ke sebuah *high impedance input* dari *oscilloscope*. Dengan 10X *probe oscilloscope* hanya melihat 1/10 tegangan aslinya sehingga dalam menggunaan 10X *probe* harus mengalikan hasil yang diukur dengan 10. *Load impedance* yang dimiliki lebih besar yaitu 10MΩ.



Gambar 2.17. Diagram Rangkaian 10X *Probe* Terhubung Dengan *Oscilloscope High Impedance Input.*
Witte, Robert A. *Attenuating Probes*. Electronic Test Instruments, Prentice Hall PTR. National Instruments Corporation. <http://zone.ni.com/devzone/devzoneweb.nsf/Opendoc?openagent&3313443A44AE7CAB8625684600556380>.

Probe dihubungkan ke sumber *square wave* yang disebut *calibrator* (tersedia dalam *oscilloscope*). *Probe* melakukan pengaturan untuk membuat

squarewave menjadi persegi yang sempurna. Gambar 2.18a dan 2.18b menunjukkan layar oscilloscope saat kompensasi dengan *overcompensated* dan *undercompensated probe*. Gambar 2.18c menunjukkan tampilan ketika *probe* sudah dikompensasikan.



Gambar 2.18. Contoh 10X *probe compensation*, (a) *Overcompensated*. (b) *Undercompensated* (c) *Properly compensated*

2.4.7. Metode-metode *Sampling*

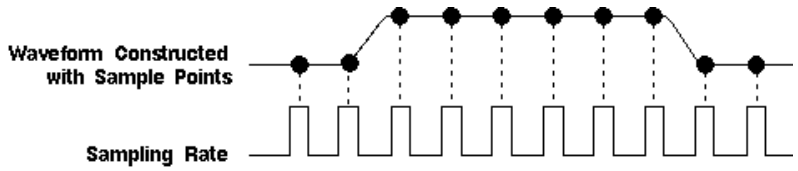
Metode *sampling* pada *oscilloscope* digital adalah cara untuk mengambil *sample point*. Untuk perubahan sinyal yang lambat, sebuah *oscilloscope* digital dengan mudah mengambil lebih dari cukup *sample points* untuk mengkonstruksikan sebuah gambar yang akurat. Untuk sinyal yang cepat,

oscilloscope tidak dapat mengambil *sample* yang cukup. *Oscilloscope* digital dapat melakukan dua hal, yaitu:

- *Oscilloscope* digital dapat mengambil beberapa *samples point* sebuah sinyal pada *real time sampling mode* dan kemudian menggunakan interpolasi. Interpolasi adalah sebuah teknik proses untuk estimasi bentuk *waveform* berdasarkan beberapa *points*.
- *Oscilloscope* digital dapat membangun gambar dari *waveform* atas waktu, selama sinyal berulang secara sendirinya (*equivalent-time sampling mode*).

2.4.2 *Real-Time Sampling with Interpolation*

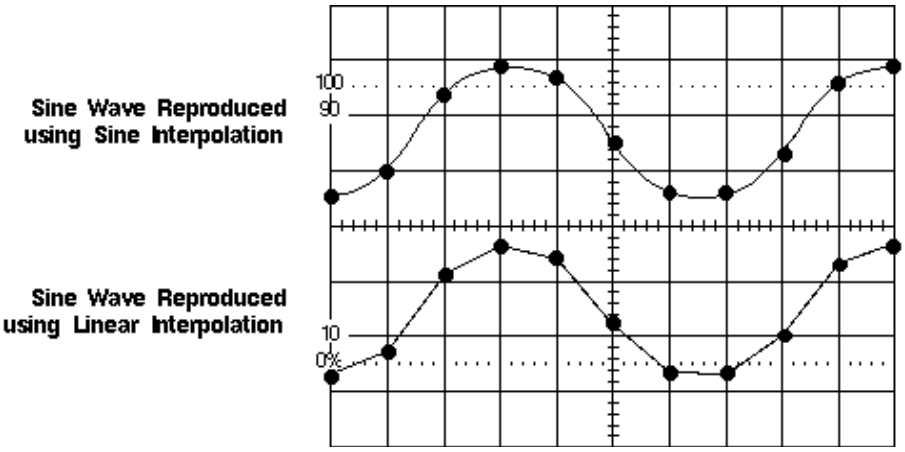
Oscilloscope digital menggunakan *real-time sampling*. Dalam *real-time sampling*, *oscilloscope* mengambil *samples* semampunya di saat ada sinyal. (lihat gambar 2.19).



Gambar 2.19. *Real-time Sampling*
"*Oscilloscope Tutorial*." Department of Computer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/oscilloscope.html>.

Oscilloscope digital menggunakan interpolasi untuk menampilkan sinyal yang sangat cepat sehingga hanya mampu mendapatkan beberapa *sample points*. Interpolasi menghubungkan antara titik-titik.

Interpolasi linier menghubungkan *sample points* dengan garis lurus. *Sine interpolation* menghubungkan *sample points* dengan kurva (lihat gambar 2.20). Dengan proses ini , sinyal yang ter-*sampling* hanya beberapa kali tiap siklus dapat secara akurat ditampilkan.



Gambar 2.20. *Linear and Sine Interpolation*
"Oscilloscope Tutorial." Department of Coumputer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/oscilloscope.html>.

2.5. *WAVEFORMAT*

Struktur *waveformat* adalah *format* dari *waveform-audio data*. Hanya *format* informasi yang umum untuk semua *format* data *waveform-audio* yang disertakan pada struktur tersebut. Struktur ini diwakilkan dengan struktur *WaveFormatEX*. Pada *waveformat* mempunyai *member* yang harus di-*set* atau ditentukan isinya dalam penggunaannya, meliputi :

- *wFormatTag*

  Tipe *format Waveform-audio*. *Waveform-audio format type. Format tags* telah teregistrasi dengan Microsoft *Corporation* dengan banyak *compression algorithms*. Daftar lengkap dari *format tags* dapat ditemui di *header file* MMREG.H.

- *nChannels*

  Jumlah *channels* dalam *waveform-audio data* yang digunakan, satu *channel* untuk mono dan dua *channel* untuk stereo data.

- *nSamplesPerSec*

  Sample rate, dalam *samples per second* (*hertz*), yang mana tiap *channel* harus memainkan atau merekam. Jika *wFormatTag* adalah *Wave_Format_PCM*, maka pada umumnya nilai dari *nSamplesPerSec* adalah 8kHz, 11.025kHz, 22.05kHz, dan 44.1kHz. Untuk *format* non-PCM, maka nilai-nilai tersebut

harus diperhitungan menurut manufaktur dari spesifikasi *format tag* yang digunakan.

- *nAvgBytesPerSec*

  Rata-rata data *transfer rate* yang dibutuhkan, dalam *bytes per second*, untuk *format tag*. Jika *wFormatTag* adalah *Wave_Format_PCM*, maka *nAvgBytesPerSec* harus sama dengan jumlah *nSamplesPerSec* dan *nBlockAlign*. Untuk *format* non-PCM, maka nilai-nilai tersebut harus diperhitungan menurut manufaktur dari spesifikasi *format tag* yang digunakan. *Playback* dan *record software* dapat mengestimasi besar *buffer* dengan menggunakan *nAvgBytesPerSec*.

- *nBlockAlign*

  Block Alignment, dalam *bytes*. *Block alignment* adalah minimum unit data untuk tipe *format wFormatTag*. Jika *wFormatTag* adalah *Wave_Format_PCM*, maka *nbBlockAlign* harus sama dengan jumlah hasil dari *nChannels* dan *wBitsPerSample*, dibagi oleh 8 (*bits per byte*). Untuk *format* non-PCM, maka nilai-nilai tersebut harus diperhitungan menurut manufaktur dari spesifikasi *format*tag yang digunakan. *Playback* dan *record software* harus memproses sebuah pengalian dari *nBlockAlign bytes* dari data pada saat saat. Data ditulis dan dibaca dari sebuah *device* harus selalu mulai dari awal *block*, maka tidak diperbolehkan untuk memulai *playback* PCM data di tengah-tengah dari *sample*.

- *wBitsPerSample*

  Bits per sample untuk tipe *format wFormatTag*. Jika *wFormatTag* adalah *format Wave_Format_PCM*, maka *wBitsPerSample* sama dengan 8 atau 16. Untuk *format* non-PCM, maka nilai-nilai tersebut harus diperhitungan menurut manufaktur dari spesifikasi *format tag* yang digunakan.

- *cbSize*

  Size (ukuran), dalam *bytes*, dari *extra format* informasi yang berada di akhir struktur *waveformatEX*. Informasi ini dapat digunakan oleh *format* non-PCM untuk menyimpan atribut-atribut *extra* untuk *wFormatTag*. Jika tidak ada informasi extra dibutuhkan oleh *wFormatTag*, maka *cbSize* harus di-*set* “0” (nol). Hanya pada *waveformat*_PCM hal ini diabaikan.

2.6. *WAVE File*

Wave file pada dasarnya merupakan metode penyimpanan suara dalam bentuk digital. *Wave file* itu sendiri tidak berdiri sendiri, karena dari pihak Microsoft selaku pembuat sistem membuat standard *RIFF* sebagai bagian induknya. Jadi *wave* merupakan bagian daripada *RIFF* Microsoft yang berfungsi sebagai penyimpanan *multimedia file*. Sebuah *file wave* dimulai dengan sebuah *file header* dan diikuti oleh serangkaian data *chunk*. Ada dua jenis *format* penyimpanan *waveform* pada *wave file*, yaitu: *PCM* dan *ADPCM*. Sampai saat ini kebanyakan *wave file* masih menggunakan *format PCM* (*Pulse Code Modulation*).

Berikut struktur *wave file* selengkapnya:

[*ChunkID*] [*ChunkSize*] [*Format*]

- [*ChunkID*] → 4 *byte*

  Berisi karakter R, I, F dan F

- [*ChunkSize*] → 4 *byte*

  Berisi panjang data selain [*ChunkID*] dan [*ChunkSize*]

- [*Format*]

  Chunk data dari *file*. Mempunyai bentuk sebagai berikut :

  [*Format*] = [*FormatID*] [*FormatChunk*] [*DataChunk*]

  - [*FormatID*] → 4 *byte*

    Berisi data berupa karakter W,A,V dan E

  - [*FormatChunk*]

    Berisi data yang menentukan *format* dari data yang terdapat dalam data *chunk.* Mempunyai bentuk sebagai berikut :

    [*FormatChunk*] = [*SubChunk1ID*] [*SubChunk1Size*] [*AudioFormat*] [*NumChannels*] [*SampleRate*] [*ByteRate*] [*BlockAlign*] [*BitsPerSample*]

    - [*SubChunk1ID*] → 4 *byte*

      Berisi data berupa karakter “fmt” dan spasi yang menandai bahwa blok tersebut merupakan *format chunk*

    - [*SubChunk1Size*] → 4 *byte*

      Adalah panjang dari data berikutnya dalam *format chunk*

    - [*AudioFormat*] → 2 *byte*

Menunjukkan kategori dari *format wave file*, biasanya menggunakan PCM (*Pulse Code Modulation*) *format* yaitu 01[1]

- [*NumChannels*] → 2 *byte*

Untuk menentukan suara yang dihasilkan *stereo* atau *mono*, 1 untuk *mono* dan 2 untuk *stereo*

- [*SampleRate*] → 4 *byte*

Menunjukkan *sampling rate* atau frekuensi yang digunakan dalam proses perekaman data suara. Frekuensi yang disediakan untuk data suara adalah 11025 Hz, 22050 Hz atau 44100 Hz.

- [*ByteRate*] → 4 *byte*

Menunjukkan rata-rata jumlah *byte* perdetik data yang harus dikirimkan.

- [*BlockAlign*] → 2 *byte*

Menunjukkan jumlah *byte* yang digunakan untuk membentuk sebuah sample tunggal.

- [*BitsPerSample*] → 2 *byte*

Berisi nilai yang menentukan berapa *bit* yang digunakan dalam sebuah sample, yaitu delapan *bit* atau enam belas *bit* per *sample*

- [*DataChunk*]

Berisi data sesungguhnya dari *wave file*, *format*-nya sesuai dengan [*FormatChunk*]. Mempunyai bentuk sebagai berikut :

[*DataChunk*] = [*SubChunk2ID*] [*SubChunk2Size*] [*Data*]

- [*SubChunk2ID*] → 4 *byte*

Berisi karakter d,a,t, dan a yang menandai data *chunk*

- [*SubChunk2Size*] → 4 *byte*

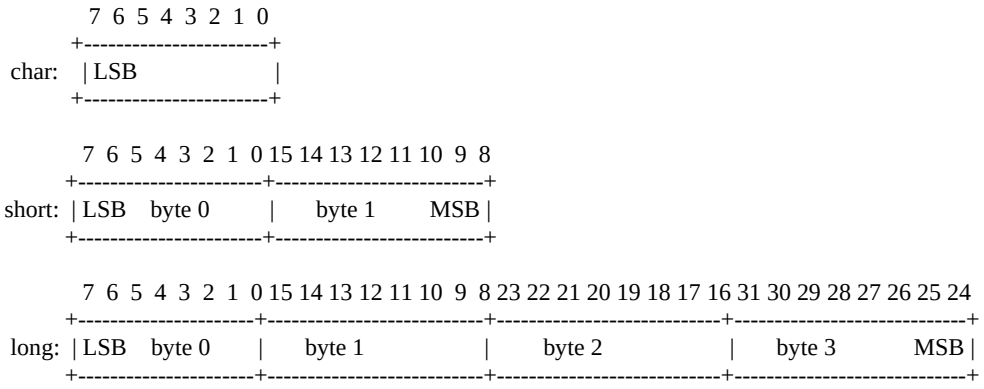
Menunjukkan panjang data yang akan mengikuti

- [Data]

Data *waveform*.

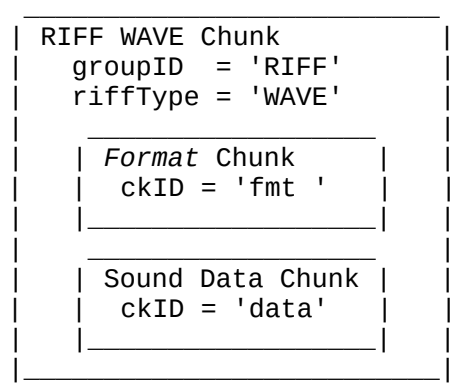
Semua data disimpan dalam 8-bit. Susunan penyimpanan diawali oleh "LSB" *least significant byte* terlebih dahulu. Susunan penyimpanan data dapat

digambarkan sebagai berikut (ditunjukkan pada gambar 2.21 dengan bit *numbers* di atasnya, “LSB” dan “MSB” *most significant byte*):



Gambar 2.21. Ilustrasi Susunan Penyimpan Data
Gilbert, Scoot. *WAVE File Format*. Cleartel Communications. 2003 <http://www.borg.com/~jglatt/tech/wave.htm>.

Sebuah *WAVE file* adalah kumpulan sebuah angka dari tipe-tipe *chunks* yang berbeda. Semua *WAVE* harus ada sebuah *format* (“fmt”) *chunk* dibutuhkan berisi parameter yang menjelaskan *waveform sample rate* dan *data chunk* yang berisikan *data waveform*. 2 *chunk* tersebut adalah *chunk* penting yang dibutuhkan dalam sebuah *WAVE file*. Dan untuk *chunk* lainnya adalah pilihan. Berikut adalah sebuah contoh tampilan, *WAVE file* minimal yang berisikan sebuah *single WAVE* terdiri dari 2 *chunks* yang dibutuhkan, *format* dan *data chunk*.

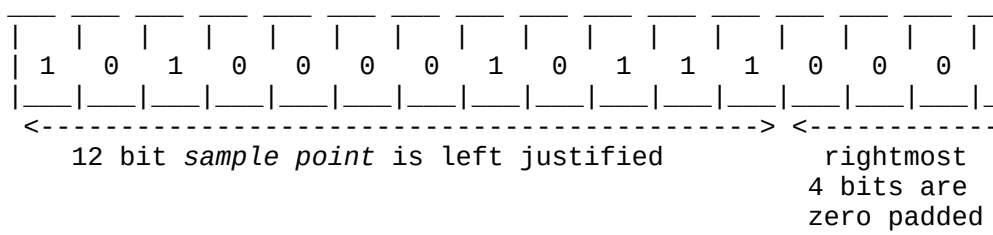


Gambar 2.22. *WAVE File* Minimal Berisi *Single WAVE* (*Format* dan Data *Chunk*)
Gilbert, Scoot. *WAVE File Format*. Cleartel Communications. 2003 <http://www.borg.com/~jglatt/tech/wave.htm>.

2.6.1. *Sample Points* dan *Sample Frames*

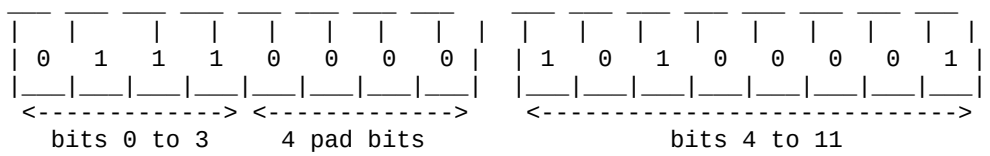
Sebuah *sample point* adalah nilai yang mewakilkan sebuah *sample* dari sebuah *sound* pada waktu tertentu. 8-bit *samples* disimpan dalam *range* 0 sampai 255. 16-bit *samples* disimpan sebagai *2's complement* integer dengan *range* 32768 sampai 327687.

Jika ADC menghasilkan *sample point* dari 1 sampai 8 bit, sebuah *sample point* harus disimpan dalam *WAVE* sebagai 8-bit byte. Jika ADC menghasilkan sebuah *sample point* dari 9-16 bit, sebuah *sample point* disimpan dalam *WAVE* sebagai 16-bit *word*. Jika ADC menghasilkan sebuah *sample point* dari 17-24 bits, sebuah *sample point* disimpan dalam WAVE sebagai 3 bytes. Jika ADC menghasilkan *sample point* dari 25-32 bits, *sample point* disimpan dalam *WAVE* sebagai 32-bit *doubleword.* Untuk 12-bit, maka *sample point* harus disimpan sebagai 16-bit *word.* 12-bits itu harus teratur ke kiri sehingga tertata pada bits 4 sampai 15 dan bits 0 sampai 3 di-*set zero*.



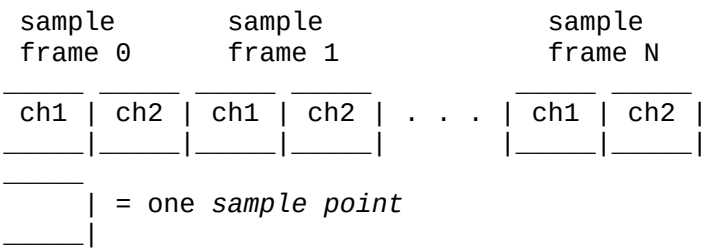
Gambar 2.23. Penyimpanan *Sample Point* Sebagai 16-bit *word*
Gilbert, Scoot. *WAVE File Format*. Cleartel Communications. 2003 <http://www.borg.com/~jglatt/tech/wave.htm>.

Karena *WAVE format* menggunakan "*intel little endian byte order*", maka "LSB" disimpan pertama kali pada *wave file* (gambar 2.24).



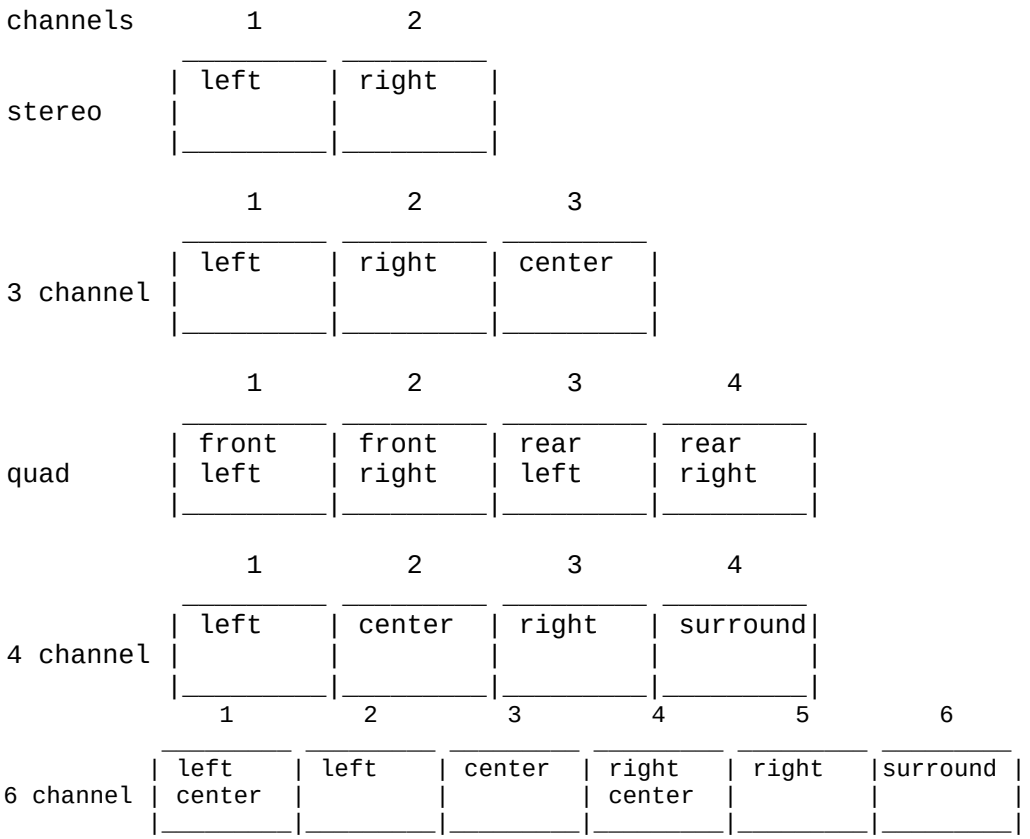
Gambar 2.24. "LSB" Pada Awal *Wave File*
Gilbert, Scoot. *WAVE File Format*. Cleartel Communications. 2003 <http://www.borg.com/~jglatt/tech/wave.htm>.

Untuk *multichannel sounds* (*stereo waveform*), *single sample point* dari tiap *channel* bergantian, jadi menyimpan sebuah *sample point* dari *left channel* dahulu, lalu menyimpan sebuah *sample points* untuk *right channel* dan seterusnya.



Gambar 2.25. *Single Sample Point* Pada *Multichannel Sounds*
Gilbert, Scoot. *WAVE File Format*. Cleartel Communications. 2003 <http://www.borg.com/~jglatt/tech/wave.htm>.

Untuk *monophonic waveform*, *sample frame* berisi *single point* dan tidak dikumpulkan secara bergantian. Berikut ilustrasi penyimpanan antar *sample frame* untuk beberapa *channel*.



Gambar 2.26. *Sample point* Pada *Monophonic Sound*
Gilbert, Scoot. *WAVE File Format*. Cleartel Communications. 2003 <http://www.borg.com/~jglatt/tech/wave.htm>.

Sample point antar sebuah *sample frame* dipaketkan bersama dan tidak ada *bytes* yang tidak berguna antar mereka (tidak ada *pad bytes*).

2.6.1 *Format* PCM

Pulse Code Modulation (PCM) dan *Adaptive Delta Pulse Code Modulation* (ADPCM) adalah sub kelas dari *format file* Microsoft *Waveform* (".WAV"). Pada PCM, data untuk *file* ".WAV" disimpan dengan menggunakan *linier samples.*

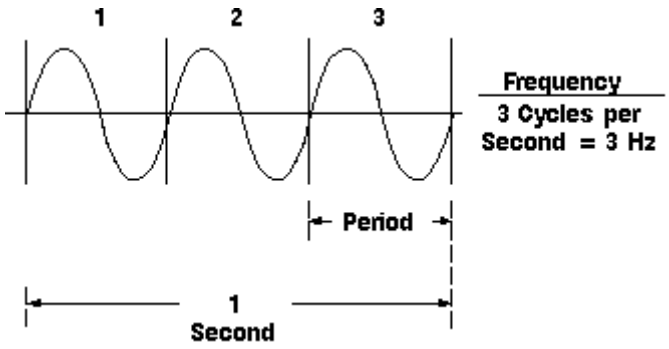
PCM dan ADPCM adalah teknik untuk penyimpanan data *audio* analog dalam *format* digital file Microsoft ".WAV". Adapun metode penyimpanan lain seperti "Mu-Law", "A-Law", *Transform Coding*, *CELP*, dan lain-lain tetapi tidak di-*support* oleh *Windows Sound System* 1.0.

PCM bekerja dengan mengambil *dicrete samples* pada interval ganjil (disebut sebagai *sampling rate*). Pada umumnya intervalnya adalah 11kHz, 22kHz, dan 44kHz. Semakin tinggi *sampling rate*, semakin bagus pula penyajiannya dari *wave* analog asli dan lebih baik pula kualitas suaranya. Tiap *samples* adalah sebuah angka *real* dengan resolusi mulai +1.0 dari nilai *full-scale* dampai -1.0 dari nilai *full-scale*. Karena data tersebut harus disimpan sebagai angka digital yang presisi, data dibulatkan ke 16-bit PCM ataupun 8-bit PCM, pada umumnya disebut 8 dan 16-bit *samples*. 16-bit data mempunyai resolusi lebih tinggi, jadi *digital waveform sound* lebih baik. 8-bit PCM mempunyai resolusi yang kurang, menyebabkan keterbatasan histerisis pada *waveform*. Tetapi juga lebih membutuhkan ruang *disc* yang lebih sedikit

ADPCM melakukan penyimpanan *waveform* dengan menggunakan 4-bits per *sample*. ADPCM harus di-*compress* atau *decompressed* dari atau ke bentuk PCM karena *Windows sound system hardware* hanya mengenali 8/16-bit PCM. Untuk melakukan ADPCM, komputer harus memiliki *Audio Compression Manager* (ACM). Adanya proses *compress* atau *decompressed* tersebut membuat data mengalami distorsi terutama pada frekuensi tinggi.

2.7. *Waveform*

Sebuah sinyal yang berulang mempunyai frekuensi. Frekuensi sama dengan nilai berapa kali sebuah sinyal berulang dalam waktu satu detik (*cycles per second*). Sinyal berulang juga mempunyai periode. Periode adalah jumlah waktu yang diperlukan sebuah sinyal untuk melalukan satu siklus gelombang penuh. Periode dan frekuensi saling berkaitan yaitu 1/periode sama dengan frekuensi dan 1/frekuensi sama dengan periode. Berikut sebuah contoh gelombang sinus (*sine wave*) pada gambar 2.27. mempunyai frekuensi 3Hz dan sebuh periode 1/3 detik.



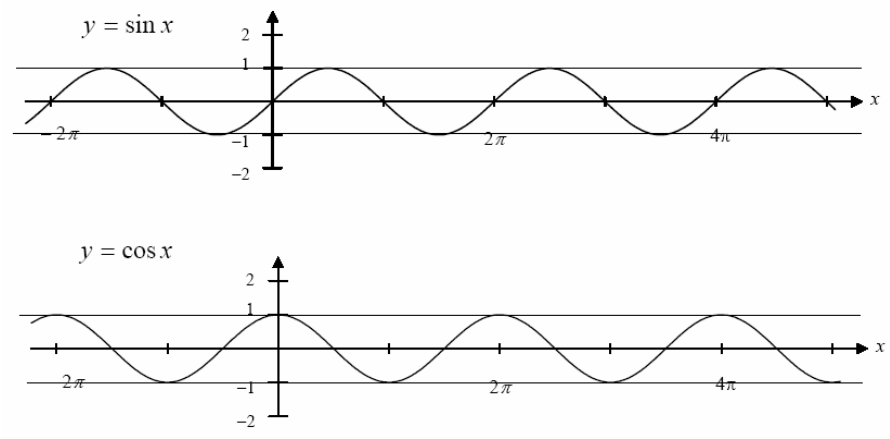
Gambar 2.27. *Frequency and Period*

"Oscilloscope Tutorial." Department of Computer Science. Trinity College, Dublin. 13 April 2005 <https://www.cs.tcd.ie/courses/baict/bac/jf/labs/scope/terminology.html>.

2.7.1. Tegangan

Tegangan adalah jumlah potensial listrik atau sejenis kekuatan sinyal antara dua titik dalam rangkaian. Biasanya salah satu dari titik tersebut adalah *ground* (*zero volts*) tetapi tidak selalu, bisa juga mengukur tegangan dari maksimum *peak* ke minimum *peak* dari *waveform*, dikenal sebagai *peak-to-peak voltage*. Amplitudo adalah teganan maksimum dari sinyal yang diukur dari *ground* atau *zero volts*. *Waveform* yang ditunjukkan pada gambar 2.28 mempunyai amplitudo 1 Volt dan tegangan *peak-to-peak* 2 Volts.

2.7.2. Sinusoidal



Gambar 2.28. Grafik Sinus dan Cosinus

Grafik di atas berisolasi antara 1 dan -1. Panjang atau periode satu gelombang penuhnya adalah 2phi. Definisi secara lengkap dari sinus dan cosinus adalah

$$y = k\sin(a\,x - \phi) + C \tag{2.6}$$

$$y = k\cos(a\,x - \phi) + C \tag{2.7}$$

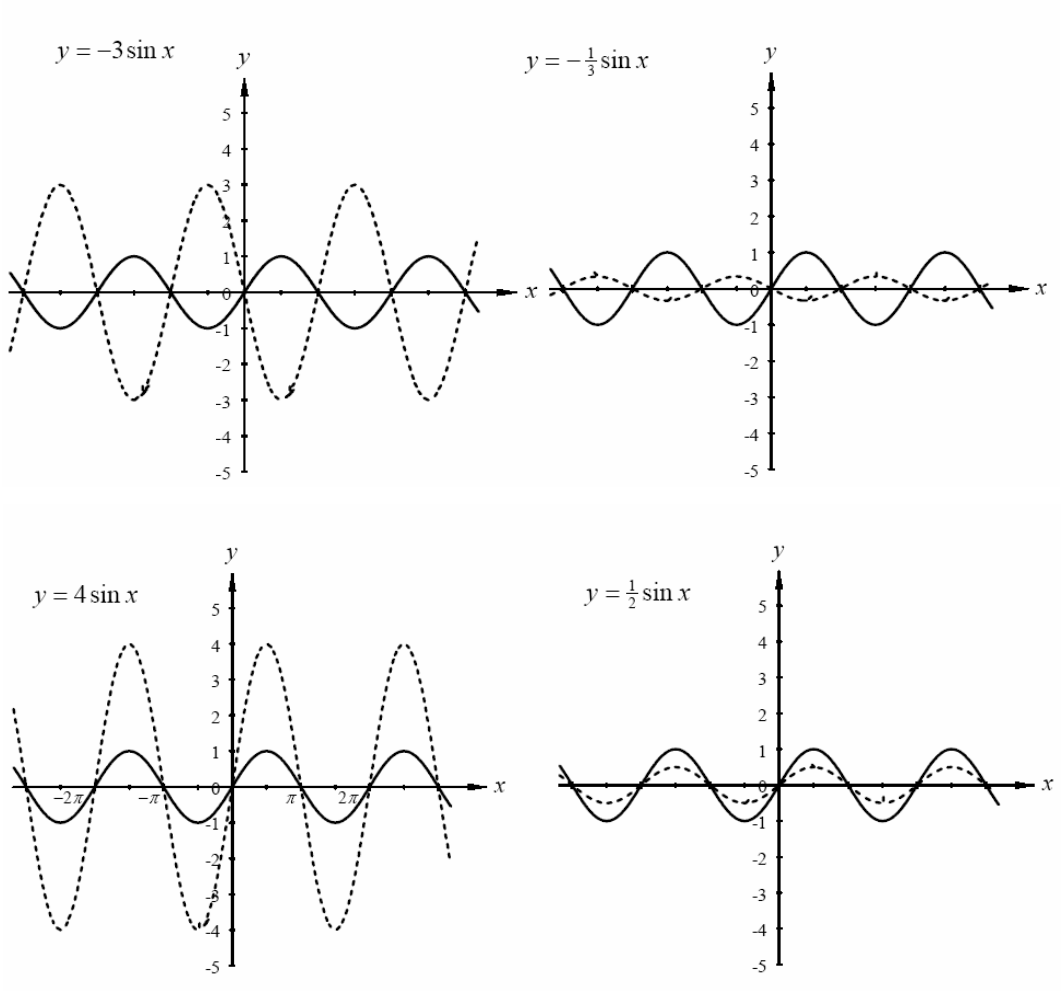
Amplitido dari sebuah grafik sinusoidal sama dengan tinggi maksimal dari sebuah gelombang dari sumbu x. Periode dari fungsi sinusoidal adalah jarak sebuah grafik untuk menyelesaikan satu gelombang penuh, dengan rumusan

$$P = \frac{2\pi}{a} \tag{2.8}$$

Angular (Circular) Frekuensi sebuah sinusoidal pada rumusan 2.8 ditunjukkan oleh a, sebuah nilai dari siklus penuh pada jarak horizontal oleh 2phi. Frekuensi linier (f) adalah kebalikan dari periode maka

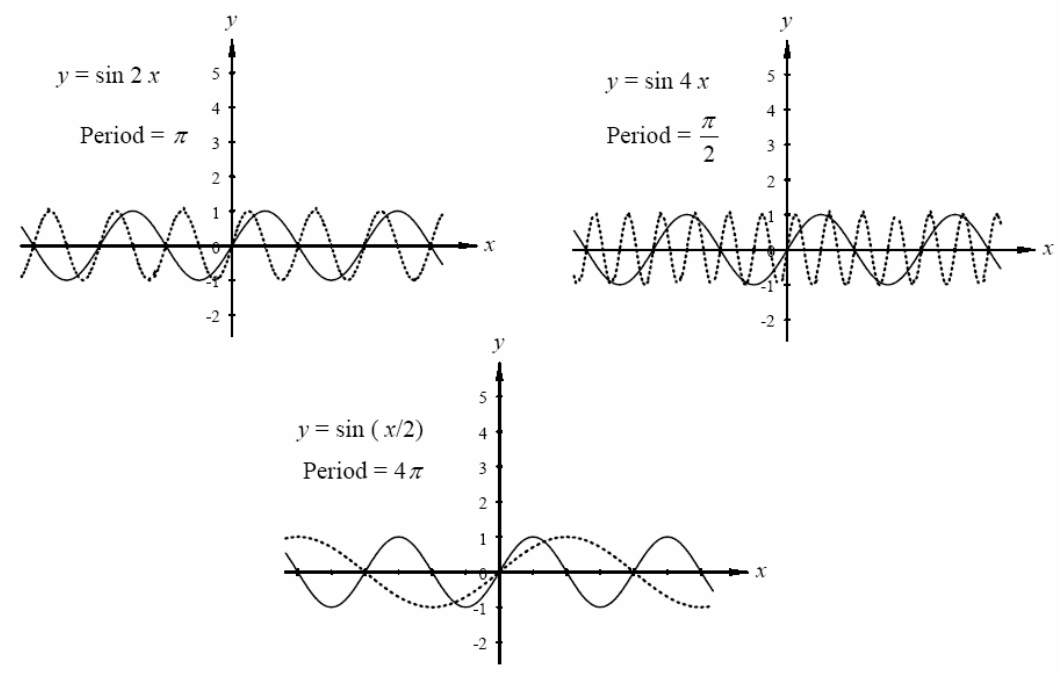
$$f = \frac{a}{2\pi} \tag{2.9}$$

Jika x mewakilkan waktu, maka f adalah banyaknya gelombang penuh per detik. *Phase* pada gelombang sinusoidal adalah titik dimana sebuah gelombang memulai siklusnya ketika x = 0. Hal ini merepresentasikan perubahan atau pergeseran pada sumbu x. Garis horizontal y = C disebut garis tengah (*center line*) di lokasi gelombang sedang berosilasi. Perubahan amplitudo ditunjukkan pada gambar 2.29.



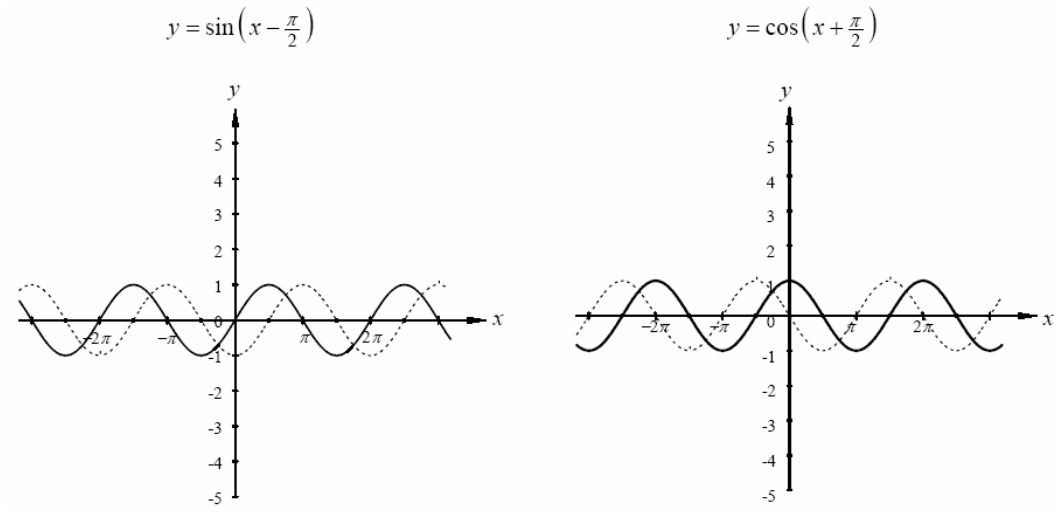
Gambar 2.29. Grafik Sinusoidal Dengan Perubahan Amplitudo

Perubahan frekuensi merupakan horizontal scaling pada perumusan dari y = sin x atau y = cos x menjadi y = sin (a x) dan y = cos (a x). Perubahan tersebut menunjukan berapa banyak siklus yang terjadi pada suatu interval.



Gambar 2.30. Grafik Sinusoidal y = sin (ax) dengan Perubahan Frekuensi

Perubahan fasa melibatkan pergeseran horizontal, seperti pada gambar 2.31.



Gambar 2.31. Grafik Sinusoidal dengan Perubahan Fasa atau Pergeseran Horizontal