BAB II
LANDASAN TEORI

2.1. Komunikasi Data

Komunikasi data merupakan proses pentransmisian data secara elektronik melalui media berupa kabel maupun tanpa kabel (nirkabel). Transmisi data adalah proses yang terjadi antara *transmitter* dan *receiver* melalui suatu media transmisi yang diklasifikasikan sebagai *guided* media, mengacu pada media transmisi fisik berupa kabel, dan *unguided* media, mengacu pada media transmisi nirkabel. Tujuan fundamental dari sebuah sistem komunikasi data adalah pertukaran data secara elektronik antara dua pihak[3].

Secara umum, model komunikasi data dapat diilustrasikan seperti pada Gambar 2.1 yang merupakan sebuah model sistem komunikasi data yang telah disederhanakan.



Gambar 2.1. Model Komunikasi Data[3]

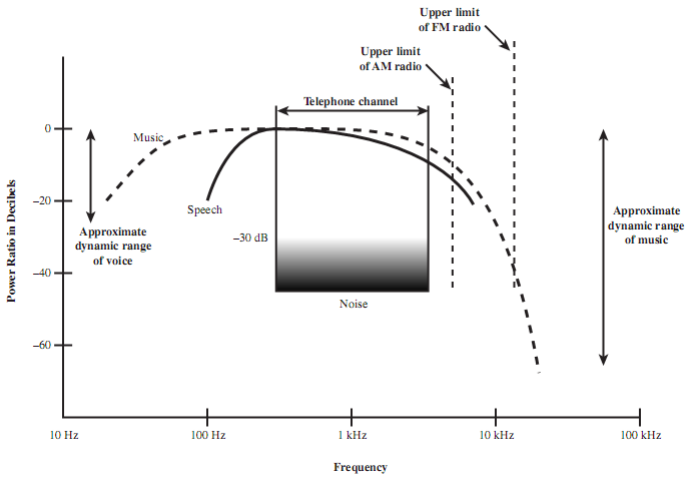
Elemen sumber pada Gambar 2.1 merupakan suatu perangkat yang membangkitkan data untuk ditransmisikan. Informasi yang diterima akan diubah kedalam bentuk data-data digital dengan tingkat ketelitian tertentu sesuai dengan kebutuhan. *Transmitter* akan melakukan pengubahan dan pengkodean informasi untuk membangkitkan suatu gelombang elektromagnetik yang dapat ditransmisikan melalui suatu media transmisi. Media transmisi dapat berupa jalur-jalur kabel (*wireline*) atau nirkabel (*wireless*) sebagai penghubung antara sumber dan tujuan. *Receiver* akan menerima sinyal dari sistem transmisi data dan mengubah sinyal tersebut kedalam bentuk yang dapat ditangani oleh perangkat tujuan. Bentuk informasi yang diterima pada perangkat tujuan bisa jadi tidak sama dengan bentuk infomasi pada sumber. Sistem penerimaan dimungkinkan untuk melakukan pengubahan terhadap data pada penerima kedalam bentuk yang berbeda dengan data dari sumber.

2.1.1. Transmisi Data Analog dan Digital

Istilah analog dan digital masing-masing berhubungan pada suatu entitas yang bersifat kontinu dan diskrit. Pada sistem komunikasi data, istilah tersebut setidaknya digunakan dalam tiga konteks, antara lain:

1. Data sebagai entitas pembawa informasi, dan sinyal sebagai representasi data dalam bentuk gelombang elektromagnetik atau gelombang listrik.
2. Pensinyalan yang merupakan propagasi fisik sinyal pada sebuah medium.
3. Transmisi, yaitu pengkomunikasian data dengan melakukan propagasi dan pemrosesan sinyal.

Data analog memiliki nilai kontinu dalam suatu interval. Contohnya, suara yang secara kontinu memiliki intensitas pola yang bervariasi. Suara merupakan bentuk data analog yang cukup umum karena pada bentuk gelombang suara akustik, suara masih dapat diterima oleh indera manusia secara langsung. Sedangkan data digital memiliki nilai diskrit seperti teks pada kode ASCII di komputer.



Gambar 2.2. Spektrum Akustik Suara dan Musik[3]

Gambar 2.2 menunjukkan spektrum akustik untuk suara manusia dan musik secara umum. Suara memiliki komponen frekuensi pada rentang 100 Hz hingga 7 kHz. Namun, frekuensi suara yang berada dibawah 600 atau 700 Hz terlalu rendah untuk dapat diterima oleh telinga manusia.

Dalam sebuah sistem komunikasi, data dipropagasikan dari satu tempat ke tempat lain dengan menggunakan sinyal elektromagnetik. Pada sinyal analog data

ditransmisikan secara kontinu dengan melakukan pengubahan terhadap gelombang elektromagnetik yang dipropagasikan melalui berbagai media. Sedangkan pada sinyal digital, data ditransmisikan sebagai urutan pulsa tegangan melalui media kabel; misalnya konstanta level tegangan positif sebagai nilai biner 0 dan konstanta level tegangan negatif sebagai nilai biner 1.

Sinyal digital dan sinyal analog dapat ditransmisikan melalui media transmisi sebagai sebuah fungsi pada suatu sistem transmisi. Pada transmisi analog, sinyal analog ditransmisikan tanpa memperhatikan isi dan sinyal dapat merepresentasikan data analog atau data digital. Namun, sinyal analog akan mengalami atenuasi setelah menempuh jarak tertentu sehingga akan dibutuhkan amplifier sebagai penguat energi pada sinyal (dan *noise*) untuk jarak yang lebih jauh. Pada data analog seperti suara, distorsi masih dapat ditolerir dan informasi masih dapat diterima. Namun pada data digital, distorsi akan menyebabkan terjadinya *error*.

Transmisi digital sangat terkait dengan isi. Sinyal digital dapat ditransmisikan pada jarak yang sangat terbatas sebelum antenuasi, *noise*, dan pelemahan sinyal menganggu integritas data. *Repeater* pada sistem transmisi digital digunakan untuk memberikan kemampuan transmisi data pada jarak yang lebih jauh. *Repeater* akan menerima sinyal digital, melakukan recovery terhadap pola data 1 dan 0, dan melakukan pentransmisian ulang terhadap sinyal, sehingga atenuasi dapat ditanggulangi.

2.1.2. Kapasitas Kanal

Kapasitas kanal merupakan laju kecepatan maksimum untuk mentransmisikan data pada suatu jalur komunikasi atau kanal dalam kondisi tertentu. Pada suatu sistem komunikasi, keseluruhan kanal yang digunakan memiliki bandwith yang terbatas. Keterbatasan tersebut muncul dari sifat fisis suatu media transmisi, atau pembatasan bandwith pada transmitter untuk menghindari adanya interferensi terhadap sumber transmisi lain. Sehingga, pada suatu kanal yang dianggap *noise-free* sekalipun, bandwith dari suatu sinyal merupakan sebuah keterbatasan untuk mentransmisikan data pada laju kecepatan maksimumnya.

Berdasarkan pada keterbatasan ini, Nyquist memformulasikan bahwa pada lebar bandwith B, kecepatan sinyal tertinggi yang dapat ditransmisikan adalah sebesar 2B. Pembatasan ini dilakukan untuk menghindari adanya interferensi yang disebabkan oleh distorsi *delay*. Jika sinyal yang ditransmisikan merupakan sinyal biner, maka laju kecepatan data yang dapat didukung oleh B Hz adalah sebesar 2B bps. Pada pensinyalan multilevel, formula Nyquist untuk kapasitas kanal, C, akan menjadi:

$$\mathrm{C} = 2\mathrm{B}\log_2\mathrm{M} \tag{2.1}$$

dengan M merupakan jumlah sinyal diskrit atau level tegangan.

Pada besar bandwith yang diberikan, laju kecepatan data dapat ditingkatkan dengan meningkatkan jumlah elemen sinyal yang berbeda. Namun, hal ini akan menambah beban pada *receiver* karena harus membedakan satu dari M kemungkinan elemen sinyal. Noise dan pelemahan lain pada jalur transmisi akan membatasi nilai praktis M.

Keberadaan noise dapat mengakibatkan pengubahan satu atau lebih bit-bit data sehingga dapat menyebabkan terjadinya *error*. Jika laju kecepatan data meningkat, maka pada tingkat *noise* tertentu, juga akan meningkatkan jumlah bit yang mengalami *error*. Sehingga, dibutuhkan kekuatan sinyal yang lebih besar agar data yang ditransmisikan masih dapat diterima oleh *receiver* dengan baik pada tingkatan *noise* tertentu. Hal ini diformulasikan lebih lanjut oleh Claude Shannon. Perbandingan antara daya pada sinyal dengan daya yang terkandung pada *noise* dan muncul pada titik tertentu pada sistem transmisi didefinisikan sebagai *signal-to-noise ratio* (SNR, atau S/N). Rasio ini sering dituliskan dalam satuan desibel:

$$\mathrm{SNR_{dB}} = 10\log_{10}\frac{\text{daya sinyal}}{\text{daya noise}} \tag{2.2}$$

Persamaan (2.2) menunjukkan bahwa kuantitas, dalam desibel, daya sinyal diharapkan lebih besar dari level *noise*. Nilai SNR yang besar menunjukkan kualitas sinyal yang tinggi dan lebih sedikit jumlah *intermediate repeater* yang dibutuhkan.

Signal-to-noise ratio merupakan perbandingan yang penting pada transmisi data digital karena menentukan batas atas untuk laju kecepatan data

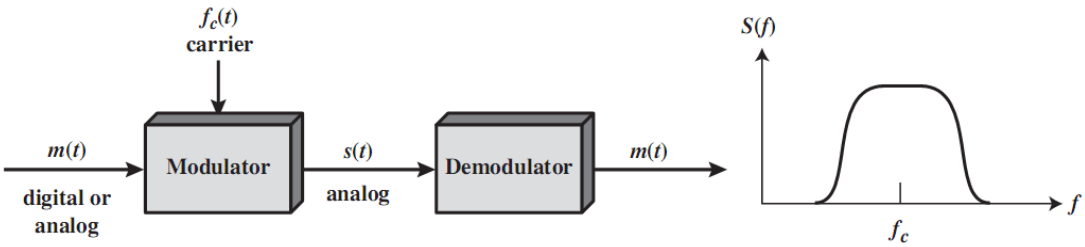
yang dapat dicapai. Kapasitas kanal maksimum, dalam bit per detik (bps), ditentukan oleh Shannon pada Persamaan (2.3) berikut.

$$C = B \log_2(1 + SNR) \tag{2.3}$$

dimana C adalah kapasitas kanal dalam satuan bit per detik (bps) dan B adalah bandwith kanal dalam Hertz. Formula Shannon merepresentasikan kapasitas kanal maksimum yang dapat dicapai secara teoritis. Namun, pada penerapan praktis, hanya pada tingkatan yang lebih rendah yang dapat dicapai.

2.2. Teknik Modulasi

Modulasi merupakan proses pengkodean data sumber ke dalam sinyal pembawa dengan frekuensi f_c. Keseluruhan teknik modulasi melibatkan tiga parameter dasar dalam domain frekuensi yaitu: amplitudo, frekuensi, dan fasa.



Gambar 2.3. Teknik Modulasi[3]

Gambar 2.3 menunjukkan proses modulasi sinyal *m(t)* pada frekuensi pembawa $f_c(t)$ menjadi sinyal *s(t)* untuk kemudian didemodulasi kembali menjadi sinyal *m(t)*. Sinyal input *m(t)* dapat berupa analog maupun digital dan disebut sebagai sinyal yang dimodulasi atau sinyal *baseband*. Hasil dari proses modulasi pada sinyal pembawa disebut sebagai sinyal termodulasi *s(t)*. Pada Gambar 2.3, sinyal *s(t)* merupakan sinyal *bandlimited* (*bandpass*).

2.2.1. *Amplitude Shift Keying* (ASK)

Amplitude Shift Keying (ASK) merupakan suatu bentuk modulasi yang merepresentasikan data sebagai perubahan amplitudo pada sinyal pembawa. Level amplitudo digunakan untuk merepresentasikan logika biner 0 atau 1 seperti pada Gambar 2.4a. Pada sinyal yang termodulasi, nilai logika 0 direpresentasikan sebagai ketiadaan sinyal pembawa.

$$s(t) = \begin{cases} A\cos(2\pi f_c t) & biner\ 1 \\ 0 & biner\ 0 \end{cases} \quad (2.4)$$

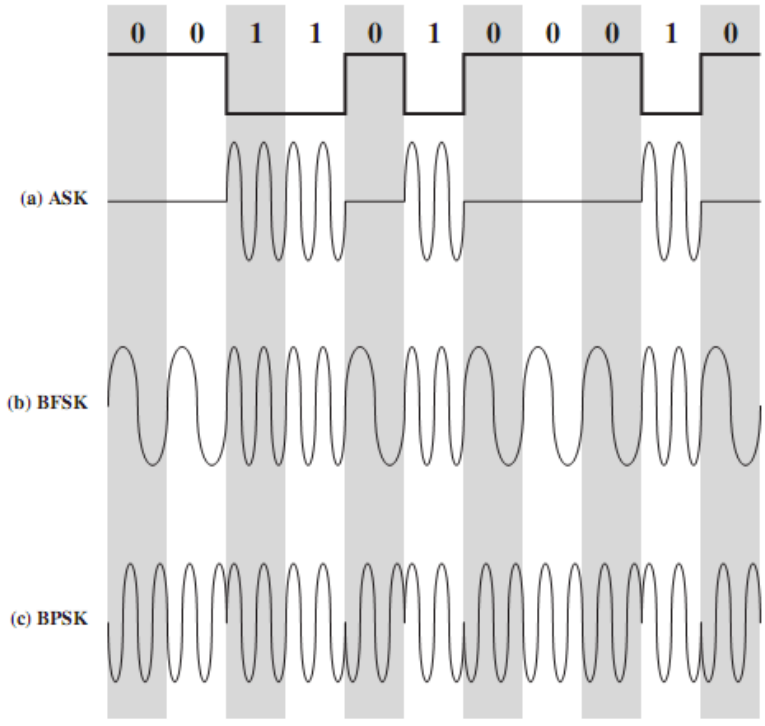
dengan $s(t)$ fungsi sinyal termodulasi pada saat t, A besar amplitudo sinyal, dan f_c merupakan frekuensi pembawa.

2.2.2. *Frequency Shift Keying* (FSK)

Frequency Shift Keying (FSK) merupakan skema modulasi frekuensi dengan perubahan pulsa pada frekuensi sinyal pembawa untuk mentransmisikan informasi digital. Bentuk yang paling umum digunakan dari teknik modulasi FSK adalah *Binary* FSK (BFSK), dengan dua nilai biner direpresentasikan oleh dua frekuensi yang berbeda mendekati frekuensi sinyal pembawa. Pada skema ini, nilai logika "0" disebut sebagai frekuensi "*space*" dan nilai logika "1" disebut sebagai frekuensi "*mark*". Gambar 2.4b menunjukkan contoh penggunaan BFSK pada pengiriman data digital. Sinyal termodulasi s(t) yang ditransmisikan untuk satu periode bit yaitu:

$$s(t) = \begin{cases} A\cos(2\pi f_1 t) & biner\ 1 \\ A\cos(2\pi f_2 t) & biner\ 0 \end{cases} \quad (2.5)$$

dengan f_1 dan f_2 merupakan offset dari frekuensi pembawa f_c yang sebanding namun berlawanan.



Gambar 2.4. Modulasi pada Sinyal Analog untuk Data Digital[3]

2.2.3. *Phase Shift Keying* (PSK)

Phase Shift Keying (PSK) merupakan skema modulasi digital yang akan mengirimkan data dengan mengubah, atau memodulasi, besar fasa dari sinyal pembawa seperti pada Gambar 2.4c. Skema yang paling sederhana pada modulasi PSK yaitu menggunakan dua buah fasa untuk merepresentasikan dua digit biner dan dikenal sebagai *Binary Phase Shift Keying* (BPSK). Sinyal yang ditransmisikan untuk satu periode bit yaitu:

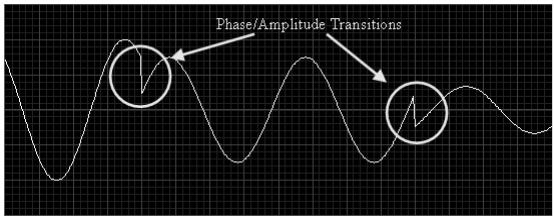
$$s(t) = \begin{cases} Acos\,(2\pi f_c t) & = Acos\,(2\pi f_c t) \quad biner\ 1 \\ Acos\,(2\pi f_c t + \pi) & = -Acos(2\pi f_c t) \quad biner\ 0 \end{cases} \tag{2.6}$$

2.2.4. *Gaussian Frequency Shift Keying* (GFSK)

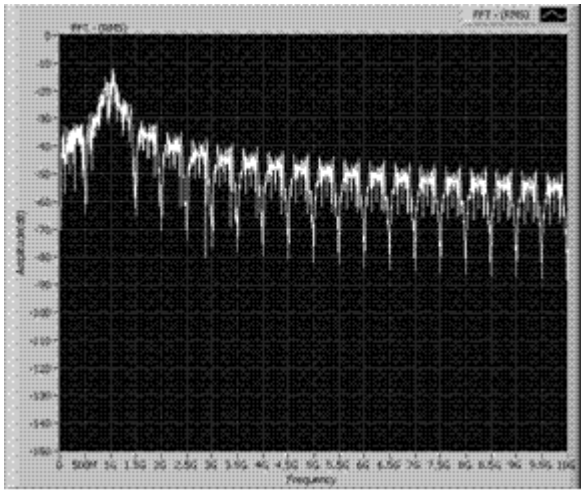
Salah satu bentuk lain dari teknik modulasi FSK adalah teknik modulasi GFSK (*Gaussian Frequency Shift Keying*). Pada teknik modulasi GFSK, sebelum sinyal *baseband* masuk ke modulator FSK, sinyal akan melewati sebuah *pulse-shape filter* yang disebut *Gaussian Filter* untuk membuat denyut sinyal menjadi lebih halus sehingga membatasi lebar spektrumnya[18]. *Pulse-shape filter* digunakan untuk memenuhi persyaratan dalam sistem komunikasi nirkabel yang salah satunya adalah untuk menghasilkan *bandlimited channel*.

Modulasi pada sebuah sinyal pembawa akan menghasilkan transisi yang konstan terhadap fasa dan amplitudo. Gambar 2.5a mengilustrasikan transisi yang terjadi tanpa ada filter dalam domain waktu. Pada Gambar 2.5a, terlihat bahwa transisi yang tajam terjadi ketika *filtering* tidak diimplementasikan.

Transisi yang tajam yang terjadi pada sinyal akan menyebabkan kemunculan komponen frekuensi tinggi dalam domain frekuensi. Gambar 2.5b menunjukkan sebuah hasil plot spektrum dari sinyal termodulasi tanpa filter pada domain frekuensi. Grafik tersebut mengilustrasikan bahwa pada kondisi tanpa filter, daya kanal pada sinyal pembawa secara signifikan melebihi lebar *bandwith* yang telah ditentukan (*out-of-band spectrum*). Pada kondisi ini, kanal akan membutuhkan daya transmisi yang sangat besar karena rentang frekuensi yang tidak terkonsentrasi, dan juga akan menyebabkan interferensi terhadap kanal yang berdekatan terutama pada sistem komunikasi *multi-channel*.



(a)

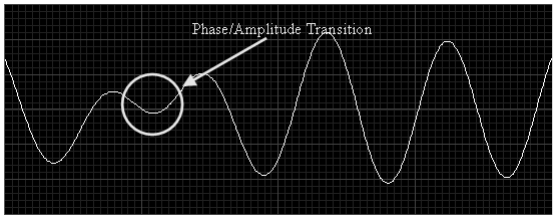


(b)

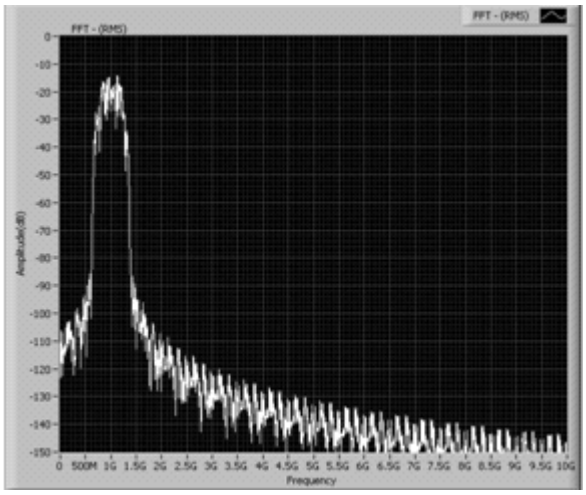
Gambar 2.5. (a) Transisi Fasa dan Amplitudo pada Sebuah Sinyal Termodulasi Tanpa Filter, (b) Bentuk Sinyal Tanpa Filter dalam Domain Frekuensi.

Penerapan sebuah *pulse-shaping filter* pada sinyal termodulasi akan menghasilkan transisi fasa dan amplitudo menjadi lebih halus dan sinyal akan dibatasi hanya pada band frekuensi tertentu. Gambar 2.6a mengilustrasikan transisi fasa dan amplitudo pada sinyal termodulasi yang terjadi secara bertahap ketika sebuah *pulse-shape filter* diimplementasikan dalam domain waktu.

Transisi fasa dan amplitudo yang terjadi secara bertahap akan menghasilkan sinyal dengan frekuensi yang terkonsentrasi hanya pada band frekuensi tertentu. Hal ini dapat dilihat pada Gambar 2.6b yang mengilustrasikan hasil plot spektrum pada sinyal termodulasi dengan menggunakan filter dalam domain frekuensi. Pada gambar tersebut, komponen frekuensi tinggi yang sebelumnya muncul telah dihilangkan sehingga mayoritas daya kanal akan terkonsentrasi hanya pada sebuah band frekuensi yang spesifik.



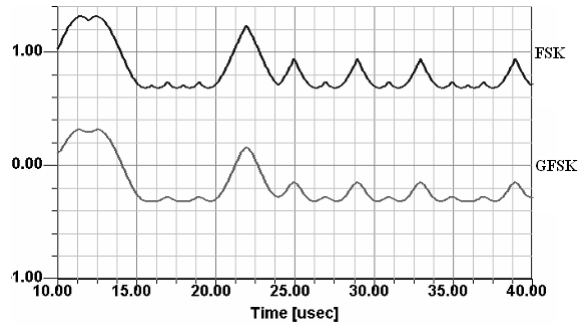
(a)



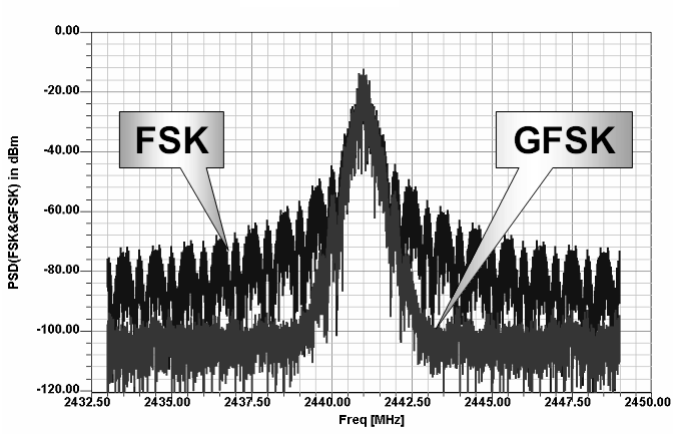
(b)

Gambar 2.6. (a) Transisi Fasa dan Amplitudo pada Sebuah Sinyal Termodulasi Dengan Filter, (b) Bentuk Sinyal Dengan Filter dalam Domain Frekuensi.

Gaussian filtering merupakan salah satu metode yang digunakan untuk mengurangi lebar spektrum. Jika dimisalkan sebuah sinyal baseband (1, -1), nilai "-1" untuk fc-fd dan nilai "1" untuk fc+fd, maka perubahan nilai dari "1" ke "-1" atau sebaliknya, dan pada perubahan bentuk sinyal yang sangat cepat akan menyebabkan terjadinya *out-of-band spectrum* yang sangat besar. Namun, jika perubahan sinyal dilakukan secara bertahap, misalnya mulai -1, -0,98, -0,93, …, 0,96, 0,99, 1, dan denyut sinyal yang lebih halus ini digunakan untuk memodulasi sinyal pembawa, maka kejadian *out-of-band spectrum* tersebut dapat dikurangi. Gambar 2.7 menunjukkan perbandingan antara bentuk spektrum dalam domain waktu (Gambar 2.7a) dan frekuensi (Gambar 2.7b) pada teknik modulasi FSK dan GFSK.



(a)



(b)

Gambar 2.7. (a) Perbandingan Modulasi FSK dan GFSK pada Domain Waktu, (b) Perbandingan Modulasi FSK dan GFSK pada Domain Frekuensi.

2.3. Teknik Komunikasi Data Digital

Pada dua perangkat yang terhubung oleh suatu media transmisi untuk melakukan pertukaran data, sebuah kerjasama yang baik sangat dibutuhkan. Terutama pada data yang ditransmisikan satu bit per satu satuan waktu melalui suatu medium. Pewaktuan (laju kecepatan, durasi, jarak) untuk tiap-tiap bit tersebut harus sama antara transmitter dan receiver. Dua teknik yang umum digunakan untuk mengendalikan pewaktuan ini yaitu asinkronisasi dan sinkronisasi.

2.3.1. Transmisi Asinkron dan Sinkron

Pada sistem transmisi asinkron, data ditransmisikan per karakter dalam satu satuan waktu, dengan panjang karakter bervariasi antara lima hingga delapan bit. Pewaktuan atau sinkronisasi harus dijaga untuk tiap karakter; *receiver*

memiliki kesempatan untuk melakukan sinkronisasi ulang pada tiap awal pentransmisian karakter yang baru.

Jenis kesalahan yang umum terjadi pada sistem transmisi asinkron adalah *framing error*. Istilah *frame* pada sistem transmisi asinkron mengacu pada 5 hingga 8 bit karakter yang dikirimkan, beserta bit *start* dan *stop*. *Framing error* terjadi ketika adanya kesalahan penerimaan bit terakhir yang dikirimkan, dan perhitungan bit yang dilakukan melewati deretan bit yang dikirimkan. *Framing error* juga dapat terjadi karena adanya noise yang menyebabkan munculnya bit start palsu pada saat keadaan transmisi sedang *idle*.

Pada sistem transmisi sinkron, suatu blok bit ditransmisikan dengan aliran bit yang tetap tanpa ada penambahan *start* atau *stop code*. Panjang blok dapat terdiri dari bit-bit dalam jumlah yang cukup banyak. Untuk menghindari adanya *drift* pewaktuan antara *transmitter* dan *receiver*, frekuensi *clock* masing-masing harus tersinkronisasi. Hal ini dapat dilakukan dengan menambah satu jalur *clock* terpisah diantara *transmitter* dan *receiver*. Metode ini dapat bekerja dengan baik untuk sistem komunikasi pada jarak dekat. Namun untuk jarak yang lebih jauh, frekuensi *clock* akan mengalami pelemahan yang sama seperti pada sinyal data, dan dapat menyebabkan terjadinya kesalahan pada pewaktuan. Alternatif lain adalah dengan menambahkan informasi *clock* pada sinyal data.

Dibutuhkan level sinkronisasi yang berbeda pada sistem transmisi sinkron agar *receiver* dapat menentukan awal dan akhir dari suatu blok data. Tiap blok data akan diawali dengan bit *preamble* dan umumnya diakhiri dengan bit *postamble*, serta ada penambahan bit control yang digunakan pada prosedur *data link control*.

2.3.2. Konfigurasi Jalur Transmisi

Dalam sistem komunikasi, secara umum dikenal 3 jenis konfigurasi jalur transmisi yaitu:

1. *Simplex*

   Pada arah komunikasi *simplex*, sistem berkomunikasi pada satu arah dengan keberadaan pemancara (pengirim) dan penerima adalah tetap. Contoh aplikasi

sistem komunikasi *simplex* yaitu pada remote TV, informasi radio, dan lain lain.

2. *Half-Duplex*

   Pada sistem komunikasi *half-duplex*, sistem dapat berkomunikasi dengan dua arah, namun dilakukan secara bergantian sehingga pada satu saat hanya ada satu perangkat yang berfungsi sebagai pemancar (pengirim) atau penerima saja. Contoh aplikasi pada sistem komunikasi *half-duplex* yaitu Radio Walky Talky, Radio CB, dan lain-lain.

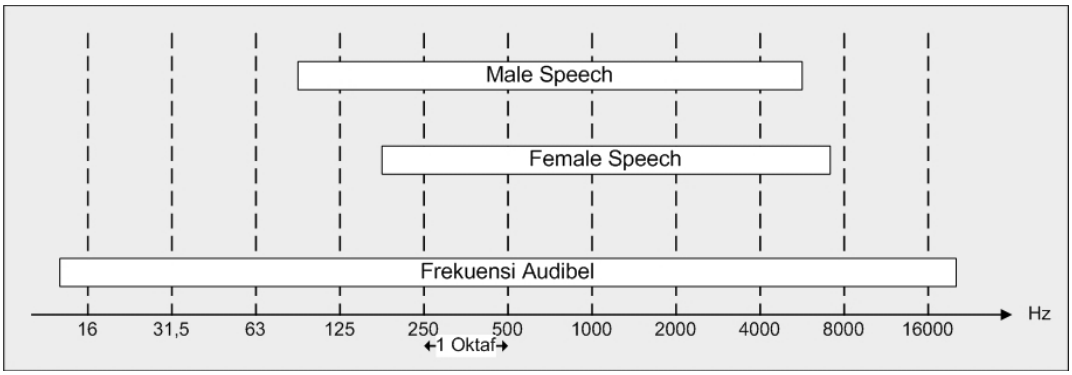
3. *Full-Duplex*

   Pada sistem komunikasi *full-duplex*, sistem dapat berkomunikasi dengan dua arah sekaligus secara langsung atau bersamaan. Contoh aplikai pada sistem komunikasi *full-duplex* yaitu telepon, handphone, dan lain-lain.

2.4. Spektrum Frekuensi Suara Manusia

Suara merupakan sebuah fenomena fisik yang menstimulasi indera pendengaran. Pada manusia, suara dapat didengar ketika frekuensi getaran berada diantara 15 Hz hingga sekitar 20 kHz. Hertz (HZ) merupakan unit frekuensi yang berarti banyaknya getaran atau *cycle* tiap detik. Tingkat sensitifitas sistem pendengaran manusia berada pada rentang frekuensi 1000 Hz hingga 4000 Hz.

Frekuensi audibel pada Gambar 2.8 dibagi menjadi beberapa tingkat rentang frekuensi, yaitu dalam satuan oktaf yang berarti interval antara dua frekuensi yang memiliki rasio sebesar 2:1. Sebagai referensi, besar frekuensi untuk nada middle C adalah 250 Hz.



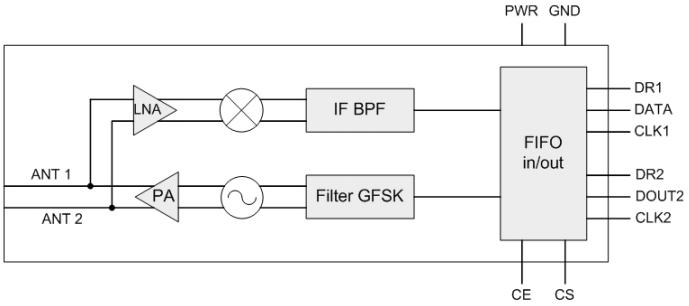
Gambar 2.8. Ilustrasi Suara dalam Domain Frekuensi

2.5. Wenshing TRW-2.4G

Wenshing TRW-2.4G merupakan sebuah modul *transceiver* berfrekuensi tinggi yang berbasis pada *single-chip* 2,4 GHz *transceiver*, nRF2401 dari NORDIC Semiconductor. Modul ini bekerja pada pita frekuensi 2,4 GHz dengan rentang antara 2,4 ~ 2,527 GHz dan menggunakan modulasi GFSK, serta bekerja dengan data rate sebesar 250 kbps dan 1 Mbps.

nRF2401 merupakan sebuah *chip* tunggal radio *tranceiver* yang terdiri dari sebuah *frequency synthesizer* terintegrasi, *power amplifier*, *crystal oscillator* dan sebuah modulator. Output kanal frekuensi dan daya dapat diprogram melalui *3-wire interface*. Konsumsi arus yang dibutuhkan juga cukup rendah, yaitu 10,5 mA untuk *power output* sebesar -5 dBm dan 18 mA pada mode penerima. *Chip* ini bekerja pada rentang level tegangan sebesar 1,9 hingga 3,6 V.

Gambar 2.9 mengilustrasikan blok diagram dari Wenshing TRW-2.4G beserta fungsi masing-masing pinnya.



Gambar 2.9. Diagram Blok dan Konfigurasi Pin Wenshing TRW-2.4G

Fungsi dari masing-masing pin TRW-2.4G pada Gambar 2.9 dideskripsikan pada Tabel 2.1 berikut:

Tabel 2.1. Deskripsi Fungsi Pin Wenshing TRW-2.4G

Pin	Nama	Fungsi	Deskripsi
1	CLK1	Digital I/O	*Clock* input (TX) & Output/Input untuk Data *Channel* 1 *3-wire interface*
2	CS	Digital Input	*Chip Select* untuk mengaktifasi *Configuration Mode*
3	CLK2	Digital I/O	*Clock* Output/Input untuk RX Data *Channel* 2
4	CE	Digital Input	*Chip Enable* untuk mengaktifkan mode pengirim (TX) atau penerima (RX)
5	GND	*Power*	*Ground* (0 V)
6	VCC/PWR	*Power*	*Power Supply* (+3 V DC)
7	DR2	Digital Output	RX *Data Ready* pada Data *Channel* 2 (Shockburst™)
8	DOUT2	Digital Output	RX Data *Channel* 2
9	DR1	Digital Output	RX *Data Ready* pada Data *Channel* 1 (Shockburst™)
10	DATA	Digital I/O	RX Data *Channel* 1/TX Data Input *3-wire interface*

2.5.1. Mode Operasi Wenshing TRW-2.4G

Mode operasi secara umum pada Wenshing TRW-2.4G dapat diatur dalam kondisi mode utama berikut berdasarkan konfigurasi tiga pin kontrol:

Tabel 2.2. Deskripsi Fungsi Pin Wenshing TRW-2.4G

Mode	PWR_UP	CE	CS
Active (RX/TX)	1	1	0
Configuration	1	0	1
Stand By	1	0	0
Power Down	0	X	X

2.5.1.1. Mode Active (RX/TX)

Wenshing TRW-2.4G memiliki dua mode aktif (RX/TX) yang ditentukan berdasarkan pada informasi dari *configuration word* yang telah diatur pada mode konfigurasi (*configuration mode*). Dua mode aktif tersebut antara lain Shockburst™, *Direct Mode*, dan DuoCeiver™ *Simultaneous Two Channel Receive Mode*.

1. Shockburst™

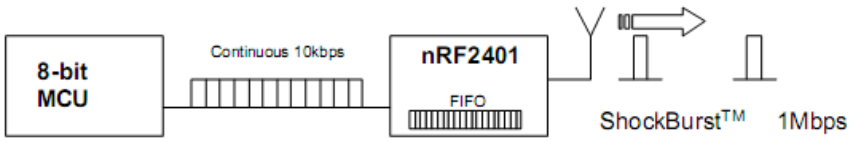
Teknologi Shockburst™ menggunakan *on-chip* FIFO untuk melakukan *clock* data masuk pada kecepatan rendah dan melakukan pengiriman pada kecepatan sangat tinggi serta mengaktifkan pengurangan daya secara ekstrim. Ketika mengaktifkan mode operasi Shockburst™ pada TRW-2.4G, kecepatan pengiriman data yang dapat dicapai adalah hingga 1 Mbps pada pita frekuensi 2,4 GHz tanpa harus menggunakan mikrokontroler berkecepatan tinggi untuk melakukan pemrosesan data.

Dengan mengaktifkan keseluruhan pemrosesan sinyal berkecepatan tinggi yang berhubungan dengan protokol RF on-chip, TRW-2.4G memiliki beberapa kelebihan yaitu:

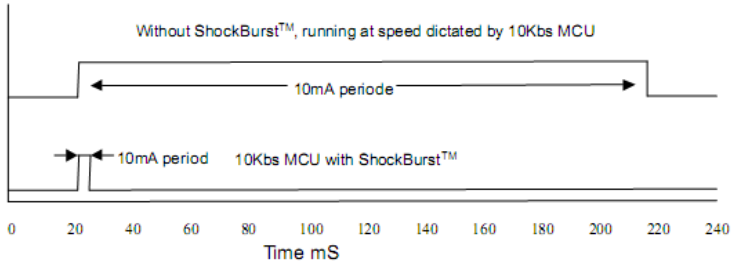
1. Mengurangi konsumsi daya.
2. Sistem yang murah untuk penggunaan mikrokontroler yang lebih murah.
3. Mengurangi resiko tabrakan "*on-air*" karena waktu transmisi yang singkat.

Wenshing TRW-2.4G dapat diprogram dengan menggunakan *3-wire interface* sederhana yang dikonfigurasikan melalui mikrokontroler. Dengan mengatur bagian lain dari aplikasi untuk bekerja dengan kecepatan rendah pada saat data

rate RF *link* bekerja pada kecepatan tinggi, mode Shockburst™ dapat mengurangi konsumsi daya rata-rata pada aplikasi. Ketika TRW-2.4G bekerja pada mode Shockburst™, operasi pengiriman (TX) atau penerimaan (RX) data akan bekerja seperti pada Gambar 2.10.



Gambar 2.10. *Clocking-in* Data dengan Mikrokontroler dan Pengiriman Data dengan Teknologi Shockburst™



Gambar 2.11. Konsumsi Arus dengan dan tanpa Shockburst™

2. Direct Mode

Pada mode Direct Mode Wenshing TRW-2.4G bekerja seperti perangkat RF biasa. Data harus berada pada 1 Mbps±200ppm atau 250 kbps±200ppm pada konfigurasi *data rate* berkecepatan rendah, agar dapat dideteksi oleh penerima.

3. DuoCeiver™ *Simultaneous Two Channel Receive Mode*

Pada mode Shockburst™ dan Direct Mode di kedua perangkat, Wenshing TRW-2.4G dapat memfasilitasi penerimaan data melalui dua *channel* frekuensi yang independen secara paralel pada *data rate* maksimum. TRW-2.4G dapat menerima data dari dua *transmitter* di kecepatan 1 Mbps dengan beda frekuensi masing-masing sebesar 8 MHz melalui sebuah antarmuka antena. Teknologi DuaCeiver™ pada TRW-2.4G menyediakan dua kanal data yang terpisah dan *dedicated* yang berfungsi sebagai penerimaan (RX) dan menggantikan kebutuhan dua buah sistem *stand alone* untuk penerimaan data.

2.5.1.2. Mode Konfigurasi

Pada mode konfigurasi, 15-byte *configuration word* di *download* ke TRW-2.4G. Hal ini dapat dilakukan dengan menggunakan *3-wire interface* yang sederhana. Tinjauan secara umum mengenai *configuration word* yang dapat diaplikasikan pada TRW-2.4G dapat dilihat pada lampiran.

2.5.1.3. Mode *Stand By*

Mode *stand by* digunakan untuk meminimalkan konsumsi arus rata-rata setelah waktu *start up* yang singkat. Pada mode ini, bagian kristal osilator akan aktif. Konsumsi arus bergantung pada frekuensi kristal (contoh: 12 µA @ 4 MHz dan 32 µA @ 16 MHz). *Configuration word* tetap terjaga pada saat *stand by*.

2.5.1.4. Mode *Power Down*

Pada mode *power down* TRW-2.4G dinon-aktifkan dengan konsumsi arus paling minimal (kurang dari 1 µA). *Configuration word* tetap terjaga pada saat *power down*.

2.5.2. Data Pewaktuan pada Wenshing TRW-2.4G

Berikut adalah beberapa data pewaktuan untuk diterapkan pada Wenshing TRW-2.4G.

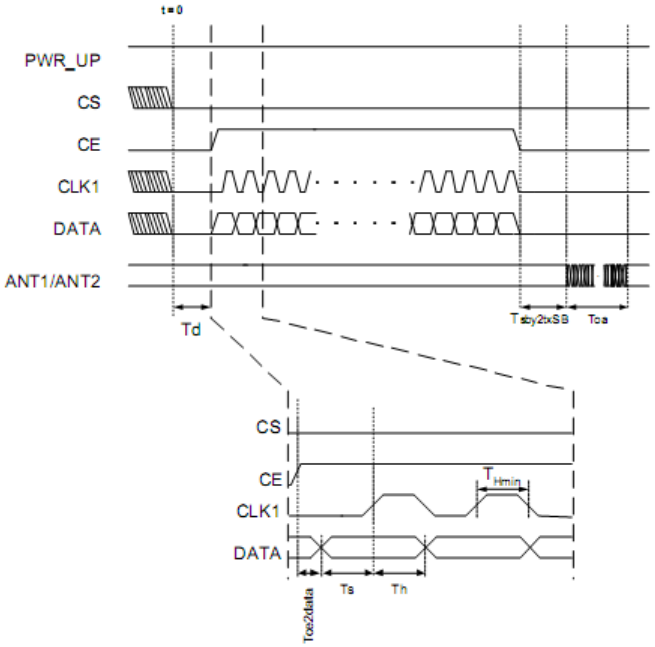
Tabel 2.3. Pewaktuan pada Operasional TRW-2.4G

No	Pewaktuan TRW-2.4G	Maks.	Min	Nama
1	PWR_DWN→Configuration Mode	3 ms		Tpd2cfgm
2	PWR_DWN→Active Mode (TX/RX)	3 ms		Tpd2a
3	ST_BY→TX Shockburst™	195 µs		Tsby2txSB
4	ST_BY→TX Direct Mode	202 µs		Tsby2txDM
5	ST_BY→RX mode	202 µs		Tsby2rx
6	Minimum delay dari CS ke data		5 µs	Tcs2data
7	Minimum delay dari CE ke data		5 µs	Tce2data
8	Minimum delay dari DR1/DR2 ke CLK		50 ns	Tdr2clk
9	Maksimum delay dari clk ke data	50 ns		Tclk2data
10	Delay untuk menyelesaikan GFSK internal data		1/data rate	Tfd
11	Minimum input clock tinggi		500 ns	Thmin

Ketika TRW-2.4G dalam keadaan *power down*, akan membutuhkan waktu untuk memasuki mode *stand by* selama 3 ms sebelum dapat memasuki mode konfigurasi atau salah satu mode aktif (TX/RX).

Configuration word akan hilang ketika VDD dalam keadaan mati (*off*) dan harus dikonfigurasi kembali sebelum memasuki salah satu mode aktif. Jika TRW-2.4G telah dikonfigurasi, maka TRW-2.4G dapat langsung memasuki salah satu mode aktif yang diinginkan dari kondisi *power down*. Konfigurasi pin CE dan CS tidak berada dalam keadaan HIGH pada waktu yang bersamaan. Pengaturannya dilakukan setelah mode konfigurasi atau mode aktif telah diaktifkan.

Pada mode Shockburst™, pewaktuan saat melakukan pengiriman data (TX) ditentukan oleh diagram pewaktuan seperti pada Gambar 2.12 berikut:

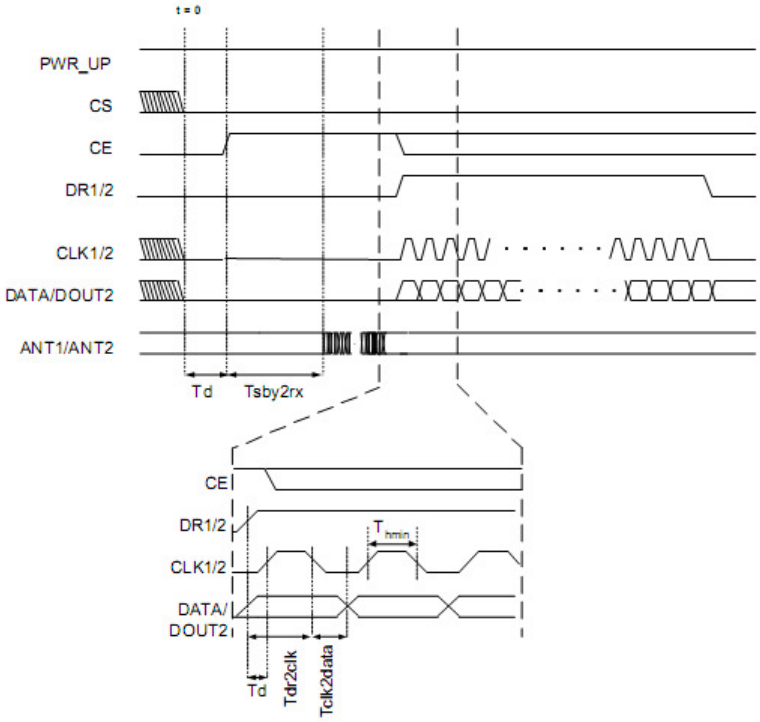


Gambar 2.12. Diagram Pewaktuan Mode Shockburst™ saat Melakukan Pengiriman Data

Panjang paket yang ditransmisikan dan *data rate* akan memberikan *delay* T_{OA} (*Time on air*) berdasarkan persamaan:

$$T_{OA} = \frac{1}{data\ rate} \cdot (\#databits + 1) \tag{2.7}$$

Penerimaan data (RX) pada mode Shockburst™ ditentukan oleh diagram pewaktuan pada Gambar 2.13.



Gambar 2.13. Diagram Pewaktuan Mode Shockburst™ saat Melakukan Penerimaan Data

Konfigurasi pin CE dapat tetap dalam kondisi HIGH ketika melakukan penerimaan data, namun akan membutuhkan konsumsi arus yang besar (18 mA). Keuntungannya, akan dibutuhkan waktu yang lebih singkat (200 µs) saat DR1 LOW.

2.6. Mikrokontroler AVR ATmega8535

Mikrokontroler merupakan sebuah sistem komputer yang dioptimasikan pada pengontrolan perangkat keras yang terintegrasi dengan sebuah prosesor, memori, dan perangkat I/O pada satu chip silikon tunggal. Atmel AVR merupakan salah satu jenis mikrokontroler pada keluarga mikrokontroler RISC 8-bit yang diproduksi oleh Atmel. RISC (*Reduced Instruction Set Computing*) memungkinkan sebuah perangkat mikrokontroler untuk beroperasi dengan cepat dengan mengurangi beberapa jenis instruksi mesin (*machine-level instruction*).

2.6.1. Memori

Atmel AVR menggunakan memori internal untuk penyimpanan data dan program. Terdapat empat tipe memori pada Atmel AVR antara lain: p*rogram flash memory*, memori data (register, I/O register, dan SRAM), EEPROM, dan fuse bit. Keempat memori tersebut terdapat dalam satu chip tunggal dan pada lokasi yang terpisah. Alamat ke-0 pada memori data berbeda dengan alamat ke 0 pada program flash memory dan alamat ke-0 pada EEPROM.

2.6.1.1 *Program Flash Memory*

ATmega8535 memiliki 8 KB o*n-chip* i*n-system reprogramable flash memory non-volatile* untuk penyimpanan program. Memori program menyimpan opcode yang dapat dieksekusi dan data statis. AVR tidak dapat menggunakan memori program eksternal; memori flash o*n-chip* merupakan satu-satunya memori program yang dapat diakses oleh *core* AVR. Karena keseluruhan instruksi AVR memiliki lebar data sebesar 16-bit, maka p*rogram flash memory* terorganisasi atas 4 KB x 16.

2.6.1.2. *Data Memory*

Data memory pada ATmega8535 terdiri dari 32 *general purpose registers* 8-bit (0x0000 - 0x001F), 64 I/O register (0x0020 - 0x005F), dan SRAM internal sebesar 512 Byte (0x0060 - 0x025F). *Data memory* tersebut dapat diakses dengan lima mode pengalamatan, yaitu: *direct addressing*, *indirect addressing with displacement*, *indirect addressing*, *indirect addressing with pre-decrement*, dan *indirect addressing with post-increment.*

2.6.2. Reset dan *Interrupt*

Interrupt akan melakukan lompatan pada sebuah *interrupt service routine* (ISR) yang melakukan apa yang seharusnya dilakukan. *Interrupt* berguna pada situasi tertentu ketika prosesor harus segera merespon suatu *interrupt*. AVR mendukung beberapa jenis sumber *interrupt* internal dan eksternal. Sebuah *interrupt* dapat terjadi dari perangkat internal yang mencapai keadaan tertentu (seperti karakter yang diterima oleh UART), atau dari kejadian eksternal seperti

level tertentu pada sebuah pin. Setiap *interrupt* akan menyebabkan program melompat ke alamat memori tertentu yang memiliki intruksi RETI (*Return from Interrupt*) untuk mengabaikan *interrupt*, atau ke sebuah *interrupt handler*.

Pada prosesor AVR, semua *interrupt* memiliki prioritas yang sama. Tidak dizinkan sebuah *interrupt* untuk melakukan *interrupt* terhadap *interrupt* lain, sehingga, sebuah *interrupt* dikatakan tidak memiliki prioritas lebih tinggi dari *interrupt* lain. Namun, adalah memungkinkan terjadinya dua atau lebih *interrupt* secara simultan. Ketika terjadi dua jenis *interrupt* yang harus dieksekusi, maka *interrupt* dengan urutan vektor terendah yang akan dieksekusi terlebih dahulu. Urutan vektor *interrupt* pada ATmega8535 dapat dilihat pada Tabel 2.4 berikut.

Tabel 2.4. Urutan Vektor *Interrupt* pada ATmega8535

Alamat Awal	Urutan Vektor	Sumber	Deskripsi
0x0000	1	Reset	Reset eksternal, power up, watchdog timeout
0x0002	2	*Interrupt* Eksternal 0	Sinyal *hardware* pada pin INT0
0x0004	3	*Interrupt* Eksternal 1	Sinyal *hardware* pada pin INT1
0x0006	4	Timer 2 Compare	Timer 2 *compare match*
0x0008	5	Timer 2 Overflow	Timer 2 *overflow*

Interrupt harus diinisialisasikan terlebih dahulu sebelum dapat dieksekusi. Inisialisasi *Interrupt* merupakan proses dua langkah, yaitu melakukan *unmasking* (berarti memberikan sebuah nilai 1 pada register yang sesuai dengan *interrupt* yang akan diaktifkan) pada *interrupt* yang akan diaktifkan, dan mengaktifkan semua *interrupt* yang telah di-*unmask* secara global.

GICR (G*eneral Interrupt Control Register*) digunakan untuk mengaktifkan *interrupt* eksternal. Men-*set* bit INT0 akan mengaktifkan *interrupt* eksternal 0. Ketika perangkat memberikan sinyal pada INT0, sinyal tersebut akan di-AND-kan secara logika dengan bit INT0 pada GICR. Jika hasilnya bernilai 1, maka *interrupt* diijinkan untuk terjadi dan sebaliknya jika bit INT0 bernilai 0 atau hasil operasi AND selalu 0. Setelah itu, prosesor akan segera mengarahkan *interrupt* ke alamat yang sesuai dengan tabel vektor *interrupt*.

ATmega8535 memiliki dua pin eksternal *interrupt*, yaitu INT0 dan INT1. AVR dapat melakukan *triggering* pada level *high* atau level *low*, pada *falling edge*, atau pada *rising edge*. Pada beberapa seri AVR terbaru mengimplementasikan *pin-change interrupt* untuk semua pin pada delapan pin dalam satu grup, menggantikan kebutuhan untuk melakukan *polling* terhadap pin. *Pin-change interrupt handler* akan memeriksa keadaan pin yang berasosiasi dengan vektor *interrupt*, dan menentukan tindakan yang akan lakukan.

RESET merupakan *interrupt* dengan urutan vektor paling rendah, sekaligus merupakan *interrupt* yang khusus yang selalu diutamakan diatas semua *interrupt* dan kode program yang sedang dieksekusi. Pin RESET pada AVR merupakan sebuah pin input *active-low* yang akan memaksa untuk me-reset prosesor dan perangkat lain yang teintegrasi. Reset dapat dikendalikan oleh sebuah generator *power-on* reset eksternal, sebuah supervisor tegangan yang akan memberikan sinyal reset ketika sumber tegangan berada dibawah batas tertentu, atau komponen lain pada sistem yang lebih besar.

2.6.3. General Purpose I/O Port

Pin *General Purpose* I/O (GPIO) merupakan I/O digital pada mikroprosesor AVR. AVR dapat mengendalikan sebuah level *high* atau *low*, atau mengkonfigurasi pin sebagai sebuah input dengan atau tanpa *pull-up*. GPIO terdiri dari grup port yang memiliki setidaknya 8 buah pin. Terdapat register control yang digunakan untuk mengatur arah data (input atau output), nilai output (atau *pull-up* aktif), dan untuk membaca nilai dari pin itu sendiri. Setiap pin dapat diakses dengan menggunakan intruksi manipulasi *bitwise*.

Setiap port pada AVR memiliki 3 register kontrol yaitu DDRx, PORTx, dan PINx. Setiap bit pada register tersebut mengendalikan tiap pin GPIO. Register DDRx mengendalikan apakah sebuah pin digunakan sebagai input atau output. Jika nilai logika register DDRx bernilai satu, maka pin Pxn akan berfungsi sebagai output. Dan jika bernilai logika nol, maka Pxn akan berfungsi sebagai input. Jika PORTxn bernilai logika satu ketika pin yang bersesuaian dikonfigurasi sebagai pin input, maka *pull-up* resistor akan diaktifkan. Untuk menonaktifkan

pull-up resistor, maka PORTxn harus diberi nilai logika nol atau pin dikonfigurasikan sebagai pin output.

2.6.4. Timer atau Counter

Timer atau counter merupakan sebuah fitur yang paling umum dan kompleks dalam sebuah mikrokontroler. Timer atau counter sangat serbaguna, beberapa fungsinya antara lain untuk mengukur periode waktu, menentukan lebar pulsa, mengukur kecepatan, mengukur frekuensi, atau menghasilkan suatu sinyal output. Timer atau counter memiliki dua mode yang berbeda, yaitu sebagai timer dan sebagai counter. Sebagai sebuah timer, pencacah biner akan menghitung periode waktu dari suatu nilai input yang telah diberikan. Dan sebagai sebuah counter, pencacah biner akan menghitung banyaknya kejadian atau pulsa.

ATmega8535 memiliki tiga buah timer atau counter, yaitu TIMER atau COUNTER 0 (8-bit), TIMER atau COUNTER 1 (16-bit), dan TIMER atau COUNTER 2 (8-bit). Timer 8-bit akan mencacah hingga nilai maksimum 0xFF (255), dan timer 16-bit akan mencacah hingga nilai maksimum 0xFFFF (65.535). Timer atau counter akan melakukan *roll over* ke nilai 0 ketika telah mencapai nilai maksimum. Kejadian *roll over* ini sangat berguna bagi program untuk dapat melakukan pembacaan dari sebuah timer atau counter secara tepat dan melakukan proses tertentu.

2.6.4.1. Prescaler dan Selektor Input Timer atau Counter

Unit timer atau counter dapat menggunakan beberapa frekuensi internal dengan *clock* sistem sebagai input, atau mendapatkan input dari pin eksternal. *Timer/Counter Control Register* (TCCRx) bersesuaian dengan timer yang memiliki *counter select bit* (CSx2, CSx1, dan CSx0) untuk mengontrol pin yang akan digunakan dengan *counter* tertentu.

2.6.4.2. Timer 0

Timer atau counter 0 merupakan timer 8-bit yang dapat berfungsi sebagai fungsi timer atau counter biasa tetapi lebih sering digunakan untuk menciptakan suatu detak program. TCCR0 mengontrol fungsi Timer 0 dengan memilih sumber

clock yang akan digunakan pada Timer 0. Gambar 2.14 menunjukkan skema bit pada register TCCR0. Penjelasan mengenai konfigurasi yang dapat dilakukan pada bit-bit tersebut dapat dilihat pada Tabel 2.5.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					CS02	CS01	CS00

Gambar 2.14. Register TCCR0

Tabel 2.5. Definisi *Prescaler Bit* pada Register TCCR0

CS02	CS01	CS00	Fungsi Interrupt
0	0	0	Stop, Timer 0 stop
0	0	1	System clock, CK
0	1	0	System clock/8, CK/8
0	1	1	System clock/64, CK/64
1	0	0	System clock/256, CK/256
1	0	1	System clock/1024, CK/1024
1	1	0	Pin eksternal T0, menghitung *falling edge*
1	1	1	Pin eksternal T0, menghitung *rising edge*

Detak program akan memberikan timing dengan tingkat akurasi tinggi. Keseluruhan pola akan terbentuk ketika sebuah nilai di-*load* ke dalam timer. Timer akan menghitung nilai tersebut hingga mencapai 255 (0xFF) dan mengulang kembali. Pengulangan kembali dapat menyebabkan terjadinya *interrupt*. *Interrupt service routine* akan me-*reload* nilai yang sama ke ke dalam timer, mengeksekusi rutin tertentu, dan kembali ke program. *Interrupt* akan terjadi secara berkala pada setiap periode waktu tertentu. Besarnya nilai yang di-*load* ke dalam timer menentukan panjang periode waktu. Semakin kecil, maka akan semakin lama counter mencapai nilai maksimum, dan semakin lama detak program yang terjadi.

2.6.4.3. Timer 1

Timer atau counter 1 adalah timer 16-bit memungkinkan eksekusi timing program yang lebih akurat dan lebih kompleks dibandingkan dengan timer 8-bit. Timer 1 memiliki satu *input capture register* 16-bit dan dua *output compare register* 16-bit. *Input capture register* digunakan untuk mengukur lebar pulsa atau

meng-*capture* waktu. *Output compare register* digunakan untuk menghasilkan frekuensi atau pulsa dari timer/conter ke sebuah pin output pada mikrokontroler.

Timer/Counter Compare Register 1 (TCCR1) memiliki dua register, TCCR1A dan TCCR1B. TCCR1A digunakan untuk mengontrol *compare mode* dan mode PWM pada Timer 1. TCCR1B digunakan untuk mengontrol *prescaler* dan *multiplexer* input untuk Timer 1, seperti pada *input capture mode*. TCCR1B *counter select bit* mengontrol input pada Timer 1 sama seperti pada counter select bit untuk Timer 0. Gambar 2.15 menunjukkan skema bit pada register TCCR1B. Penjelasan mengenai konfigurasi yang dapat dilakukan pada bit-bit tersebut dapat dilihat pada Tabel 2.6.

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
ICNC1	ICES1		WGM13	WGM12	CS12	CS11	CS10

Gambar 2.15. Register TCCR1B

Tabel 2.6. Definisi Bit pada Register TCCR1B

Bit	**Fungsi**
ICNC1	*Input Capture Noise Canceller* (1 = aktif)
ICES1	*Input Capture Edge Select* (1 = aktif)
WGM1x	*Output waveform control*
CS12	*Counter Input Select Bit* Memiliki definisi yang sama dengan Timer 0
CS11	
CS10	

Untuk mengukur suatu kejadian, waktu pada Timer 1 akan melakukan *capture* pada awal waktu kejadian dan pada akhir waktu kejadian, serta melakukan operasi pengurangan diantara kedua waktu tersebut untuk menentukan besar waktu yang dibutuhkan oleh suatu kejadian. Pada Timer 1, fungsi ini dikendalikan oleh *input capture register* (ICR1). ICR1 merupakan register 16-bit, terdiri dari ICR1H dan ICR1L, yang akan melakukan *capture* terhadap pembacaan aktual Timer 1 ketika mikrokontroler menerima sinyal tertentu. Sinyal yang menyebabkan proses *capturing* dapat terjadi pada saat *rising edge* atau *falling edge*, sesuai dengan pengaturan yang telah ditentukan pada *input capture pin*, ICP, pada mikrokontroler. Pemilihan *capture* pada saat *rising edge* atau *falling edge* dikontrol oleh *input capture edge select bit*, ICES1. Nilai logika 1 akan mengijinkan ICR1 untuk melakukan *capture* terhadap waktu Timer 1 pada saat

rising edge, dan nilai logika 0 akan mengijinkan ICR1 untuk melakukan *capture* waktu pada saat *falling edge*.

Karena hanya tersedia satu register *capture* untuk Timer 1, maka proses *capture* harus segera dibaca sesegera mungkin untuk menghindari terjadinya *overwriting* data dari pembacaan sebelumnya. Terdapat sebuah *interrupt* untuk menandakan bahwa proses *capture* telah selesai dilakukan pada ICR1. Setiap proses *capture* telah selesai dilakukan, program harus menentukan apakah sinyal *interrupt* pada awal atau akhir even yang telah ditentukan sehingga dapat memproses data di dalam ICR1 secara tepat.

Timer 1 juga menyediakan fitur *input noise canceller* untuk mencegah adanya sinyal yang tidak diinginkan pada ICP yang disebabkan terjadinya proses *capture* pada waktu yang tidak ditentukan. Ketika fitur *noise canceller* diaktifkan, maka ICP harus aktif (berada pada level *high* untuk *rising edge*, atau *low* untuk *falling edge*) untuk empat sampel yang berurutan sebelum mikroprosesor akan memproses *trigger* untuk validasi dan *capture* data. Fitur ini dapat diaktifkan dengan men-*set* bit *input capture noise canceller*, ICNC1, pada register TCCR1B.

Output compare mode digunakan oleh mikrokontroler untuk menghasilkan sinyal output. Sinyal output dapat berupa sinyal kotak atau gelombang asimetris, dan dapat terdiri dari beberapa komponen frekuensi atau simetris. Pada *output compare mode*, program akan me-*load* nilai pada *output compare register*. Nilai pada *output compare register* akan dibandingkan dengan nilai pada *timer/counter register*, dan akan menyebabkan *interrupt* ketika dua nilai tersebut sesuai. *Interrupt* ini akan menjadi semacam alarm yang menyebabkan prosesor mengeksekusi fungsi yang berhubungan dengan sinyal yang dihasilkan tepat saat dibutuhkan.

Untuk membangkitkan sebuah *interrupt*, *output compare mode* dapat secara otomatis di-*set*, *clear*, atau dengan men-*toggle* pin tertentu pada port output. Pada Timer 1, *output compare mode* dikontrol oleh TCCR1A. *Bit compare mode control* akan menentukan tindakan yang akan dilakukan ketika ada kesesuaian antara *compare register* dan *timer register*. Pin output yang bersesuaian dapat dibuat tidak terpengaruh, *toggle*, *set*, atau *clear*. Kesesuaian nilai register juga dapat menyebabkan *interrupt*. Tujuan dari rutin *interrupt*

tersebut adalah untuk melakukan reset atau *reload compare register* untuk kesesuaian nilai register selanjutnya. Gambar 2.16 menunjukkan skema bit pada register TCCR1A. Penjelasan mengenai konfigurasi yang dapat dilakukan pada bit-bit tersebut dapat dilihat pada Tabel 2.7.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
COM1A 1	COM1A 0	COM1B 1	COM1B 0	FOC1 A	FOC1 B	WGM1 1	WGM1 0

Gambar 2.16. Register TCCR1A

Tabel 2.7. Definisi Bit pada Register TCCR1A

COM1x1	COM1x0	Fungsi
0	0	*No output*
0	1	*Compare match toggle the OC1x line*
1	0	*Compare match clear the OC1x line to 0*
1	1	*Compare match set the OC1x line to 1*

2.6.4.3. Timer 2

Timer 2 merupakan sebuah *timer/counter* 8-bit yang dilengkapi dengan fitur *output compare* dan PWM seperti Timer 1. Timer 2 dapat menggunakan kristal yang terpisah dari sistem clock sebagai sumber clock-nya. Pemilihan sumber clock eksternal pada Timer 2 dapat ditentukan dengan mengatur bit AS2 pada *asynchronous status register* (ASSR). Dengan menggunakan sumber clock terpisah dari sistem clock, Timer 2 bekerja secara asinkron terhadap sistem clock. Tiga bit lain pada ASSR digunakan untuk memastikan bahwa data tidak ditulis ke dalam register Timer 2 pada saat yang sama ketika perangkat keras melakukan update terhadap nilai register Timer 2. Register kontrol TCCR2 digunakan untuk mengendalikan operasi dari Timer 2.

2.6.5. *Analog to Digital Conversion* (ADC)

Perangkat *analog to digital converter* (ADC) digunakan untuk merepresentasikan proporsi sinyal analog yang telah disampel ke dalam bentuk bilangan digital diskrit. ADC memiliki rentang tegangan input yang dapat diterima, misalnya 0 V hingga 5 V, dan rentang output digital yang dihasilkan, misalnya 8-bit atau 10-bit.

Perangkat ADC pada mikrokontroler AVR memiliki resolusi bit maksimum sebesar 10-bit, dapat beroperasi hingga kecepatan 10 KSPS (*kilo-samples per second*), dan dapat melakukan pembacaan sinyal analog dari salah satu dari 8 pin input yang berbeda. Dua register yang mengontrol ADC yaitu ADC *control and status register* (ADCSRA) untuk mengontrol fungsi ADC, dan ADC *multiplexer select register* (ADMUX) untuk mengontrol delapan kemungkinan input yang berbeda. Gambar 2.17 menunjukkan skema bit pada register ADCSRA. Penjelasan mengenai konfigurasi yang dapat dilakukan pada bit-bit tersebut dapat dilihat pada Tabel 2.8.

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Gambar 2.17. Register ADCSRA

Tabel 2.8. Definisi Bit pada Register ADCSRA

Bit	**Fungsi**
ADEN	*ADC Enable bit. Set to enable the ADC.*
ADSC	*ADC Start Conversion bit. Set to start conversion.*
ADATE	*ADC Free Running Select bit. Set to enable free run mode*
ADIF	*ADC Interrupt Flag bit. Is set by hardware at the end of a conversion cycle*
ADIE	*ADC Interrupt Mask bit. Set to allow interrupt to occur at the end of a conversion*
ADPS2	*ADC Prescaler select bit*
ADPS1	
ADPS0	

ADC membutuhkan clock sebesar 50 kHz hingga 200 kHz untuk beroperasi pada resolusi maksimum. Semakin tinggi clock frekuensi ADC akan menurunkan besar resolusi bit. Clock pada ADC berasal dari sistem clock dengan menggunakan *prescaler* sama seperti pada *timer/counter*. 3 bit LSB pada register ADCSRA mengontrol besar rasio pembagi *prescaler* untuk memberikan clock ADC antara 50 kHz hingga 200 kHz. Salah satu metode langsung untuk menentukan besar faktor preselector yaitu dengan membagi clock sistem dengan 200 kHz dan memilih faktor pembagi terbesar dari nilai yang dihasilkan. Hal ini akan menjamin ADC untuk beroperasi pada clock tertingginya tetapi tetap dalam rentang dibawah 200 kHz.

ADC bekerja dengan kecepatan dibawah kecepatan prosesor, sehingga ADC umumnya digunakan pada mode *interrupt-driven*. Namun, ADC tetap

dimungkinkan untuk bekerja pada mode *free running* yang akan melakukan konversi secara kontinu secepat mungkin. Pembacaan hasil konversi ADC pada mode *free running* dilakukan dengan menonaktifkan *interrupt* atau memberhentikan proses konversi *free running*, melakukan pembacaan hasil, dan mengaktifkan kembali *interrupt* dan mode *free running*. Langkah ini dilakukan agar data hasil pembacaan ADC adalah akurat, program tidak akan melakukan pembacaan data selama prosesor belum melakukan *update* ke register hasil pembacaan ADC. Konfigurasi bit untuk menetukan rasio prescaler ADC dapat dilihat pada Tabel 2.9.

Tabel 2.9. Rasio *Preselector* ADC

ADPS2	ADPS1	ADPS0	Faktor Pembagi
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Proses inisialisasi ADC biasanya dilakukan untuk mengatur konfigurasi ADC, mengaktifkan, dan memulai proses konversi pertama sekaligus. Langkah-langkah proses inisialisasi ADC adalah sebagai berikut:

1. Mengatur faktor pembagi pada 3 bit LSB di register ADCSRA.
2. Mengatur ADIE pada level tinggi untuk mengaktifkan mode interrupt.
3. Mengatur ADEN pada level tinggi untuk mengaktifkan ADC.
4. Mengatur ADSC untuk segera melakukan konversi data.

2.6.6. *Pulse-width Modulation* (PWM)

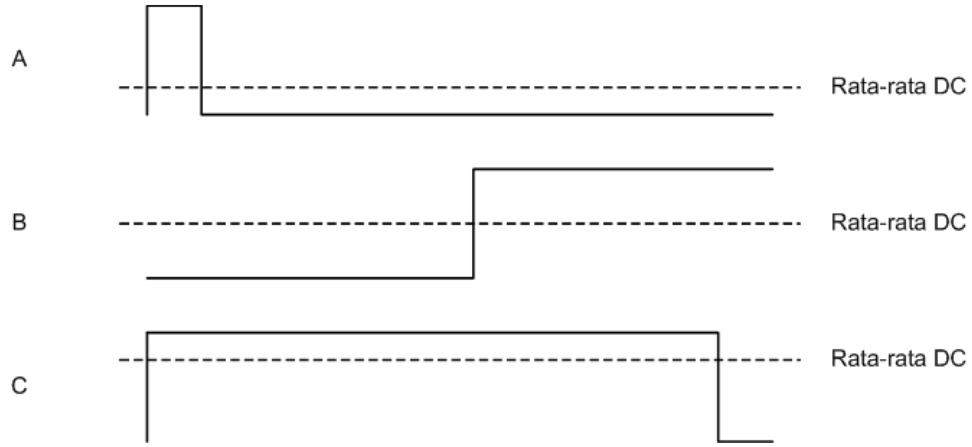
Pulse-width Modulation (PWM) merupakan metode yang digunakan untuk mengubah suatu bilangan digital diskrit ke dalam bentuk sinyal analog. PWM menggunakan kontrol digital untuk menciptakan sebuah sinyal kotak yang diaktifkan pada kondisi hidup (*on*) dan mati (*off*). Pola *on-off* ini mensimulasikan tegangan, misalnya 5 V pada kondisi *on* dan 0 V pada kondisi *off*, dengan mengubah-ubah durasi waktu pada kondisi *on* dan *off*. Lama durasi waktu (*duty*

cycle) pada kondisi *on* disebut sebagai *pulse-width* yang dapat digunakan untuk membangkitkan sebuah nilai analog tertentu. *Duty cycle* sinyal kotak output yang bervariasi akan menghasilkan sinyal DC yang bervariasi dan dengan memberikan filter gelombang output maka akan diperoleh gelombang rata-rata DC.

Pada mikrokontroler AVR, metode untuk menghasilkan sinyal PWM dilakukan dengan memanfaatkan fitur *pulse width modulation mode* pada Timer 1. Metode ini dilakukan dengan menggunakan Timer 1 yaitu dengan memanfaatkan *output compare register* dan pada tiap kesesuaian nilai register yang terjadi, memberikan variasi nilai untuk membentuk sinyal PWM. Timer 1 menyediakan suatu sistem *built-in* untuk menghasilkan sinyal PWM tanpa harus memprogram secara khusus fungsi yang melayani *compare register* untuk menciptakan suatu sinyal PWM.

Timer 1 akan mengubah mode operasinya untuk membangkitkan sinyal PWM. Ketika beroperasi pada PWM mode, Timer 1 akan menghitung secara menaik dan menurun, sehingga sulit digunakan bersamaan dengan mode lainnya. Saat melakukan operasi PWM, Timer 1 akan menghitung mulai dari nol hingga ke nilai maksimum dan kembali lagi ke nol. Nilai maksimum ditentukan oleh besar resolusi bit yang diinginkan. Tiga besar resolusi bit yang didukung yaitu 8-bit (255), 9-bit (511), dan 10-bit (1023). Besar resolusi bit ini ditentukan oleh *Waveform Generation Mode* (WGM) select bit pada TCCR1A (Gambar 2.16).

Duty cycle aktual sebagai output pada PWM mode bergantung pada nilai yang di-*load* pada *output compare register* sebagai *timer/counter*. Pada normal PWM mode, ketika *counter* menghitung mundur, maka bit output akan di-*set* tiap ada kesesuaian nilai dengan *output compare register*. Ketika *counter* menghitung maju, maka bit output akan di-*clear* tiap ada kesesuaian nilai dengan *output compare register*. *Load* nilai ke *output compare register* sebesar 20% dari nilai maksimum akan menghasilkan gelombang dengan *duty cycle* 20%. Contoh bentuk gelombang PWM terhadap tegangan rata-rata DC dapat dilihat pada Gambar 2.18 berikut.



Gambar 2.18. Bentuk Gelombang PWM

2.7. *Headset*

Istilah *headset* digunakan untuk menyebutkan perangkat *headphone* yang dikombinasikan dengan *microphone* dan digunakan untuk melakukan komunikasi dua arah. *Headset* memiliki fungsi yang sama dengan sebuah *handset* telepon yang dapat dioperasikan secara *hands-free*, biasanya digunakan disebuah sentral telepon atau kegiatan yang berhubungan dengan penggunaan telepon secara intensif.

2.7.1. *Headphone*

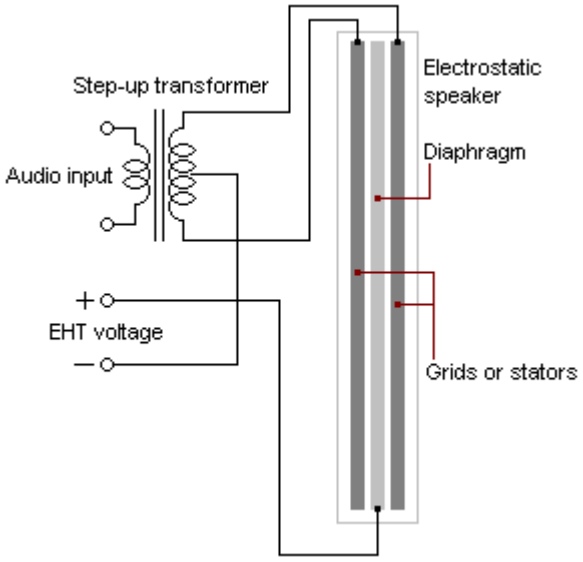
Headphone merupakan transducer *electro-to-acoustic* yang mengubah sinyal elektrik menjadi sinyal suara. *Headphone* umumnya terdiri dari sepasang *loudspeaker* berukuran kecil, atau sebuah *speaker*, yang dipasangkan pada telinga pengguna dan dihubungkan dengan sumber sinyal audio seperti *amplifier*, radio, atau CD *player*.

Transducer *headphone* menggunakan beberapa metode untuk mereproduksi suara, antara lain dengan menggunakan *moving-coil* atau *dynamic driver*, dan *driver* elektrostatis.

Moving-coil driver, atau lebih sering disebut "*dynamic*" *driver* merupakan jenis transducer yang paling banyak digunakan pada *headphone*. Prinsip kerjanya berada pada elemen magnetis stasioner yang dilekatkan pada rangka sebuah *headphone* untuk menciptakan medan magnet statis. Elemen magnetis pada *headphone* biasanya berupa ferrite atau neodymium. Diafragma, yang terpasang

pada sebuah kumparan (*voice coil*) dibenamkan pada medan magnet statis dari magnet stasioner. Diafragma akan dipengaruhi oleh sinyal audio yang melewati kumparan. Medan magnet yang ditimbulkan oleh arus yang melewati kumparan akan bereaksi melawan perubahan medan magnet statis, menyebabkan kumparan dan diafragma yang terpasang menggerakkan udara, dan memproduksi suara.

Driver elektrostatis terdiri dari sebuah diafragma beraliran listrik yang tipis, dan berada diantara dua plat metal (elektroda) berlubang. Sinyal elektrik audio yang melalui elektroda akan menciptakan medan listrik, sehingga diafragma akan mendekati salah satu elektroda, tergantung pada polaritasnya. Udara akan dipaksa melalui lubang; dengan perubahan sinyal elektrik kontinu yang mengendalikan membran, akan menghasilkan sebuah gelombang suara.



Gambar 2.19. *Driver* Elektrostatis Headphone

2.7.2. *Microphone*

Microphone merupakan sebuah transducer *acoustic-to-electric*, atau sensor yang mengubah sinyal suara menjadi sinyal elektrik. *Microphone* digunakan digunakan pada banyak aplikasi seperti telepon, *tape recorder*, *audio engineering*, *broadcasting*, dan lain-lain.

Elemen transducer sensitif pada *microphone* disebut sebagai elemen atau kapsul. Jenis-jenis *microphone* mengacu pada transducer yang digunakan dan karakteristik arahnya. Beberapa jenis elemen transducer pada *microphone* antara lain elektrostatis dan *dynamic* atau *moving-coil*.

Pada *microphone* elektrostatis, atau disebut juga *condencer* atau kapasitor, diafragma berlaku seperti salah satu plat dari sebuah kapasitor, dan getaran pada diafragma akan mengubah jarak antar plat. Salah satu metode pengubahan sinyal suara menjadi sinyal elektrik pada transducer yaitu *radio frequency* (RF) atau *high frequency* (HF) *condencer*. RF *condencer microphone* menggunakan perbandingan RF tegangan rendah, yang dihasilkan oleh sebuah osilator *low-noise*. Osilator dapat termodulasi frekuensi oleh perubahan kapasitansi yang dihasilkan oleh gelombang suara yang menggerakkan kapsul diafragma, atau kapsul menjadi bagian dari rangkaian resonansi yang akan memodulasi amplitudo sinyal osilator pada frekuensi tetap. Demodulasi akan menghasilkan sebuah sinyal frekuensi audio *low-noise* dengan impedansi sumber yang sangat rendah. Teknik ini memungkinkan penggunaan diafragma dengan ketegangan bebas, yang dapat digunakan untuk mencapai respon frekuensi yang lebih luas. Proses RF biasing akan menghasilkan kapsul dengan impedansi elektrik yang lebih rendah, sehingga memungkinkan *microphone* RF *condencer* dioperasikan pada keadaan lembab.

Microphone dynamic atau *moving-coil microphone*, bekerja berdasarkan induksi elektromagnetik. Prinsip kerja pada *microphone dynamic* merupakan kebalikan dari *dynamic driver* pada headphone. Sebuah kumparan konduktor kecil yang dapat bergerak, diletakkan pada bidang medan magnet dari sebuah magnet permanen, yang dilekatkan pada sebuah diafragma. Gelombang suara akan menggerakkan diafragma yang akan mengakibatkan kumparan bergerak dalam medan magnet, menghasilkan perubahan arus melalui proses induksi elektromagnetik.

2.7.3. Jenis-jenis *Headset*

Seiring dengan perkembangan teknologi komunikasi, saat ini *headset* tidak hanya digunakan pada aplikasi militer. Berdasarkan penggunaannya, jenis-jenis *headset* kategorikan menjadi *telephone headset*, *computer headset*, dan *mobile phone headset*.

Telephone headset biasanya digunakan pada sistem telepon *fixed-line*, PSTN atau PABX. *Telephone headset* berfungsi untuk menggantikan *handset* dari sebuah telepon.

Computer headset biasanya memiliki dua jenis koneksi yaitu dengan menggunakan 3,5 mm *jack* standar dan USB. Headset dengan *jack* standar 3,5 mm memiliki dua buah konektor, satu untuk dikoneksikan pada *microphone jack* (*line-in*) dan satu lagi untuk dikoneksikan pada *speaker jack* (*line-out*) pada komputer. *Computer headset* dikoneksikan pada komputer melalui sebuah *sound card* yang akan melakukan pengubahan sinyal digital pada komputer ke sinyal analog untuk *headset* dan sebaliknya.

Computer headset yang menggunakan koneksi USB memiliki kualitas suara yang lebih baik dibandingkan dengan menggunakan *jack* standar 3,5 mm, dan juga memiliki harga yang lebih mahal dibandingkan dengan *headset jack* standar 3,5 mm. Terdapat dua sub-tipe dari *headset* USB, yaitu *headset* dengan koneksi USB dan *direct* USB *headset*.

Headset dengan koneksi USB merupakan *computer headset* dengan *jack* standar 3,5 mm yang dibundel dengan sebuah USB adaptor. Salah satu ujung dari USB *adaptor* merupakan USB *plug* standar, namun pada ujung yang lain memiliki pasangan *jack* 3,5 mm untuk menghubungkan *computer headset* dengan *jack* standar 3,5 mm. Sedangkan *direct* USB *headset* biasanya memiliki *digital volume controller* yang juga termasuk modul USB. *Headset* USB jenis ini menawarkan kualitas suara yang lebih baik.

Pada *mobile phone headset* biasanya hanya terdapat satu buah *earpiece* dan satu buah *microphone*, atau sering disebut juga sebagai *mobile hands-free*. Namun, seiring dengan adanya fasilitas *music player* pada *mobile phone*, saat ini *mobile phone headset* didesain memiliki dua buah *earpiece* stereo yang digunakan untuk memfasilitasi pengguna mendengarkan musik.

2.8. *Noise*

Pada setiap pentransmisian data, sinyal yang diterima akan terdiri dari sinyal yang ditransmisikan oleh pengirim, sinyal yang termodifikasi oleh berbagai distorsi pada sistem transmisi, serta sinyal-sinyal yang tidak diinginkan yang masuk pada saat melakukan pentransmisian dan penerimaan data, atau sinyal-sinyal yang dianggap sebagai *noise*. *Noise* merupakan batasan terbesar pada keseluruhan performansi sistem komunikasi.

Noise dapat dibagi menjadi 4 kategori, antara lain:

1. Thermal noise.
2. Intermodulation Noise.
3. Crosstalk.
4. Impulse Noise.

1. *Thermal Noise*

Thermal noise terjadi karena adanya agitasi thermal pada elektron yang muncul pada semua perangkat elektronik dan media transmisi. *Thermal noise* terdistribusi secara seragam pada bandwith yang digunakan dalam sistem komunikasi. Karena itu *thermal noise* disebut juga sebagai *white noise*. Kepadatan spektral daya pada *thermal noise* hampir sama diseluruh spektrum frekuensi. *Thermal noise* tidak dapat dihilangkan.

2. *Intermodulation Noise*

Intermodulation noise muncul ketika beberapa sinyal yang memiliki frekuensi yang berbeda menggunakan satu media transmisi yang sama. Pengaruh dari *intermodulation noise* yaitu akan menghasilkan sinyal pada suatu frekuensi yang merupakan penjumlahan atau selisih dari dua frekuensi yang berbeda atau perkalian dari frekuensi tersebut.

Intermodulation noise dihasilkan dari sifat-sifat non-linier pada *transmitter*, *receiver*, dan/atau pada media transmisi. Jika diasumsikan input pada sebuah sistem linier adalah sinyal satu frekuensi, maka output yang dihasilkan juga berupa sinyal pada frekuensi yang sama; hanya amplitudo dan fasa yang berbeda dari sinyal input. Namun, pada sistem non-linier, jika sinyal input adalah satu frekuensi, f_a, maka output yang dihasilkan adalah sebuah sinyal beserta dan sinyal-sinyal kelipatan integer dari frekuensi sinyal input; (f_a, $2f_a$, $3f_a$, …). Intermodulasi terjadi ketika input pada sistem non-linier terdiri dari dua atau lebih frekuensi.

3. *Crosstalk*

Crosstalk merupakan kejadian ketika terdapat penggabungan jalur sinyal yang tidak diinginkan. Pada percakapan dengan menggunakan jalur telepon, *crosstalk* terjadi ketika percakapan telepon pada suatu jalur sinyal masuk dan terdengar ke dalam jalur sinyal percakapan telepon yang lain. *Crosstalk* dapat terjadi karena adanya *electrical coupling* antara kabel *twisted pairs* yang berdekatan, atau jalur kabel koaksial yang membawa sinyal lebih dari satu.

4. *Impulse Noise*

Impulse noise merupakan *noise* yang bersifat non-kontinu, terdiri atas pulsa pada durasi yang sangat singkat dan dengan besar amplitudo yang relatif cukup tinggi. Impulse noise memiliki beragam penyebab, termasuk karena gangguan sinyal elektromagnetik eksternal, seperti petir, dan kesalahan atau kecacatan pada sistem komunikasi.

Pada komunikasi data analog, *impulse noise* umumnya merupakan gangguan minor yang dapat diabaikan. Namun pada komunikasi data digital, *impulse noise* merupakan penyebab utama yang dapat mengakibatkan *error*.

Noise pada *headset* terjadi saat munculnya suara-suara yang tidak diinginkan dari lingkungan sekitar atau disebut juga sebagai *environmental noise*. *Noise* yang terjadi pada *headset* umumnya termasuk dalam kategori *acoustic noise*. *Acoustic noise* mencakup semua suara dalam domain akustik, baik yang diinginkan maupun tidak diinginkan.

Metode yang digunakan pada *headset* untuk mengurangi kemunculan suara-suara yang tidak dikehendaki tersebut disebut *active noise control* (ANC), atau dikenal juga dengan *noise cancellation*, *active noise reduction* (ANR) atau *antinoise*. Metode ini diterapkan pada masing-masing *headphone* dan *microphone*.

Noise-cancelling headphone akan mengurangi suara-suara yang tidak diinginkan (yaitu, *acoustic noise*) dengan metode *active noise control* (ANR). Pada dasarnya, metode ini melibatkan penggunaan *microphone*, ditempatkan di dekat telinga, dan rangkaian elektrik yang akan menghasilkan sebuah gelombang

suara "*antinoise*" dengan polaritas yang berlawanan dengan gelombang suara yang masuk ke microphone.

Sedangkan *noise-cancelling microphone* dirancang untuk memfilter suara yang masuk dari suara-suara yang tidak diinginkan. *Microphone* ini memiliki setidaknya dua buah *port* suara masuk; bagian depan berorientasi pada arah suara yang dikehendaki dan bagian lain pada arah lain untuk menerima *noise*. Diafragma *microphone* terletak diantara kedua *port*. Sumber suara hanya akan diterima oleh *port* bagian depan sehingga akan memberikan tekanan lebih kuat terhadap diafragma.

2.9. Rangkaian *Filter*

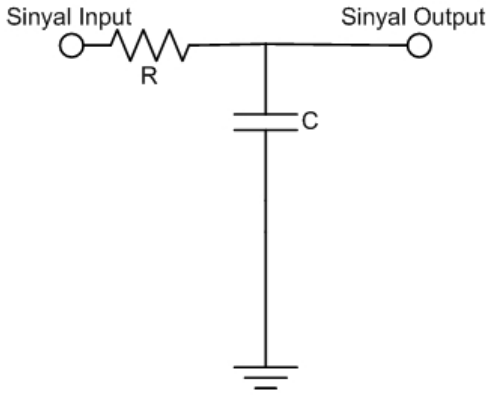
Rangkaian *filter* atau *filter* merupakan suatu rangkaian yang dibangun untuk menyaring satu frekuensi atau rentang frekuensi tertentu dari beberapa frekuensi yang berbeda dalam suatu rangkaian secara selektif. Beberapa jenis rangkaian *filter* sederhana antara lain:

1. *Low-Pass Filter*

Rangkaian *low-pass filter* merupakan rangkaian *filter* yang hanya akan meloloskan sinyal dengan frekuensi rendah di bawah frekuensi *cutoff*. Besar frekuensi *cutoff* ini ditentukan oleh Persamaan (2.12) berikut.

$$f_{cutoff} = \frac{1}{2\pi RC} \tag{2.12}$$

f_c merupakan besar frekuensi *cut-off* dalam satuan Hertz (Hz), R adalah nilai resistor (Ohm) dan C adalah nilai kapasitor (Farad). Dan bentuk rangkaian ini secara umum diilustrasikan seperti pada Gambar 2.20.



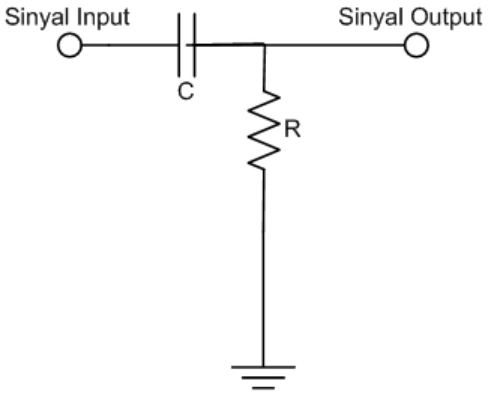
Gambar 2.20. Rangkaian *Low-Pass Filter*

2. *High-Pass Filter*

Rangkaian *high-pass filter* merupakan rangkaian *filter* yang berfungsi sebagai kebalikan dari *low-pass filter*, yaitu hanya akan meloloskan sinyal dengan frekuensi tinggi di atas frekuensi *cutoff*. Besar frekuensi *cutoff* ini ditentukan oleh Persamaan (2.13) berikut.

$$f_{cutoff} = \frac{1}{2\pi RC} \tag{2.13}$$

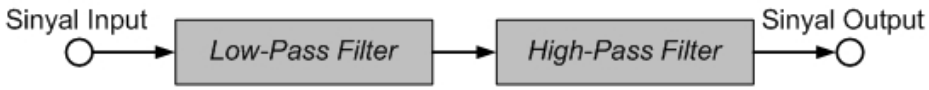
f_c merupakan besar frekuensi *cut-off* dalam satuan Hertz (Hz), R adalah nilai resistor (Ohm) dan C adalah nilai kapasitor (Farad). Dan bentuk rangkaian ini secara umum diilustrasikan seperti pada Gambar 2.21.



Gambar 2.21. Rangkaian *High-Pass Filter*

3. *Band-Pass Filter*

Rangkaian *band-pass filter* merupakan rangkaian *filter* yang ditujukan hanya untuk meloloskan sinyal yang berada pada rentang frekuensi tertentu. Rangkaian ini secara sederhana merupakan penggabungan rangkaian *low-pass filter* dan *high-pass filter* menjadi sebuah *filter*. Sebuah *band-pass filter* dapat diilustrasikan dengan menggunakan diagram blok seperti pada Gambar 2.22 berikut.



Gambar 2.22. Diagram Blok *Band-Pass Filter*