

BAB II LANDASAN TEORI

Pada bab ini akan dijelaskan tentang berbagai teori yang digunakan untuk melakukan penelitian ini. Teori yang berhubungan seperti penjelasan *moment* secara umum, *Zernike polynomials*, *Zernike moments*, modifikasi pada *Zernike moments*, *butterworth filtering* dan *euclidian distance* akan menjadi bahasan utama pada bab ini.

2.1 Moments

Moment function banyak digunakan untuk analisis citra seperti pada pengenalan pola, klasifikasi objek, rekonstruksi citra, dan estimasi posisi. Kumpulan *moment* yang diperoleh dari hasil komputasi citra dapat merepresentasikan karakteristik global dan memberikan informasi geometrik pada sebuah citra.

Sebuah citra dapat direpresentasikan dengan menggunakan fungsi distribusi $f(x,y)$ dimana nilai dari fungsi tersebut merupakan nilai pixel pada lokasi (x,y) . Asumsikan ζ merupakan luas wilayah tertentu pada $f(x,y)$, maka *moment functions* ϕ_{pq} dari *order* $(p+q)$ dari $f(x,y)$ adalah:

$$\phi_{pq} = \iint \Psi_{pq}(x,y)f(x,y)dx dy; \quad p,q = 0,1,2,3,\dots \quad (2.1)$$

Dimana $\Psi_{pq}(x,y)$ adalah fungsi kontinu pada (x,y) di wilayah ζ , yang disebut dengan *weighting kernel* atau *basis set*.

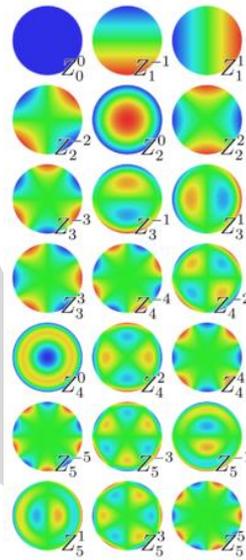
Persamaan fungsi *moment* pada persamaan (2.1) di atas dapat pula didefinisikan dengan menggunakan koordinat polar (r,θ) . Berikut fungsi *moment* yang dinyatakan menggunakan koordinat polar:

$$\phi_{pq} = \iint r^{p+q+1}\Psi_{pq}(\theta)f(r, \theta)dr d\theta; \quad p,q = 0,1,2,3,\dots \quad (2.2)$$

2.2 Zernike Polynomials

Zernike polynomials adalah sebuah deret *polynomial* yang *orthogonal* (tegak lurus) terhadap *unit disk* yang merupakan suatu kumpulan titik P yang memiliki jarak < 1 . *Zernike polynomials* ditemukan oleh Zernike pada tahun 1934, pada awalnya dibuat untuk membantu merepresentasikan hasil dari *optical test* yang

biasanya ditujukan untuk menggambarkan data *wavefront* dalam bentuk *polynomial* [MAT09].



Gambar 2. 1 Zernike Polynomial[WIK09]

Zernike polynomial dapat dibedakan menjadi *Zernike polynomial* genap dan ganjil yang dirumuskan seperti dibawah ini:

$$\begin{aligned} \text{o } U_n^m(\rho, \phi) &= R_n^m(\rho) \sin(m\phi) \\ \text{e } U_n^m(\rho, \phi) &= R_n^m(\rho) \cos(m\phi) \end{aligned} \quad (2.3)$$

Dimana *radial function* $R_n^m(\rho)$ dengan $n \geq m \geq 0$ yaitu:

$$R_n^m(\rho) = \begin{cases} \sum_{l=0}^{(n-m)/2} \frac{(-1)^l (n-l)!}{l! \left[\frac{1}{2}(n+m)-l\right]! \left[\frac{1}{2}(n-m)-l\right]!} \rho^{n-2l} & \text{for } n-m \text{ even} \\ 0 & \text{for } n-m \text{ odd.} \end{cases} \quad (2.4)$$

Zernike polynomials juga sering direpresentasikan dengan rumus yang berbeda pula yaitu dengan ϕ sebagai sudut azimuthal antara $0 \leq \phi \leq 2\pi$ dan ρ adalah *radial distance* antara $0 \leq \rho \leq 1$. *Polynomial* genap dan ganjil direpresentasikan dengan rumus di bawah ini:

$$Z_n^{-m}(\rho, \phi) = \text{o } U_n^m(\rho, \phi) = R_n^m(\rho) \sin(m\phi)$$

$$Z_n^m(\rho, \phi) = {}^e U_n^m(\rho, \phi) = R_n^m(\rho) \cos(m\phi). \quad (2.5)$$

Bentuk lain dari *Zernike polynomials* juga dapat direpresentasikan menggunakan *Gamma function* dan *hypergeometric function*.

$$R_n^m(\rho) = \frac{\Gamma(n+1) {}_2F_1\left(-\frac{1}{2(m+n)}, \frac{1}{2(m-n)}; -n; \rho^{-2}\right)}{\Gamma(1/2(2+n-m))\Gamma(1/2(2+n+m))} \rho^n \quad (2.6)$$

Dimana $n - m$ adalah ganjil dan $m \neq n$, $\Gamma(z)$ adalah *gamma function*, dan ${}_2F_1(a, b; c; z)$ adalah *hypergeometric function*.

2.3 Zernike Moments

Zernike moments pertama kali diperkenalkan oleh Teague [MUK98]. Bila dilihat dari sisi penghitungan, *Zernike moments* melibatkan penghitungan yang lebih kompleks dibandingkan jenis *moment* yang lain seperti *geometric* maupun *legendre moments*. Namun, *Zernike moments* telah dibuktikan sebagai salah satu metode ekstraksi ciri yang baik karena kemampuan dalam merepresentasikan sebuah citra yang mengalami distorsi dan rotasi [BIN02].

Zernike moments termasuk pada *region-based shape descriptor*. Jenis *moments* ini dikenal sangat efisien pada penggunaannya untuk pengenalan pola sebab memiliki sifat ortogonalitas pada *Zernike polynomials* dalam hasil ekstraksi ciri yang dibentuk serta memiliki properti yang tidak bergantung pada rotasi citra. Berikut persamaan yang digunakan untuk mencari *Zernike moments* dari suatu citra.

R = himpunan dari *real number*;

Z = himpunan dari *complex number*;

2D *Zernike moments* $A_{nm} \in Z$ dari suatu citra f dapat didefinisikan sebagai berikut [MUK98]:

$$A_{nm} = \frac{(n+1)}{\lambda} \iint_{x^2+y^2 \leq 1} R_{nm}(\rho) e^{-jm\theta} f(x, y) dx dy \quad (2.7)$$

Dimana λ adalah konstanta normalisasi yang dihitung dari jumlah nilai pixel yang berada dalam satuan lingkaran yang didefinisikan oleh π yang berada pada domain kontinu. $n = 0, 1, 2, \dots, \infty$ adalah order dari *moment* tersebut. M adalah pengulangan yang memiliki konstraint yaitu $n - |m|$ adalah genap dan $|m| \leq n, j = \sqrt{-1}, \rho = \sqrt{x^2 + y^2}, \theta = \arctan(y/x)$, dimana $(x,y) \in [-1,1]^2$. $R_{nm}(\rho)$ adalah *radial polynomials* dimana $R_{nm} \in [0,1]$.

$$R_{nm}(\rho) = \sum_{\substack{k=m \\ (n-k) \text{ is even}}}^n B_{nmk} \rho^k \quad (2.8)$$

Dimana koefisien $B_{nmk} = (-1)^k \frac{(n-k)!}{k! \left(\frac{n+|m|}{2} - k\right)! \left(\frac{n-|m|}{2} - k\right)!}$. B_{nmk} dapat diefisienkan pada komputasinya dengan menggunakan *recurrence relation* [MUK98].

$$B_{nmn} = 1$$

$$B_{n(m-2)n} = B_{nmn} \frac{n+m}{n-m+2}$$

$$B_{nm(k-2)} = -B_{nmk} \frac{(k+m)(k-m)}{(k+n)(n-k+2)} \quad (2.9)$$

Persamaan (2.7) untuk mencari *Zernike moments* ($Z_{n,m}$) di atas masih bisa disederhanakan dengan menerapkan *symmetry method* [MUK98]:

$$Z = \frac{(n+1)}{\pi} \sum \sum_{x^2+y^2 \leq 1} f(x,y) V_{n,m}^*(x,y) \quad (2.10)$$

$V_{n,m}^*$ merupakan konjugasi dari $V_{n,m}$, dimana [MUK98]:

$$V_{n,m}(x,y) = V_{n,m}(\rho, \theta) = R_{n,m}(\rho) e^{jm\theta} \quad (2.11)$$

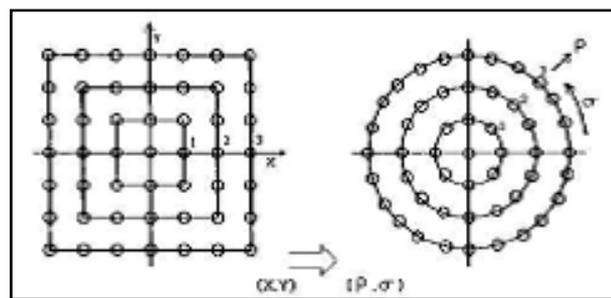
Persamaan (2.11) di atas terkait dengan formula Euler. Formula Euler merupakan konsep analisis matematika pada bilangan kompleks yang menyatakan keterhubungan antara fungsi trigonometri dan eksponensial bilangan kompleks. Formula Euler menyatakan untuk semua bilangan real [MOS02]:

$$e^{ix} = \cos(x) + i \sin(x) \quad (2.7)$$

Sesuai dengan persamaan di atas, maka $e^{-jm\theta}$ dapat dihitung dengan

$$e^{-jm\theta} = \cos(m\theta) + (-j)\sin(m\theta) \quad (2.8)$$

Dengan perumusan *Zernike moments* dengan menggunakan koordinat polar, maka *Zernike polynomials* yang harus dihitung pada masing-masing pixel akan menghasilkan komputasi yang besar dibandingkan pada penghitungan pada *Legendre Moment* dan *Geometric Moment*. Oleh karena itu, diperlukan sebuah algoritma untuk mengubah *square image* menjadi *circular image*, agar *Zernike polynomials* dapat dihitung hanya sekali untuk semua pixel yang dipetakan pada lingkaran yang sama.



Gambar 2. 2 *Square to Circle Transformation* [MUK98]

Sebuah pixel pada citra dapat dianggap sekumpulan *concentric square* dan dapat dipetakan menjadi *concentric circle* dengan melakukan *square to circular image transform* seperti Gambar 2.2 di atas. Jika sistem koordinat *square image* adalah (x,y) yang merupakan jarak titik tersebut dari titik pusat $(0,0)$, sedangkan pada *circle image* dapat direpresentasikan dengan menggunakan ρ, σ dimana ρ adalah jari-jari lingkaran dan σ adalah index posisi dari pixel pada suatu lingkaran.

Adapun algoritma untuk mengubah *square image* menjadi *circular image* sebagai berikut [MUK98]:

$$\rho = \max(|x|, |y|)$$

$$\text{if } |x| = \rho, \text{ then } : \sigma = 2(\rho - x) \frac{x}{|y|} + \frac{xy}{\rho},$$

$$\text{if } |y| = \rho, \text{ then } : \sigma = 2y - \frac{xy}{\rho} \quad (2.9)$$

Transformasi diatas bersifat *one to one* dan *invertible* sehingga $f(\rho, \sigma) = f(x, y)$. Jika N merupakan ukuran sebuah citra, maka rentang nilai pada *circle image* adalah:

$$-\frac{N}{2} \leq x, y \leq \frac{N}{2}; \quad 0 \leq \rho \leq \frac{N}{2}; \quad 1 \leq \sigma \leq 8\rho \quad (2.10)$$

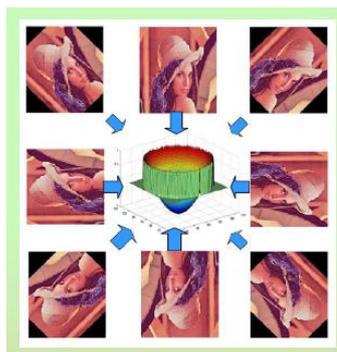
Adapun cara untuk mendapatkan koordinat polar (r, θ) pada pixel (ρ, σ) adalah sebagai berikut:

$$r = \frac{2\rho}{N}, \quad \theta = \frac{\pi\sigma}{4\rho} \quad (2.11)$$

Sebagai alat ekstraksi ciri, *Zernike moments* memiliki banyak keunggulan dibandingkan teknik moment yang lain seperti *Legendre Moment* dan *Geometric Moment*. Adapun beberapa properti yang dimiliki oleh *Zernike moments* sebagai berikut [YUL00][REV09]:

- *Rotation Invariance*

Zernike moments dapat menjadi deskriptor yang baik walaupun citra tersebut mengalami rotasi. Sifat ini disebabkan *Zernike moments* memiliki *rotation invariant*.



Gambar 2. 3 Citra Rotasi

- *Robustness*

Zernike moments bersifat *robust* pada *noise*.

- *Expression Efficiency*

Pada *Zernike moments*, tidak ada informasi yang *redundant* sebab setiap informasi tersebut saling *orthogonal*.

- *Effectiveness*

Sebuah citra dapat diekstraksi dengan lebih baik menggunakan *Zernike moments* dibandingkan dengan menggunakan jenis penghitungan *moment* yang lain.

- *Multi-level representation*

Zernike moments pada order yang kecil dapat mengekstraksi ciri global sebuah citra dan pada order yang tinggi dapat merepresentasikan citra tersebut secara lebih detail.

Ada banyak penelitian-penelitian yang bertujuan untuk meningkatkan kecepatan komputasi *Zernike moment* serta mengurangi kompleksitas pada penghitungan moment di order-order yang tinggi, diantaranya *Zernike moments* via *geometric moments method*, *Q-Recursive method*, *coefficient method*, dan *symmetry method*.

2.4 Modifikasi Pada Zernike Moments

Penelitian pada *Zernike moments* berkembang cukup cepat. Penelitian yang dilakukan dikonsentrasikan pada pengurangan waktu komputasi, kompleksitas penghitungan, serta peningkatan tingkat pengenalan. Pada pengembangan perangkat lunak ini, peneliti menggunakan modifikasi Zernike moment yang dikembangkan oleh Mukundan [MUK98]. Modifikasi yang dilakukan adalah dengan menggunakan penghitungan rekursif *Zernike Polynomials*, dan normalisasi *Zernike Moments*.

2.4.1 Penghitungan Rekursif Zernike Polynomials

Penelitian terhadap *Zernike moments* sangat menekankan pada kecepatan penghitungan pada order-order yang tinggi dan kompleksitas komputasi. Para peneliti berusaha untuk mengurangi waktu komputasi pada penghitungan *Zernike moments* suatu citra namun tetap memiliki ketelitian penghitungan yang tinggi. Salah satu penelitian yang telah diusulkan adalah teknik rekursif yang diterapkan

pada penghitungan *Zernike polynomial*. Teknik rekursif ini terbukti dapat mempercepat penghitungan *Zernike polynomial* [MUK98].

Untuk menghitung *Zernike polynomial* $R_{nm}(r)$, sebagai berikut [MUK98]:

$$R_{n,m}(r) = \frac{(k_2 r^2 + k_3)R_{n-2,m}(r) + k_4 R_{n-4,m}(r)}{k_1}$$

$$k_1 = (n-1)(n+1)(n-2)/2$$

$$k_2 = 2n(n-1)(n-2)$$

$$k_3 = -(n-1)^3$$

$$k_4 = -n(n-1)(n-3)/2 \quad (2.12)$$

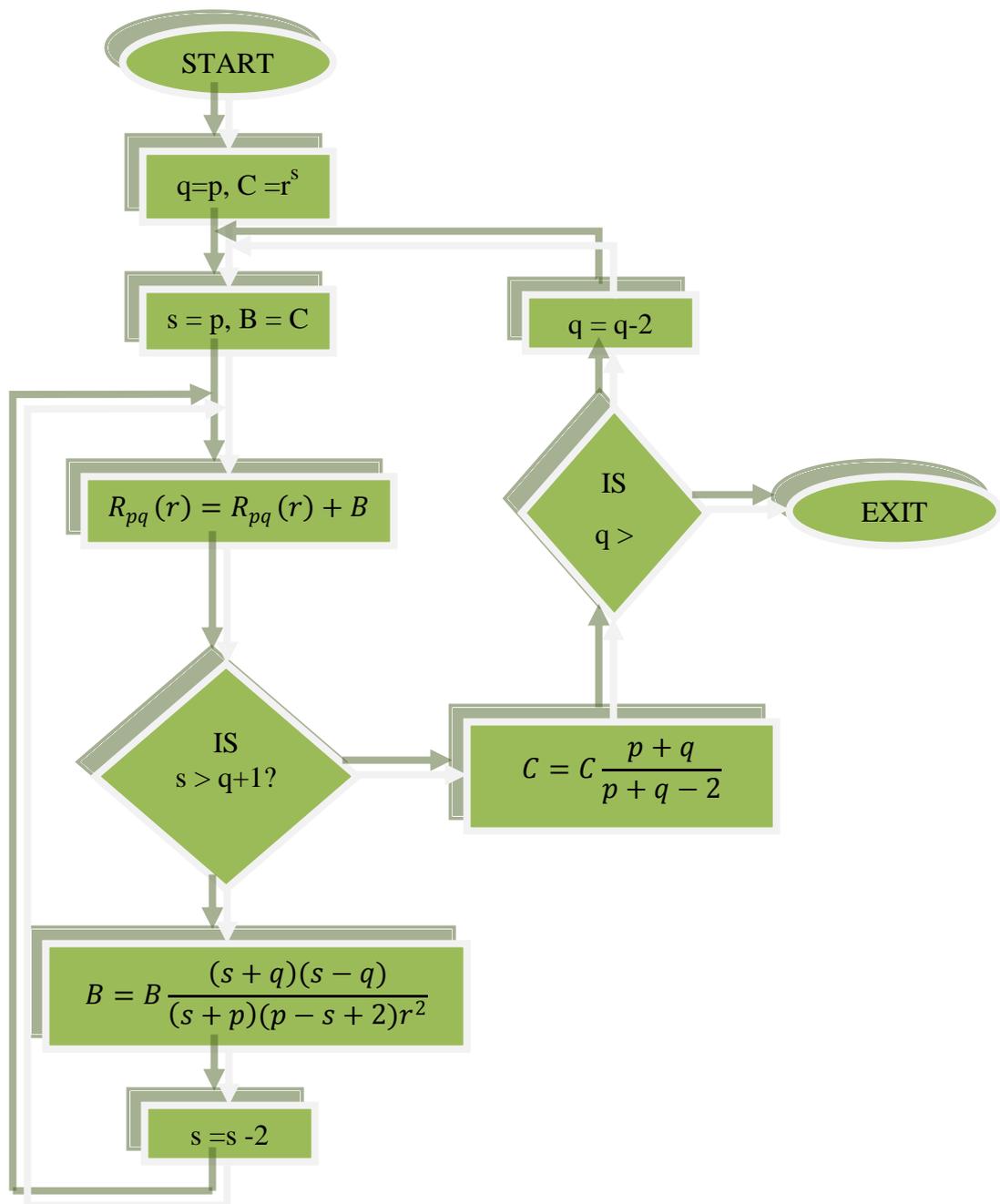
Teknik rekursif di atas merupakan skema rekursif yang hanya berada pada indeks n , namun bila ingin mendapatkan keseluruhan proses skema rekursif tersebut harus diulang sebanyak m kali. Ada skema rekursif yang berbeda untuk menghitung *Zernike polynomial*. Skema ini melakukan penghitungan pada koefisien B_{pqk} (*radial polynomial*), yaitu [MUK98]:

$$B_{ppp} = 1,$$

$$B_{p(q-2)p} = B_{pqp} \frac{(p+q)}{(p-q+2)},$$

$$B_{pq(s-2)} = B_{pqs} \frac{q^2-s^2}{(p+s)(p-s+2)}, \quad (2.13)$$

Berikut *flow chart* yang dapat menggambarkan penghitungan komputasi *Zernike polynomials*. ($p = \text{order of moment}$, $q = \text{repetition}$, $r = (n-2s)$, $s = 0 \dots (n-|m|)/2$)



Gambar 2. 4 Zernike Polynomial Flow Chart [MUK98]

2.4.2 Normalisasi Zernike Moments

Zernike moments pada order tinggi tidak selalu dapat memberikan tingkat pengenalan yang tinggi, sebab pada suatu citra terutama citra *dental x-ray* banyak terdapat *noise* yang justru dapat mengganggu tingkat pengenalan pada suatu citra

dental x-ray. Hal ini ditambah juga dengan *magnitude* pada *Zernike moments invariant* cukup memiliki perbedaan yang besar.

Untuk mengurangi besarnya perbedaan pada *Zernike moments* Z_{pq} , dilakukan normalisasi terhadap *Zernike moments* (\hat{Z}_{pq}) tersebut [BIN02].

$$\hat{Z}_{pq} = \frac{Z_{pq}}{m_{00}}; m_{00} = \sum_x \sum_y f(x, y) \quad (2.14)$$

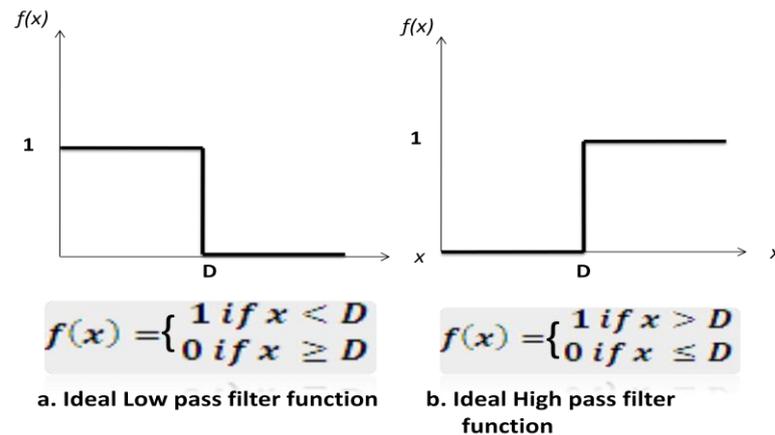
Normalisasi yang dilakukan adalah dengan membagi seluruh nilai *Zernike moments* pada suatu citra dengan total dari nilai masing-masing pixel atau *geometric moment* pada titik (0,0).

Dengan penggunaan normalisasi *Zernike moment* ini diharapkan perangkat lunak yang sedang dikembangkan ini dapat order yang tinggi namun tidak mengurangi tingkat pengenalan akibat noise yang ada pada citra *dental x-ray*, namun sebaliknya yaitu meningkatkan akurasi dan pengenalan pada *feature description*.

2.5 Butterworth Filtering

Butterworth filtering merupakan salah satu teknik *filtering* yang menggunakan frekuensi sebagai domain [GON02]. Untuk merubah *spatial domain* ke *frequency domain* digunakan *fourier transform*. Tujuan perubahan domain tersebut adalah untuk mempermudah penghitungan dan mengurangi kompleksitas penghitungan.

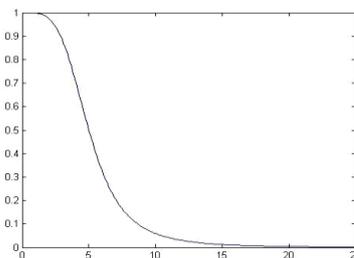
High-pass filtering adalah sebuah filter yang memunculkan frekuensi tinggi, namun menghilangkan frekuensi rendah dan sebaliknya *low-pass filtering* adalah memunculkan frekuensi rendah dan menghilangkan frekuensi tinggi [GON02]. Batas tinggi dan rendahnya sebuah frekuensi disebut dengan *cutoff* (D) seperti pada Gambar 2.5 di bawah ini.



Gambar 2. 5 Ideal Filtering

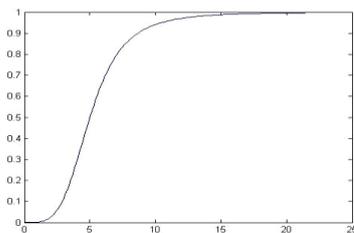
Butterworth filtering lebih baik dibandingkan *ideal filtering* sebab menggunakan fungsi *filtering* yang lebih halus sehingga pemotongan frekuensi atau filtering yang terjadi memperhatikan presisi bilangan desimal. Bentuk fungsi *filtering* pada *butterworth filtering* yang lebih halus biasa terlihat pada Gambar 2.6 dan 2.7 di bawah ini. Berikut fungsi *filtering* yang digunakan, yaitu:

- *Low-pass filtering*: $f(x) = \frac{1}{1 + (x/D)^{2n}}$



Gambar 2. 6 High-Pass Butterworth Filtering

- *High-pass filtering*: $f(x) = \frac{1}{1 + (D/x)^{2n}}$



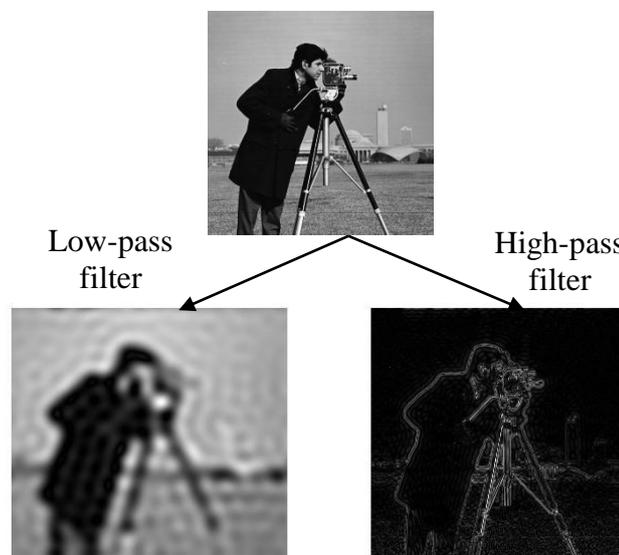
Gambar 2. 7 Low-Pass Butterworth Filtering

Adapun beberapa tahapan untuk melakukan proses *butterworth filtering* adalah sebagai berikut [GON02]:

- *Low-pass filtering*
 1. Buat sebuah *circular low-pass filter*.
 2. Baca citra yang akan dilakukan *filtering* dan hitung DFT citra tersebut.
 3. Kalikan *circular low-pass filter* dengan DFT yang telah dihasilkan.
 4. Cari *inverse DFT* dari hasil perkalian di atas.
- *High-pass filtering*

High-pass filtering memiliki tahapan yang sama seperti *low-pass filtering*, namun *filter* yang dibuat adalah *high-pass filtering*.

Proses *filtering* ini biasanya dilakukan untuk memperbaiki sebuah citra. Proses *low-pass filtering* akan memperhalus bentuk citra tersebut. Pada *high-pass filtering*, bentuk citra akan dipertajam *edge* untuk lebih memperlihatkan bentuk dari citra tersebut.



Gambar 2. 8 *High-Pass dan Low-Pass Filtering*

2.6 Euclidian Distance

Jarak Euclidian adalah jarak terpendek antara dua buah titik. Jika terdapat dua buah titik, maka jarak terpendek tersebut didapatkan dengan cara menarik garis lurus yang menghubungkan kedua titik tersebut. Dalam ruang Euclidian berdimensi n , R^n , jarak antara titik x dan y dapat dirumuskan sebagai berikut:

$$D = |x - y| = \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \quad (2.15)$$

Dimana n adalah jumlah titik dalam R^n .

Bila bilangan bekerja dalam ruang Euclidian berdimensi dua, maka jarak Euclidian antara titik $p(x_1, y_1)$ dan $q(x_2, y_2)$ dapat dihitung dengan menggunakan rumus berikut:

$$D(p, q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.16)$$

Proses identifikasi pada perangkat lunak ini menggunakan *euclidian distance* sebagai alat untuk mencari kemiripan hasil ekstraksi pada suatu citra. *Zernike moments* yang dihasilkan pada proses penghitungan berupa bilangan kompleks sehingga *data type* yang diperlukan pada pengimplementasian kode nantinya diperlukan sebuah *complex number data type*.

$$D = \sum_{(p, q) \in D} \sum (|Z_{pq}| - |Z'_{pq}|)^2 \quad (2.17)$$

Pemenang dari proses identifikasi ini adalah jarak yang paling minimum antara citra dental yang akan diidentifikasi dan dental yang ada pada basis data (*dental records*).