BAB II

DASAR TEORI

2.1 Anatomi Otot Pada Tangan Manusia

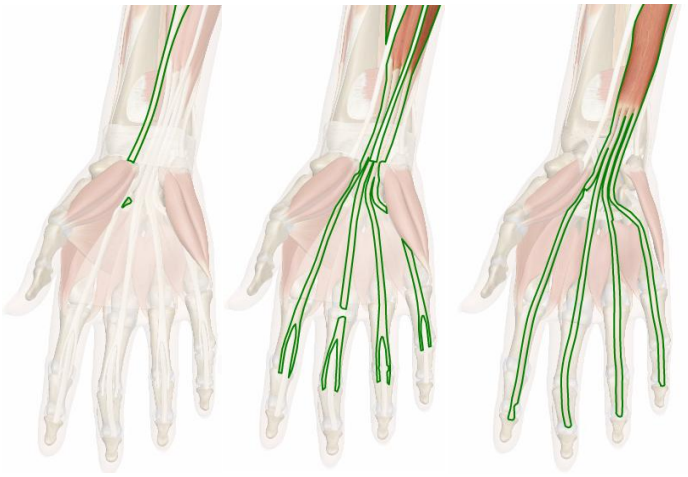
Anatomi adalah salah satu cabang dari ilmu biologi yang mempelajari secara mendalam bagian dari struktur tubuh makhluk hidup [4]. Kata anatomi ini berasal dari dua kata serapan bahasa Yunani, yaitu “ana” yang berarti ke atas dan “tomos” yang berarti memotong. Salah satu bagian dari ilmu anatomi ini adalah anatomi manusia, salah satunya yang akan dibahas adalah pada bagian tangan dan lengan tangan.

Tangan merupakan salah satu bagian dari tubuh manusia yang strukturnya begitu komplek dan rumit. Tangan tersusun dari banyak otot yang masing-masing melakukan pergerakan yang tidak sama/bermacam-macam. Lebih dari enam puluh jumlah otot di tangan dan lengan bawah yang saling berkerja untuk membuat gerakan yang bermacam-macam itu. Otot-otot pada tangan itu memiliki kelenturan, gerakan yang sangat presisi, dan kekuatan genggaman yang dibutuhkan untuk melakukan aktifitas mulai dari menulis, menekan sampai menggenggam benda [5]. Pada dasarnya gerakan tangan manusia terdiri atas.

1. *Flexion*: telapak tangan berbelok mendekati lengan tangan atau jari berbelok mendekati telapak tangan.
2. *Extension*: telapak tangan berbelok menjauhi dari lengan tangan atau jari berbelok menjauh dari telapak tangan.
3. *Abduction*: tangan bergerak ke samping; jempol mendekati sisi lengan tangan.
4. *Adduction*: tangan bergerak ke samping; jempol menjauhi sisi lengan tangan.

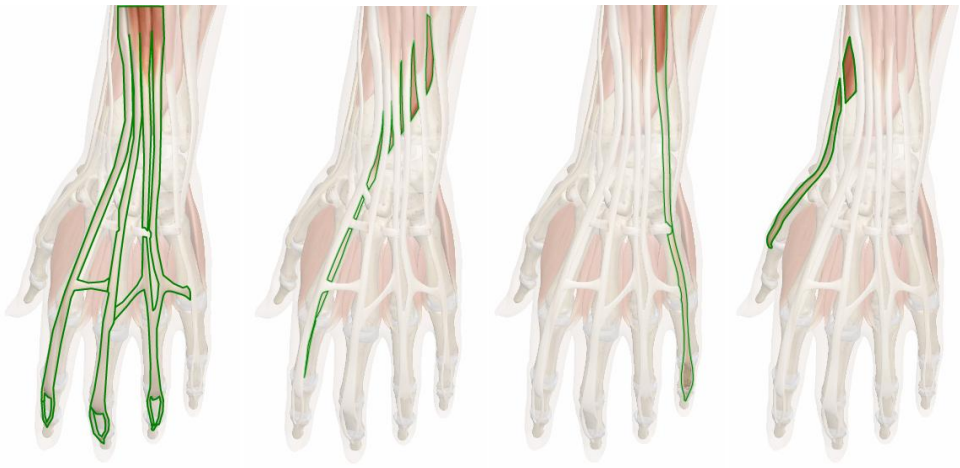
5. *Supination*: gerakan menengadahkan tangan.
6. *Pronation*: gerakan menelungkupkan tangan.

Pada **Gambar 2.1** dan **Gambar 2.2** berikut adalah anatomi otot tangan manusia yang berfungsi untuk menggerakan jari-jari tangan.



(a) (b) (c)

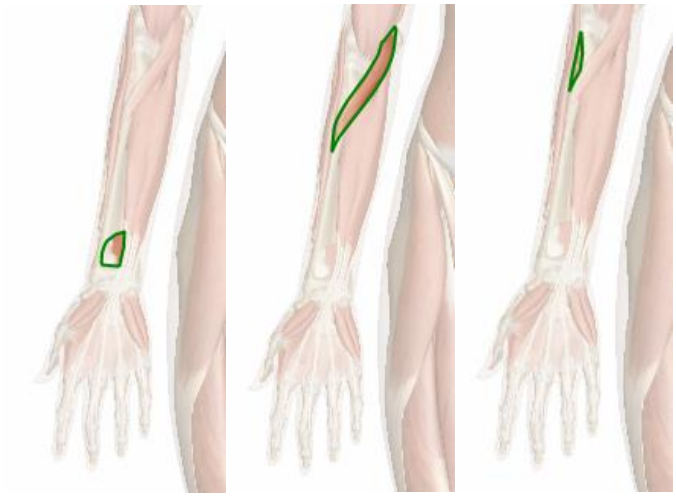
Gambar 2.1. (a) *Flexor Pollicis Longus,* (b) *Flexor Digitorum Superficialis,* (c) *Flexor Digitorum Profundus* [5]



(d) (e) (f) (g)

Gambar 2.2. (d) *Extensor Digitorum,* (e) *Extensor Indicis,* (f) *Extensor Digiti Minimi,* (g) *Extensor Pollicis Longus and Brevis* [5]

Dan pada **Gambar 2.3** berikut adalah anatomi otot pada lengan manusia yang berfungsi untuk melakukan gerakan pronasi dan supinasi.



(h) (i) (j)

Gambar 2.3. (h) *Pronator Quadratus*, (i) *Pronator Teres*, (j) *Supinator* [5]

Berikut adalah penjelasan fungsi masing-masing otot yang ditunjukan pada **Gambar 2.1**, **Gambar 2.2** dan **Gambar 2.3** sebelumnya.

(a). *Flexor pollicis longus*: fleksi jari jempol.

(b). *Flexor digitorum superficialis*: fleksi keempat jari (telunjuk, tengah, manis dan kelingking).

(c). *Flexor digitorum profundus*: fleksi keempat jari selain jempol.

(d). *Extensor digitorum*: ekstensi keempat jari selain jempol.

(e). *Extensor indicis*: ekstensi jari telunjuk.

(f). *Extensor digiti minimi*: ekstensi jari kelingking.

(g). *Extensor pollicis longus and brevis*: ekstensi jari jempol.

(h). *Pronator Quadratus*: pronasi lengan bawah.

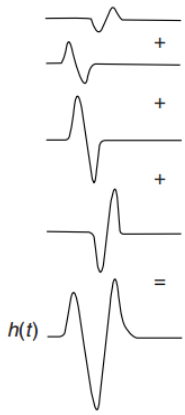
(i). *Pronator Teres*: pronasi lengan bawah.

(j). *Supinator*: supinasi lengan bawah.

2.2 *Electromyography* (EMG)

Electromyography (EMG) adalah teknik pengukuran sinyal elektris yang berhubungan dengan aktifitas otot. Aktifitas otot ini dapat berasal dari kontraksi otot yang disengaja maupun tidak disengaja. Aktifitas EMG dari kontraksi otot yang disengaja berhubungan dengan ketegangan otot. Bagian yang berfungsi dalam kontraksi otot adalah pada bagian motornya, yang tersusun atas satu α-motoneuron dan banyak urat otot. Satu bagian motor dapat memiliki 3-2000 urat otot. Urat otot ini akan berkontraksi ketika motor menghantarkan impuls listrik hingga mencapai kondisi depolarisasi. Depolarisasi di sini maksudnya adalah keadaan dimana urat otot sedang aktif. Depolarisasi ini akan menghasilkan sinyal elektris yang disebut sebagai tegangan/voltase [6].

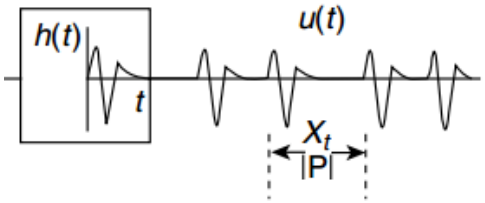
Sinyal elektris yang berasal dari kontraksi otot yang terdeteksi ini disebut *motor unit action potential* (MUAP). MUAP inilah yang merupakan bagian pokok dari sinyal EMG. Representasi dari MUAP ini dapat dilihat pada **Gambar 2.4** berikut.



Gambar 2.4. Representasi dari *Motor Unit Action Potential* (MUAP) [7]

Untuk menahan kontraksi otot tersebut, motor akan terus aktif berulang kali. Hasil sekuensial ini disebut *motor unit action potential train* (MUAPT) [7].

Gelombang MUAP yang berada di dalam MUAPT nilainya akan selalu konstan yang dapat dilihat pada **Gambar 2.5**. MUAPT ini dapat dirumuskan dengan persamaan sebagai berikut.



Gambar 2.5. *Motor Unit Action Potential Train* (MUAPT) [7]

$$u_i(t) = \sum_{k=1}^{n} [h_i(t - t_k)] \qquad (2.1)$$

$$t_k = \sum_{l=1}^{k} [x_l] \quad for\ k, l = 1,2,3, \dots, n \qquad (2.2)$$

Pada persamaan 2.1 dan 2.2 diatas t_k merepresentasikan lokasi waktu dari MUAP, x merepresentasikan interval sinyal dalam MUAPT. Sinyal EMG sendiri adalah kumpulan jumlah linier dari MUAPT [7] yang dirumuskan sebagai berikut.

$$m(t, F) = \sum_{i=1}^{p} [u_i(t, F)] \qquad (2.3)$$

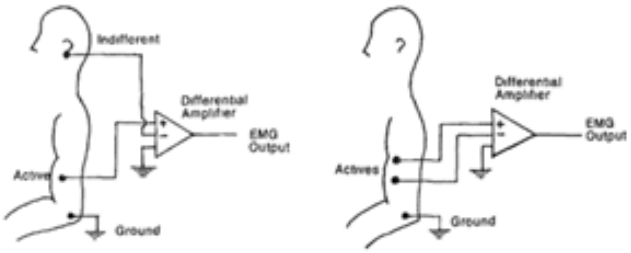
Untuk mendeteksi sinyal EMG ini digunakan sebuah sensor elektroda yang terbuat dari bahan Ag/AgCl (Perak Klorida). Ada dua jenis elektroda, yaitu elektroda permukaan (*surface electrode*) dan elektroda jarum atau kabel (*wire or needle electrode*) yang ditunjukan pada **Gambar 2.6**.



(a) (b)

Gambar 2.6. (a) *Surface Electrode*, (b) *Needle Electrode* [8]

Konfigurasi pemasangan elektroda ada dua, yaitu secara unipolar dan bipolar yang dapat dilihat pada **Gambar 2.7**. Secara unipolar perekaman sinyal EMG dilakukan dengan memasang satu elektroda pada otot yang akan diukur dan elektroda lain diletakan di bagian yang minim gerakan sebagai referensi. Sedangkan untuk konfigurasi bipolar dua elektroda diletakan berdekatan di bagian otot yang akan diukur [8].

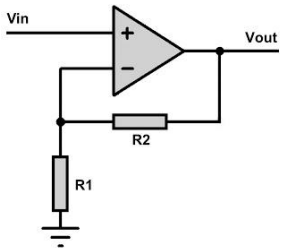


(a) (b)

Gambar 2.7. Konfigurasi Pemasangan Elektroda. (a) Unipolar, (b) Bipolar [8]

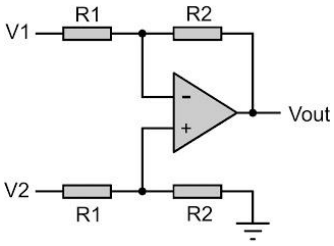
2.3 Penguat Operasional

Beberapa jenis dari penguat operasional adalah penguat instrumentasi dan penguat *non-inverting*. Penguat instrumentasi adalah salah satu penerapan dari fungsi penguat operasional (op-amp). Penguat ini bersifat linier, yaitu *output* sinyal yang dihasilkan dari penguat ini nilainya tidak berubah/konstan. Penguat instrumentasi ini adalah gabungan dari penguat *non-inverting* dan penguat *differensial*. Penguat *non-inverting* merupakan penguat yang berfungsi memperkuat sinyal tanpa membalik fasa dari sinyal tersebut, sedangkan penguat *differensial* merupakan penguat yang mampu memperkuat sinyal kecil yang berada dalam sinyal yang jauh lebih besar [9]. Pada **Gambar 2.8** dan **Gambar 2.9** berikut adalah rangkaian penguat *non-inverting* dan penguat *differensial*.



$$Vout = \left(\frac{R2}{R1} + 1\right).Vin \qquad (2.4)$$

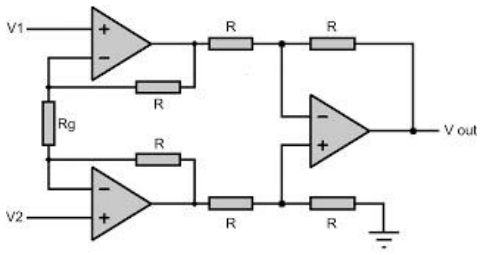
Gambar 2.8. Penguat *Non-inverting* [9]



$$Vout = \left(\frac{R2}{R1} + 1\right).(V2 - V1) \qquad (2.5)$$

Gambar 2.9. Penguat *Differensial* [9]

Dari dua rangkaian penguat di atas, yaitu penguat *non-inverting* dan penguat *differensial* dapat digabung menjadi satu rangkaian, yaitu yang disebut penguat instrumentasi yang ditunjukan pada **Gambar 2.10** berikut.



Gambar 2.10. Penguat Instrumentasi [9]

Dari persamaan 2.4 dan persamaan 2.5 sebelumnya didapat.

$$Vout = \left(\frac{2R}{Rg} + 1\right).(V1 - V2) \qquad (2.6)$$

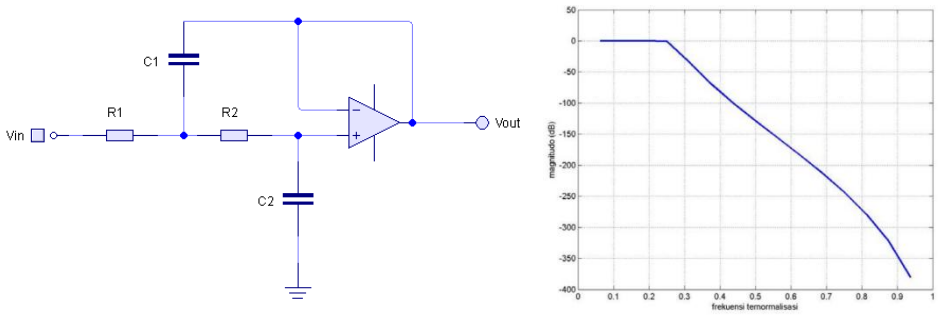
2.4 Filter

Filter adalah sebuah jaringan elektris yang merubah karakteristik amplitudo dan/atau fasa suatu sinyal tetapi tidak mengubah frekuensinya. Idealnya, *filter* tidak menambah frekuensi yang baru pada sinyal masukannya. *Filter* biasanya digunakan di dalam sistem elektronik untuk meloloskan sinyal pada *range* frekuensi tertentu dan menolak sinyal pada *range* frekuensi yang lainnya [10].

Berdasarkan sifatnya, *filter* terbagi menjadi dua macam, yaitu *filter* pasif (*filter* yang hanya terdiri dari komponen pasif, seperti kapasitor, resistor atau induktor) dan *filter* aktif (*filter* yang berisi komponen aktif, seperti transistor dan op-amp). Kemudian berdasarkan jenisnya, *filter* terbagi menjadi empat macam, yaitu.

1. *Low Pass Filter* (LPF)

   LPF merupakan *filter* yang meloloskan frekuensi rendah dan meredam/menolak frekuensi tinggi. Pada **Gambar 2.11** adalah contoh dari rangkaian aktif *low pass filter Butterworth* dengan menggunakan desain *Sallen-key* dan respon frekuensinya.



Gambar 2.11. Rangkaian Aktif *Low Pass Filter Butterworth Sallen-key* Orde 2 dan respon frekuensinya

Jika pada rangkaian di atas parameter R1=R2=R maka dapat dirumuskan.

$$2\pi . f_c = \frac{1}{R^2 . C_1 . C_2} \qquad (2.7)$$

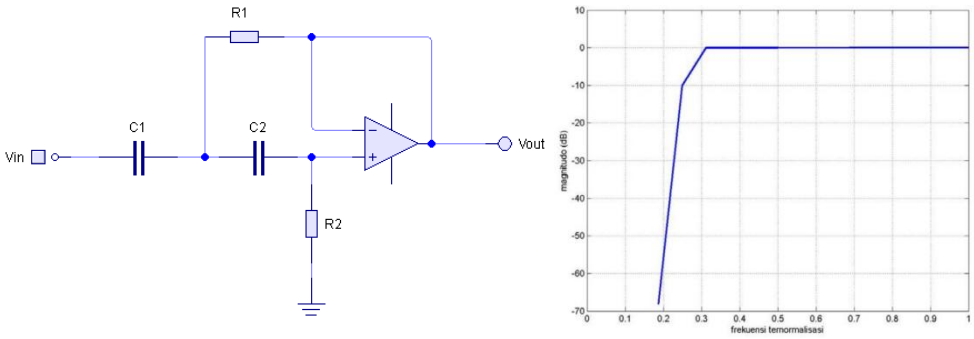
$$Q = \frac{1}{2} . \sqrt{\frac{C_1}{C_2}} \qquad (2.8)$$

Dimana: f_c = frekuensi *cut-off* (Hz)

Q = faktor kualitas.

2. *High Pass Filter* (HPF)

HPF merupakan kebalikan dari LPF, yaitu *filter* yang meloloskan frekuensi tinggi dan meredam/menolak frekuensi rendah. Pada **Gambar 2.12** adalah contoh dari rangkaian aktif *high pass filter Butterworth* dengan menggunakan desain *Sallen-key* dan respon frekuensinya.



Gambar 2.12. Rangkaian Aktif *High Pass Filter Butterworth Sallen-key* Orde 2 dan respon frekuensinya

Jika pada rangkaian di atas parameter C_1=C_2=C maka dapat dirumuskan.

$$2\pi . f_c = \frac{1}{C^2 . R_1 . R_2} \qquad (2.9)$$

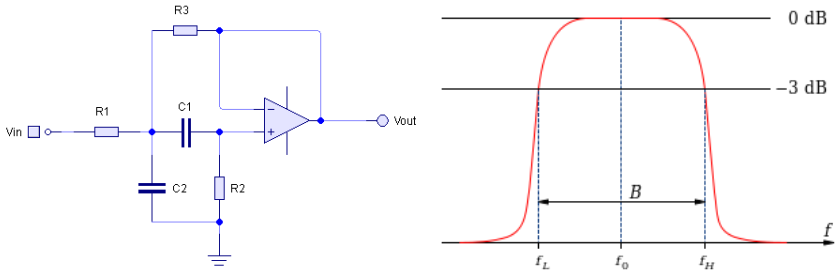
$$Q = \frac{1}{2} . \sqrt{\frac{R_2}{R_1}} \qquad (2.10)$$

Dimana: f_c = frekuensi *cut-off* (Hz)

Q = faktor kualitas.

3. *Band Pass Filter* (BPF)

BPF merupakan gabungan dari LPF dan HPF, yaitu *filter* yang meloloskan frekuensi dengan *range* tertentu, dan yang lainnya akan diredam. *Range* frekuensi ini biasanya disebut sebagai *bandwidth*. Pada **Gambar 2.13** adalah contoh dari rangkaian aktif *band pass filter Butterworth* dengan menggunakan desain *Sallen-key*.



Gambar 2.13. Rangkaian Aktif *Band Pass Filter Butterworth Sallen-key* Orde 2 dan respon frekuensinya

Jika pada rangkaian di atas parameter R1=R3=R, R2=2.R dan C1=C2=C, maka dapat dirumuskan.

$$2\pi . f_0 = \frac{1}{C.R} \qquad (2.11)$$

$$Q = \frac{f_0}{BW} \qquad (2.12)$$

$$BW = f_H - f_L \qquad (2.13)$$

$$f_0 = \sqrt{BW} \qquad (2.14)$$

Dimana: f_0 = frekuensi tengah (Hz)

Q = faktor kualitas

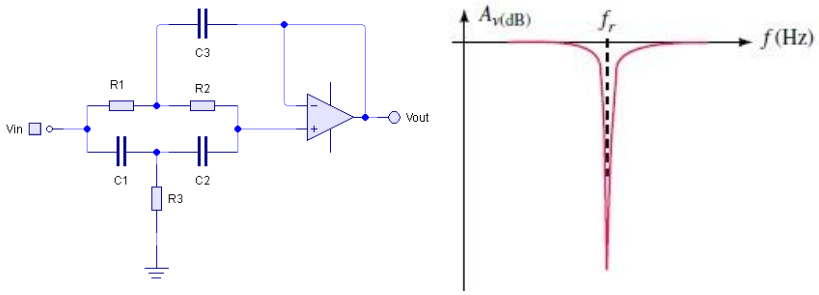
BW = *bandwidth* (lebar frekuensi) (Hz)

f_H = frekuensi tinggi (Hz)

f_L = frekuensi rendah (Hz).

4. *Band Reject Filter/ Band Stop Filter/ Notch Filter*

Notch filter merupakan *filter* yang menolak/meredam frekuensi tertentu dan meloloskan frekuensi yang lainnya. Pada **Gambar 2.14** adalah contoh dari rangkaian aktif *Notch filter Butterworth* dengan menggunakan desain *Sallen-key*.



Gambar 2.14. Rangkaian Aktif *Notch Filter Butterworth Sallen-key* Orde 2 dan respon frekuensinya

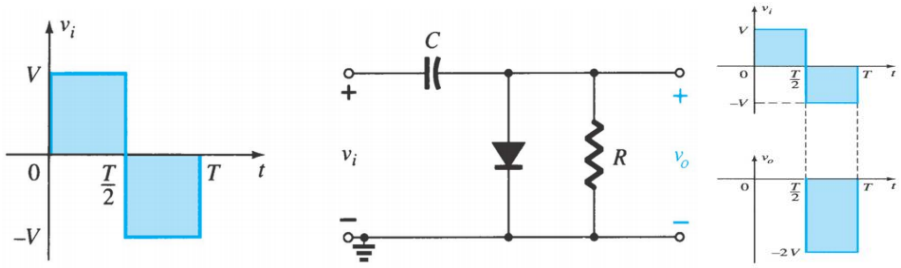
Jika pada rangkaian di atas parameter R1=R2=R, R3=R/2, C1=C2=C dan C3=2.C, maka dapat dirumuskan.

$$2\pi . f_r = \frac{1}{C.R} \qquad (2.15)$$

f_r = frekuensi tengah (Hz).

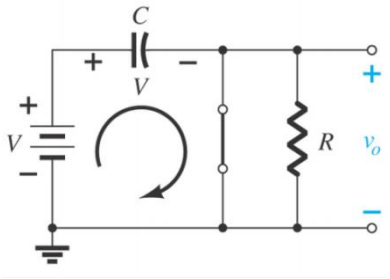
2.5 *Clamper*

Clamper atau penggeser adalah salah satu rangkaian elektronika yang terdiri dari dioda, resistor dan kapasitor yang menggeser gelombang ke level dc yang lain tanpa mengubah bentuk dari sinyal tersebut. *Clamper* terbagi menjadi dua jenis, yaitu *clamper* positif yang menggeser gelombang ke arah positif dan *clamper* negatif yang menggeser gelombang ke arah negatif [11]. Pada **Gambar 2.15** berikut adalah contoh dari rangkaian *clamper* negatif dan **Gambar 2.18** adalah contoh dari rangkaian *clamper* positif.



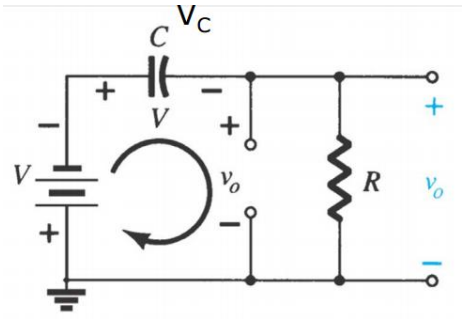
Gambar 2.15. Rangkaian *Clamper* Negatif dan Hasilnya [11]

Pada **Gambar 2.15** di atas jika dioda dianggap ideal, pada saat dioda dibias maju (0 sd T/2), dioda akan menjadi *short circuit* dan kapasitor akan mengisi hingga mencapai tegangan V, seperti ditunjukan pada **Gambar 2.16**. Maka dapat dicari V_o= V + (-V_c) = 0.

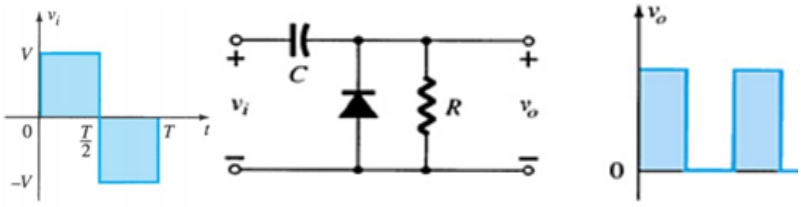


Gambar 2.16. *Clamper Negatif* Saat Bias Maju [11]

Dan pada saat dioda dibias mundur (T/2 sd T), dioda akan menjadi *open circuit* dan kapasitor akan melepaskan tegangan yang sudah diisinya, seperti yang ditunjukan pada **Gambar 2.17**. Maka dapat dicari V_o = -V + (-V_c).



Gambar 2.17. *Clamper Negatif* Saat Bias Mundur [11]



Gambar 2.18. Rangkaian *Clamper* Positif dan Hasilnya [11]

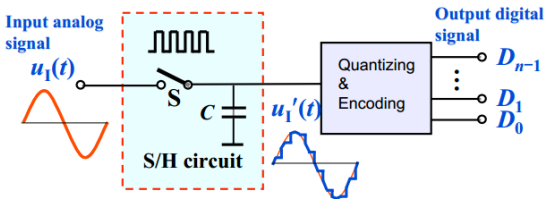
Untuk *clamper* positif seperti **Gambar 2.18** di atas, saat dioda dibias mundur (0 sd T/2), maka dioda akan mengalami *open circuit* dan kapasitor akan melepas tegangan yang sudah diisi. Maka dapat dirumuskan $V_o = V_i + (-V_c)$. Dan pada saat dioda dibias maju (T/2 sd T), maka dioda akan mengalami *short* circuit dan kapasitor akan mengisi hingga mencapai tegangan V_i. Maka dapat dirumuskan $V_o = V_i + V_c$. Nilai R dan C harus dipilih hingga ketika kapasitor diisi tidak terlalu cepat dilepaskan, yang dirumuskan sebagai berikut.

$$\tau = R.C \qquad (2.16)$$

2.6 Analog to Digital Converter (ADC)

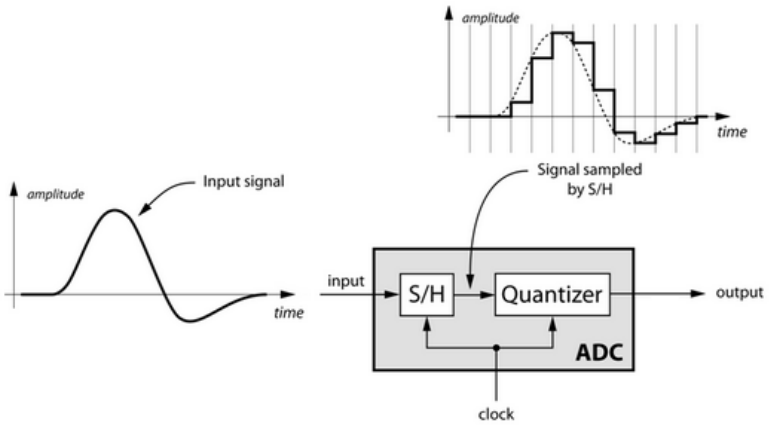
ADC adalah rangkaian elektronika yang sudah terintegrasi yang mengubah sebuah sinyal dari bentuk analog (*continuous*) ke bentuk digital (diskrit). Sinyal analog adalah jumlah yang terukur secara langsung, sedangkan sinyal digital hanya memiliki dua keadaan, yaitu *high* dan *low* atau dalam binary, yaitu 1 dan 0. ADC banyak dibutuhkan beberapa diantaranya karena jika sinyal dalam keadaan bentuk digital tidak akan mudah terkena *noise* atau gangguan, mikroprosesor hanya bisa melakukan proses dalam sinyal digital, dan ADC lebih memudahkan dalam penanganan data dan proses pengolahan sinyal [12]. ADC banyak digunakan dimana sinyal analog harus diproses, disimpan atau dikirim melalui bentuk digital.

Di dalam ADC terjadi dua proses, yaitu *sampling and holding* (S/H) dan *quantizing and Encoding* (Q/E) yang ditunjukan pada **Gambar 2.19**.



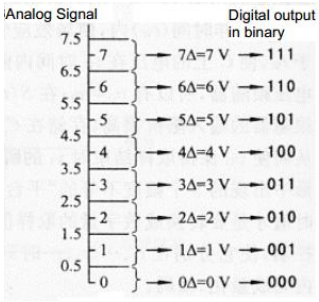
Gambar 2.19. Proses ADC [12]

Proses S/H ini ditunjukan pada **Gambar 2.20.**



Gambar 2.20. Proses *Sampling and Holding* (S/H) dalam ADC [13]

Proses *holding* berguna untuk keakurasian dari konversi A/D. Dan menurut teory Nyquist, *sampling* harus sedikitnya dua kali dari data frekuensi tertinggi dari sinyal analog. Sedangkan proses Q/E ini ditunjukan pada **Gambar 2.21**.



Gambar 2.21. Proses *Quantizing and Encoding* (Q/E) ADC [12]

Dalam proses *quantizing* dilakukan proses partisi dari sinyal referensi ke dalam kuantum angka-angka diskrit, lalu menyamakan sinyal masukan ke dalam kuantum yang benar. Sedangkan dalam proses *encoding* dilakukan proses memasukan kode digital ke dalam tiap kuantum. Dalam proses *quantizing* dibutuhkan nilai resolusi seperti dirumuskan sebagai berikut.

$$\Delta V = \frac{V_r}{2^N} \qquad (2.17)$$

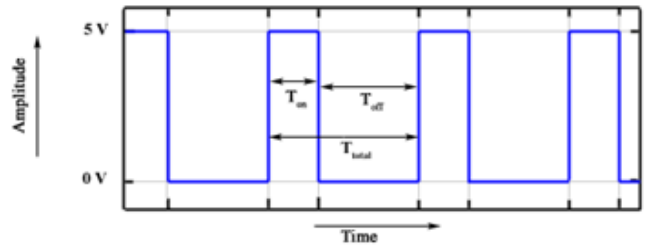
Dimana: Vr = tegangan referensi

N = jumlah bit

ΔV = resolusi.

2.7 Pulse Width Modulation (PWM)

Pulse width modulation (PWM) secara umum adalah salah satu teknik yang digunakan untuk mengatur lebarnya sinyal dalam satu perioda seperti pada **Gambar 2.22**. PWM dapat diperoleh dengan mengubah *duty cycle* suatu sinyal. *Duty Cycle* adalah persentasi perbandingan antara lamanya waktu sinyal dalam keadaan *high* dengan perioda suatu sinyal [14].



Gambar 2.22. *Pulse Width Modulation* (PWM) [14]

Pada **Gambar 2.22** di atas dapat ditentukan beberapa rumus sebagai berikut.

$$T_{total} = T_{on} + T_{off} \qquad (2.18)$$

$$D = \frac{T_{on}}{T_{on} + T_{off}} x100\% = \frac{T_{on}}{T_{total}} x100\% \qquad (2.19)$$

Dimana: T_{total} = perioda

T_{on} = lama sinyal *high*

T_{off} = lama sinyal *low*

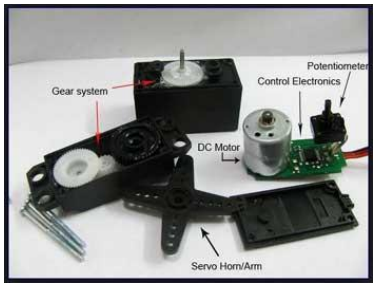
D = *duty cycle*.

2.8 Motor Servo

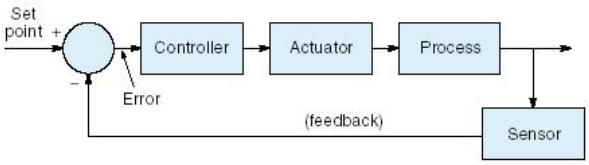
Motor servo sebenarnya adalah motor DC sederhana yang dikontrol dengan pengaturan sudut rotasi tertentu dengan bantuan penambahan mekanisme servo (sistem kontrol umpan balik *loop* tertutup). Penggunaan motor servo ini biasanya digunakan pada mobil mainan remot kontrol untuk mengendalikan arah atau gerakan dan juga biasanya digunakan sebagai motor untuk menggerakan CD atau DVD *player*. Alasan utama untuk menggunakan motor servo ini adalah karena dapat memberikan sudut yang presisi, misalnya dapat berputar sebanyak yang kita mau kemudian berhenti dan menunggu sinyal selanjutnya untuk melakukan aksi selanjutnya. Tidak seperti motor elektronik normal yang bergerak ketika diberikan *supply power* lalu berputar secara terus menerus sampai kita mematikan *supply power* tersebut. Kita tidak dapat mengontrol proses rotasi dari motor elektronik, tetapi hanya bisa mengontrol kecepatan rotasinya dan menghidupkan atau mematikannya [15].

Sistem servo biasanya terdiri dari 3 komponen dasar, yaitu *controller*, sensor, dan sistem umpan balik yang ditunjukan pada **Gambar 2.23**. Mekanisme servo ini menggunakan sistem umpan balik otomatis yang dapat dilihat pada **Gambar 2.24**, pada bagian *controller* dilakukan proses perbandingan sinyal

keluaran dengan sinyal referensi dari sinyal masukan, dan untuk sensor yang digunakan pada motor servo ini biasanya adalah potensiometer.



Gambar 2.23. Bagian-bagian dari Motor Servo [15]



Gambar 2.24. Sistem Umpan Balik *Loop* Tertutup Otomatis pada Motor Servo [15]

Ketika sinyal masukan sebagai referensi dimasukan ke dalam sistem, kemudian akan dilakukan proses perbandingan dengan sinyal keluaran yang dihasilkan oleh sensor (potensiometer) dan kemudian hasil perbandingan disebut sinyal ketiga yang dihasilkan dari sistem umpan balik. Sinyal ketiga ini adalah sebagai sinyal input yang masuk ke *controller*. Proses ini akan berulang terus sampai *controller* mencapai nilai yang diinginkan (*set point*) sehingga tidak ada lagi proses perbandingan sinyal input dan sinyal keluaran.

2.9 Fitur EMG

Para peneliti telah menggunakan berbagai macam fitur EMG yang biasanya digunakan sebagai masukan/*input* dalam proses klasifikasi. Ada tiga tipe dari ekstraksi fitur EMG dalam beberapa domain, yaitu pada domain waktu, domain

frekuensi dan domain waktu-frekuensi [16]. Berikut adalah beberapa jenis dari fitur EMG dalam domain waktu yang akan digunakan sebagai *input*/masukan dalam proses klasifikasi [17].

a. *Integrated* EMG (IEMG)

IEMG dihitung berdasarkan penjumlahan dari nilai absolut dari amplitudo sinyal EMG. Pada umumnya, IEMG digunakan sebagai indeks untuk mendeteksi aktivitas otot sebagai pengontrol devais. IEMG dapat dirumuskan sebagai berikut.

$$IEMG = \sum_{n=1}^{N} |x_n| \qquad (2.20)$$

b. *Root Mean Square* (RMS)

Root mean square (RMS) adalah akar dari rata-rata nilai data yang memiliki ‘n’ sampel. Vrms ini digunakan khususnya jika data memiliki nilai positif dan negatif. Berikut adalah persamaan yang digunakan dalam menghitung nilai Vrms.

$$V_{rms} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} {x_n}^2} \qquad (2.21)$$

c. *Mean Absolute Value* (MAV)

MAV adalah rata-rata dari nilai absolute dari sinyal EMG. MAV adalah salah satu *feature* yang paling banyak digunakan dalam proses klasifikasi sinyal EMG. Berikut adalah persamaannya.

$$MAV = \frac{1}{N}\sum_{n=1}^{N} |x_n| \qquad (2.22)$$

d. *Simple Square Integral* (SSI)

SSI menggunakan energi dari sinyal EMG sebagai fiturnya yang dirumuskan sebagai berikut.

$$SSI = \sum_{n=1}^{N} |x_n|^2 \qquad (2.23)$$

e. *Variance of* EMG (VAR)

VAR menggunakan tenaga dari sinyal EMG sebagai fiturnya. VAR adalah nilai rata-rata kuadrat dari nilai deviasi variabel tersebut. VAR dapat dirumuskan sebagai berikut.

$$VAR = \frac{1}{N-1} \sum_{n=1}^{N} {x_n}^2 \qquad (2.24)$$

f. *Zero Crossing* (ZC)

ZC adalah banyaknya nilai dari amplitudo sinyal EMG yang melewati nilai nol pada aksis y. Dalam fitur EMG, kondisi *threshold* digunakan untuk mengurangi dampak dari *noise*. ZC ini dirumuskan sebagai berikut.

$$ZC = \sum_{n}^{N-1} [sgn(x_n . x_{n+1}) \cap |x_n - x_{n+1}| \geq \text{threshold}] \qquad (2.25)$$

$$sgn(x) = \begin{cases} 1, & if\ x \geq threshold \\ 0, & otherwise \end{cases}$$

Dimana: IEMG = nilai *integrated* EMG

RMS = nilai *root mean square*

MAV = nilai *mean absolute*

SSI = nilai *simple square integral*

VAR = nilai *variance of* EMG

ZC = nilai *zero crossing*

xn = amplitudo pada sampel ke-n

n = nomor urutan sampel

N = jumlah sampel.

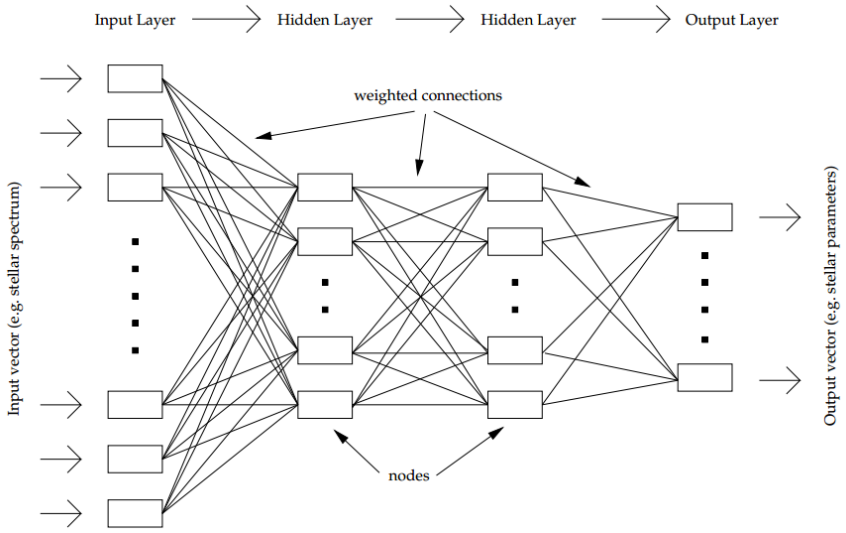
2.10 *Artificial Neural Network* (ANN)

Artificial neural network (ANN) diperkenalkan secara sederhana pada tahun

1943 oleh Waren McCulloch dan Walter Pits melalui beberapa komputasi menggunakan neuron-neuron sederhana dan mengembangkannya menjadi sebuah sistem baru (sistem neural) yang mempunyai kemampuan komputasi yang lebih baik. *Artificial neural network* (ANN) adalah salah satu bidang yang menggunakan *artificial intelligence* (kecerdasan buatan), yaitu dengan memanfaatkan mesin seperti komputer yang meniru kecerdasan manusia untuk memecahkan suatu persoalan tertentu.

ANN ini tersusun dari beberapa elemen pemroses yang saling berhubungan (*neurons*) yang bekerja sama untuk menyelesaikan suatu masalah tertentu. ANN ini sama seperti manusia, yaitu belajar dari contoh/pola. ANN ini biasa digunakan untuk aplikasi tertentu, seperti pengenalan pola (*pattern recognition*) atau klasifikasi data (*data classification*), melalui proses pembelajaran (*learning process*) [18].

Arsitektur ANN terdiri dari sejumlah *neurons* yang disusun dalam lapisan-lapisan (*layers*). *Neurons* tersebut terdiri dari *input*, *hidden* dan *output* yang ditunjukan pada **Gambar 2.25** berikut.



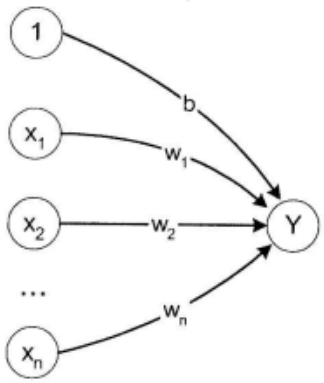
Gambar 2.25. Arsitektur *Artificial neural network* (ANN)

Model pembelajaran pada ANN ini hadir dalam dua bentuk, yaitu.

1. *Supervised learning*, yaitu model pembelajaran yang membutuhkan target sebagai acuan atau pembelajaran.
2. *Unsupervised learning*, yaitu model pembelajaran yang tidak membutuhkan target sebagai acuan arah pembelajaran.

Beberapa dari jenis *Artificial neural network* (ANN*)* yang biasa digunakan adalah *perceptron* dan *backpropagation*. Model jaringan yang hanya akan dibahas adalah model jaringan *perceptron*. Model jaringan *perceptron* dikembangkan oleh Rosenblatt (1962) dan Minsky-Papert (1969). Model tersebut merupakan model yang memiliki aplikasi dan pelatihan yang paling baik pada era tersebut.

Arsitektur jaringan *perceptron* mirip dengan arsitektur jaringan Hebb yang ditunjukan pada **Gambar 2.26** berikut.



Gambar 2.26. Arsitektur Jaringan *Perceptron*

Jaringan terdiri beberapa unit masukan x_i yang dikalikan dengan sejumlah bobot w_i (ditambah sebuah bias b) dan memiliki sebuah unit keluaran Y. Berikut adalah algoritma yang digunakan dalam pembelajaran dengan model jaringan *perceptron*.

1. Meng-inisialisasi semua bobot dan bias dengan nilai yang kecil, biasanya

diberi nilai antara -0.5 ≤ w_i ≤ 0.5 dan -0.5 ≤ b ≤ 0.5. Tentukan laju pembelajaran α yang biasanya diberikan nilai 0 ≤ α ≤ 1, dan tentukan nilai *threshold* θ yang digunakan.

2. Mengeset masukan x_i dimana i=1, 2, 3, . . ., n. x_i ini adalah pola masukan ke-i. Setiap masukan x_i mempunyai pasangan nilai target t_i.
3. Menghitung keluaran net dari pola masukan x_i dengan rumus sebagai berikut.

$$net = \sum_i x_i w_i + b \qquad (2.26)$$

4. Hasil keluaran net tersebut kemudian dilakukan proses fungsi aktivasi untuk mendapatkan nilai *output* -1, 0 atau 1.

$$y = f(net) = \begin{cases} 1 & jika\ net > \theta \\ 0 & jika - \theta \leq net \leq \theta \\ -1 & jika\ net < -\theta \end{cases} \qquad (2.27)$$

5. Bila *output* ≠ target, maka perbaikan bobot dan bias dilakukan, jika tidak lewati dan lanjut ke pasangan masukan x_i dan target t_i berikutnya.

$$w_{i+1} = w_i + \Delta w \qquad \text{dengan } \Delta w = \alpha.t.x_i \qquad (2.28)$$

$$b_{i+1} = b_i + \Delta b \qquad \text{dengan } \Delta b = \alpha.t \qquad (2.29)$$

6. Perbaikan bobot dan bias dilakukan berulang-ulang dari pasangan masukan x_i dan target t_i sampai semua $y_i = t_i$.

Ada beberapa hal yang perlu diperhatikan dalam algoritma tersebut:

a. Bias (b_i) adalah salah satu node input dari ANN yang sifatnya khusus, karena selalu bernilai 1. Penggunaan bias dapat mempercepat pelatihan, hal ini dapat terjadi karena keberadaan bias yang berguna sebagai faktor koreksi terhadap kecukupan variabel-variabel *input* yang telah ditetapkan.
b. Bobot (w_i) adalah suatu nilai yang merepresentasikan ingatan manusia terhadap pembelajaran yang dilakukan.

c. Iterasi dilakukan terus hingga semua pola memiliki keluaran jaringan yang sama dengan targetnya (jaringan sudah memahami pola). Iterasi tidak berhenti setelah semua pola dimasukkan seperti yang terjadi pada model Hebb.

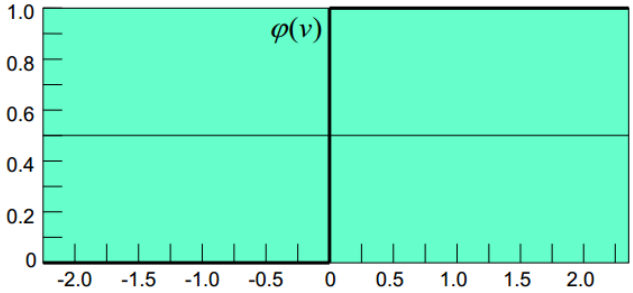
d. Pada langkah 3 (perbaikan bobot) hanya dilakukan pada pola yang mengandung kesalahan (keluaran jaringan ≠ target). Perubahan tersebut merupakan hasil kali unit masukan dengan target dan laju pemahaman.

e. Kecepatan iterasi ditentukan pula oleh laju pemahaman α yang dipakai. Semakin besar harga α, semakin sedikit iterasi yang diperlukan. Akan tetapi jika α terlalu besar, maka akan merusak pola yang sudah benar shingga pemahaman menjadi lambat.

Pada ANN digunakan beberapa fungsi aktivasi untuk mendapatkan keluaran/*output*. Berikut adalah beberapa fungsi aktivasi yang biasa digunakan dalam proses pembelajaran ANN [19].

1. Fungsi *threshold* atau fungsi *Heaviside* yang ditunjukan pada **Gambar 2.27** berikut.

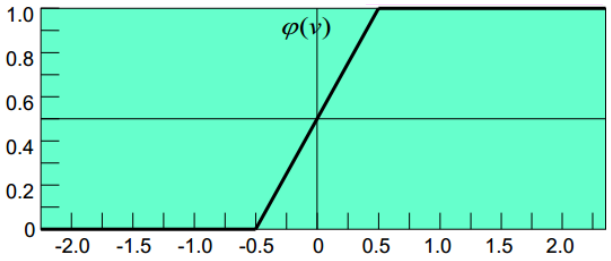
$$\varphi(v) = \begin{cases} 1 \; if \; v \geq threshold \\ 0 \; if \; v < threshold \end{cases}$$



Gambar 2.27. Fungsi *Threshold* atau Fungsi *Heaviside* [19]

2. Fungsi *Piecewise-Linier* yang ditunjukan pada **Gambar 2.28** berikut.

$$\varphi(v) = \begin{cases} 1 & if\ v \geq threshold \\ v & if\ threshold > v > -threshold \\ 0 & if\ v \leq -threshold \end{cases}$$

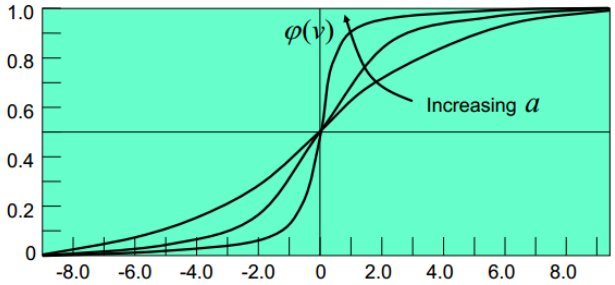


Gambar 2.28. Fungsi *Piecewise-Linier* [19]

3. Fungsi sigmoid yang ditunjukan pada **Gambar 2.29** berikut.

- Sigmoid Function: $\varphi(v) = \frac{1}{1+\exp(-av)}$

where a is the slope parameter of the sigmoid function.



Gambar 2.29. Fungsi Sigmoid [19]

2.11 *Filter* Digital

Filter digital adalah proses komputasi (algoritma) yang mengubah satu sekuen angka x[n] yang merepresentasikan *input* ke sekuen y[n] yang merepresentasikan *output*. Komputasi di sini dapat berupa fungsi integrasi, diferensiasi dan estimasi. *Filter* digital memiliki banyak kelebihan dibandingkan *filter* analog, baik dalam performa yang lebih tinggi dengan daerah transisi yang lebih kecil, ketahanan, serta fleksibilitas dalam menentukan rentang kerjanya (Smith, 1997: 327).

Ada dua metode yang digunakan untuk mendesain sebuah *filter* digital. Metode pertama menggunakan proses konvolusi antara sinyal *input* dengan *impulse response* dari *filter* yang dikehendaki, *filter* jenis ini disebut *filter* FIR (*finite impulse response*). Metode kedua adalah dengan menggunakan proses rekursif, yang merupakan lanjutan dari proses konvolusi. Bila dalam proses konvolusi perhitungan dilakukan dengan hanya menggunakan sampel *input* saja, maka dalam proses rekursif perhitungan dilakukan dengan menggunakan sampel *input* dijumlahkan dengan sampel *output* sebelumnya. Hal ini membuat *impulse response filter* menjadi sangat panjang mendekati titik tak hingga (*infinity*), oleh karena itu *filter* jenis ini disebut *filter* IIR (*infinite impulse response*) [20].

Keuntungan *filter* IIR ini antara lain membutuhkan koefisien *filter* yang lebih sedikit untuk respon frekuensi yang curam, sehingga dapat mengurangi lama waktu komputasi. Fungsi transfer *filter* IIR ini dapat dirumuskan sebagai berikut.

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \qquad (2.30)$$

Dimana: H(z) : fungsi transfer dari *filter* IIR

$a_1, a_2, ..., a_N$: koefisien *feed back* dari *filter* IIR

$b_1, b_2, ..., b_N$: koefisien *feed forward* dari *filter* IIR

Dari fungsi transfer *filter* IIR pada rumus 2.30 di atas dapat dicari persamaan bedanya yang dirumuskan sebagai berikut.

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \qquad (2.31)$$

Kemudian berikut adalah algoritma yang digunakan untuk mendesain sebuah contoh *filter* digital IIR *band pass Butterworth* dengan menggunakan metode *billinear z-transform* (BZT).

1. Menentukan spesifikasinya, seperti frekuensi *passband* (f_p) dan frekuensi *stopband* (f_s), frekuensi *sampling* ($F_{sampling}$), *ripple pass band* maksimum (r_p) dan atenuasi minimum *stop band* (r_s).
2. Melakukan perhitungan frekuensi *prewarp* dari digital ke analog dengan rumus sebagai berikut.

$$w_p = \tan\left(\frac{\pi . f_p}{f_{sampling}}\right) \qquad (2.32)$$

$$w_s = \tan\left(\frac{\pi . f_s}{f_{sampling}}\right) \qquad (2.33)$$

3. Men-transformasikan ke dalam *low pass prototype* dengan rumus sebagai berikut.

$$w_A = \frac{{w_s}^2 - (w_{p1} . w_{p2})}{w_s (w_{p2} - w_{p1})} \qquad (2.34)$$

4. Mencari nilai orde dari *low pass prototype* yang didapat.

$$w_A = \min(|w_{A1}|, |w_{A2}|) \qquad (2.35)$$

$$n \geq \frac{\log\left(\frac{\left(10^{0,1.|r_s|} - 1\right)}{\left(10^{0,1.|r_p|} - 1\right)}\right)}{2.\log(w_A)} \qquad (2.36)$$

5. Menentukan fungsi transfer $H(s)=1/B_n(s)$ dari nilai orde yang didapat, yang ditunjukan pada **Gambar 2.30** berikut.

n (order)	Normalized Denominator Polynomials in Factored Form
1	$(1+s)$
2	$(1+1.414s+s^2)$
3	$(1+s)(1+s+s^2)$
4	$(1+0.765s+s^2)(1+1.848s+s^2)$
5	$(1+s)(1+0.618s+s^2)(1+1.618s+s^2)$
6	$(1+0.518s+s^2)(1+1.414s+s^2)(1+1.932s+s^2)$
7	$(1+s)(1+0.445s+s^2)(1+1.247s+s^2)(1+1.802s+s^2)$
8	$(1+0.390s+s^2)(1+1.111s+s^2)(1+1.663s+s^2)(1+1.962s+s^2)$
9	$(1+s)(1+0.347s+s^2)(1+s+s^2)(1+1.532s+s^2)(1+1.879s+s^2)$
10	$(1+0.313s+s^2)(1+0.908s+s^2)(1+1.414s+s^2)(1+1.782s+s^2)(1+1.975s+s^2)$

Gambar 2.30. Tabel Denominator Polinomial *Butterworth*

6. Denormalisasi fungsi transfer H(s) dari *filter Butterworth* dengan transformasi

low pass ke *band pass* untuk mendapatkan fungsi H(z) dengan rumus sebagai berikut.

$$H(s) = \frac{1}{B_n(s)} \qquad (2.37)$$

$$s = \frac{a^2 + w_{p1} w_{p2}}{a(w_{p2} - w_{p1})} \qquad (2.38)$$

$$a = \frac{z-1}{z+1} \equiv \frac{1-z^{-1}}{1+z^{-1}} \qquad (2.39)$$

7. Mengubah kembali fungsi H(z) ke dalam persamaan beda fungsi diskrit y(n).