BAB 2
LANDASAN TEORI

2.1 Optical Character Recognition (OCR)

Optical Character Recognition merupakan suatu teknologi yang dapat mengubah atau mengkonversikan berbagai jenis dokumen seperti dokumen kertas yang di *scan,* file berextensi dokumen digital, maupun gambar yang telah diambil sebelumnya menggunakan kamera smartphone atau kamera digital lainnya menjadi data yang dapat diolah dan dicari. OCR pun berfungsi untuk mengekstraksi teks dari gambar secara mekanik dan elektronik menjadi teks yang dikodekan Dallam kode yang dikenali mesin[10]. Nantinya dapat menjadi aplikasi tambahan untuk *scanner*.

Sub bidang dari OCR ini dapat digunakan untuk digunakan untuk melakukan komputerisasi pada pekerjaan yang masih bersifat dikarenakan OCR merupakan salah satu sub bidang dari Computer Vision yang bertujuan untuk melakukan pengenalan terhadap karakter-karakter hasil cetak dari pemindaian dokumen[11].

Sebelum OCR melakukan ekstraksi teks pada gambar yang di scan, OCr telebih dahulu meratakan gambar, dimana apabila gambar masih miring atau terbalik maka akan disejajarkan atau diluruskan. Selanjutnya OCR akan melakukan analisis, yaitu membedakan antara gambar dan teks. Setelah itu secara otomatis akan mengatur arah gambar sesuai dengan arah teks. Berikutanya memisahkan setiap huruf dan angka, mengidentifikasi gambar, dan terakhir menghasilkan file yang berbentuk dokumen yang berisi teks setelah di ekstrak.

2.2 Library Tesseract

Yaitu merupakan salah satu dari library open source Optical Character Recognition (OCR) yang sangat terkenal. Diantara library yang dimiliki OCR, terserach merupakan salah satu OCR open source yang paling akurat pada saat ini[12]. Walaupun tesseract dapat membaca teks pada citra secara akurat, apabila pada gambar terdapat noise yang besar, akan membuat tesseract menjadi tidak

mampu untuk melakukan ekstraksi teks dari citra sehingga dibutuhkannya pengolahan terhadap citra untuk mengurangi noise pada citra tersebut. Agar dapat mengenali karakter, tesseract membutuhkan tiga proses yaitu *image processing*, *segmentation*, *word recognition*[12].

2.3 Image Processing

Merupakan Suatu bentuk dari pengolahan atau pemrosesan sinyal dengan input berupa citra (image) dan ditransformasikan menjadi citra lain sebagai keluarannya dengan teknik tertentu. Image processing pun dilakukan untuk memperbaiki kesalahan data sinyal gambar yang terjadi akibat transmisi dan selama akuisisi sinyal, serta untuk meningkatkan kualitas penampakan gambar agar lebih mudah diinterpretasi oleh sistem penglihatan manusia baik dengan melakukan manipulasi dan juga penganalisisan terhadap gambar. Pada saat melakukan image processing, terdapat 3 tahapan yang terdiri *grayscaling, smoothing, dan thresholding*[12].

2.3.1 Grayscaling

Pengertian lain dari grayscale adalah berbagai macam warna yang bernuansa monokratik atau cahaya yang hanya terdiri dari satu warna yang Panjang dan tidak dapat diuraikan kembali sehingga dari hitam menjadi putih. Dengan kata lain, grayscaling merupakan suatu proses megnubah citra yang tadinya bewarna atau RGB menjadi citra yang hanya memiliki derjaat keabuan[12]. Berikut adalah rumus yang digunakan untuk mengubah citra RGB menjadi citra grayscale.

$$Gray = 0.299 * red + 0.587 * green + 0.114 * blue \quad (2.1)$$

2.3.2 Thresholding

Yaitu merupakan suatu proses untuk memisahkan background dair objek yang diamati dengan mengubah gambar menjadi warna hitam putih[12]. Pada tahap ini, umumnya *thresholding* pada citra *grayscale* bertujuan unutk

menghasilkan nilai citra biner[13]. Berdasarkan penelitian terhadap segmentasi citra ikan menggunakan thresholding, secara matematis rumus terhadap citra biner dari citra grayscale seperti rumuh di bawah ini.

$$g(x,y) = \begin{cases} 1, if\ f(x,y) \geq T \\ 0, if\ f(x,y) < T \end{cases} \quad (2.2)$$

Persamaan 1 diatas menyatakan g(x,y) adalah citra biner dari citra grayslcae f(x,y), serta T adalah nilai yang menyatakan threshold[13].. untuk dapat menntukan nilai T maka dibutuhkannya metode thresholding global dan thresholding local.

2.3.3 Maximally Stable Extremal Regions (MSER)

Maximally Stable Extremal Regions (MSER) adalah metode yang digunakan untuk mendeteksi gumpalan pada sebuah citra. MSER didasarkan pada sebuah ide dalam mengambil daerah yang hampir sama berdasarkan dari nilai threshold[14]. Semua piksel yang berada di bawah nilai threshold yang diberikan adalah berwarna putih, selain itu berwarna hitam. Pada urutan citra threshold, dapat dilihat terlebih dulu gambar hitam, kemudian bintik-bintik putih yang sesuai dengan intensitas lokal minima akan muncul, kemudian berkembang lebih besar. Bintik-bintik putih ini pada akhirnya akan bergabung, sampai seluruh gambar putih. Himpunan semua komponen terhubung secara berurutan merupakan bentuk himpunan dari semua daerah extremal[14]. Kata dari extremal tersebut mengacu pada sifat dimana semua pixel dalam MSER memiliki intensitas baik yang lebih tinggi atau lebih rendah dari semua pixel pada batas luarnya. Dari masing-masing pixel nantinya diurutkan berdasarkan level pada nilai threshold menggunakan connected component tree.

Connected component tree adalah suatu struktur yang menghubungkan beberapa simpul (node) menjadi bentuk akar pohon yang saling berhubungan. Sebelumnya, connected component tree telah digunakan untuk pengimplementasian metode watersed pada segmentasi grayscale oleh Couprise et al[15]. Component tree pun dapat digunakan untuk pendeteksian daerah MSER pada gambar, karena setiap node pada connected component tree mewakilkan

daerah yang terhubung di dalam gambar masukan dan membentuk rangkaian pixel yang teratur disetiap levelnya[16].

Pada metode MSER, terdapat beberapa variable yang nantinya akan digunakan untuk proses pencarian daerah yang stable pada MSER, diantaranya :

1. MinArea

   Parameter ini digunakan untuk menentukan batas area terkecil yang dapat diterima pada Connected Component.

2. MaxArea

   Parameter ini digunakan untuk menentukan batas area terbesar yang dapat diterima pada Connected Component.

3. Delta

   Parameter ini bertujuan untuk menentukan berapa banyak tingkat keabuan (grayscale level) yang diperlukan agar area dapat dianggap stabil ataupun dianggap stabil secara maksimal.

4. MaxVariation

   Parameter ini digunakan sebagai penentu dari batas maksimal variasi yang dapat dterima dan dikatakan stabil.

Setiap hirarki daerah yang terdiri dari beberapa area ekstremal, daerah tersebut memiliki masing-masing nilai variasinya sendiri. Berikut adalah persamaan yang digunakan untuk menghitung nilai variasi pada setiap daerah.

$$Variation_i = \frac{Area^{i-\Delta} - Area^{i+\Delta}}{Area_i} \qquad (2.3)$$

Hasil dari perhitungan nilai variasi pada persamaan 3.4 diatas nantinya akan digunakan untuk menentukan daerah mana yang bernilai stable dan dijadikan daerah kandidat MSER.

2.4 Citra

Pengertian dari citra adalah bentuk dari fungsi intesitas 2 dimensi f(x,y), dimana x dan y merupakan koordinat special dan f pada titik (x,y) adalah tingkat dari kecerahan citra pada suatu titik[17]. Citra juga berperan penting sebagai salah

satu komponen multimedia dalam penyebaran informasi visual. Citra pun di bagi menjadi 2 jenis yaitu citra analog dan citra digital.

2.4.1 Citra Analog

Pengertian dari Citra Analog adalah citra yang memiliki sifat kontinu, seperti gambar pada monitor televisi, foto pada sinar X, foto yang telah tercetak dikertas foto, pemandangan alam, lukisan, hasil dari CT , dan lain sebagainya[18].

Nilai intensitas cahaya pada citra analog memiliki jarak antara 0 s.d ~. Alat akuisisi citra analog antara lain mata manusia dan kamera analog.

2.4.2 Citra Digital

Citra yang dapat diolah oleh komputer disebut dengan citra digital. Sehingga, pengertian dari citra digital adalah suatu larik bersifat dua dimensi atau suatu matriks yang dimana elemen-elemennya mentakan tingkat keabuan dari suatu elemen pada gambar[19]. Nilai intensitas cahaya pada citra digital bergantung pada kedalaman bit yang menyusunnya (materi lebih lanjut mengenai kedalaman bit suatu citra dapat dilihat pada laman berikut: Kedalaman Bit Suatu Citra Grayscale). Alat akuisisi citra digital antara lain yaitu kamera digital, smartphone, webcam, scanner, mikroskop digital, pesawat radiodiagnostik seperti CT Scan, CR, MRI, USG, dll.

2.5 Pengolahan Citra

Pengolahan pada citra merupakan proses dimana gambar asli yang meniiliki warna (Red, Green, Blue) diubah menjadi gambar yang dinginkan[20]. Lalu terdapat pula pengolahan citra digital yang merupakan teknik pemrosesan gambar 2 dimensi dengan menggunakan komputer yang berguna untuk memperbaiki kualitas pada suatu citra sehingga nantinya dapat lebih mudah dilihat oleh mata manusia[21].

2.6 Segmentasi

Segmentasi adalah suatu proses pemisahan objek, dimana satu objek dipisahkan dengan objek lainnya didalam suatu citra[22]. Degan kata lain, segmentasi

merupakan suatu bagian yang sangat penting dalam analisis citra secara otomatis, sebab pada prosedur ini obyek yang diinginkan akan disadap untuk proses selanjutnya, misalnya pada pengenalan pola. Proses segmentasi yang akan digunakan pada penelitian ini adalah segmentasi baris dan segmentasi karakter.

2.6.1 Segmentasi Baris

Segmentasi baris merupakan suatu proses segmentasi yang bertujuan untuik menghitung berapa jumlah baris karakter yang terdapat pada citra dan untuk mendapatkan area baris-baris pada karakter[23]. Segmentasi baris sangat diperlukan karena proses pada segmentasi karakter dilakukan secara perbaris, agar karakter yang di deteksi dapat berurut sesuai dengan barisnya.

2.6.2 Segmentasi Karakter

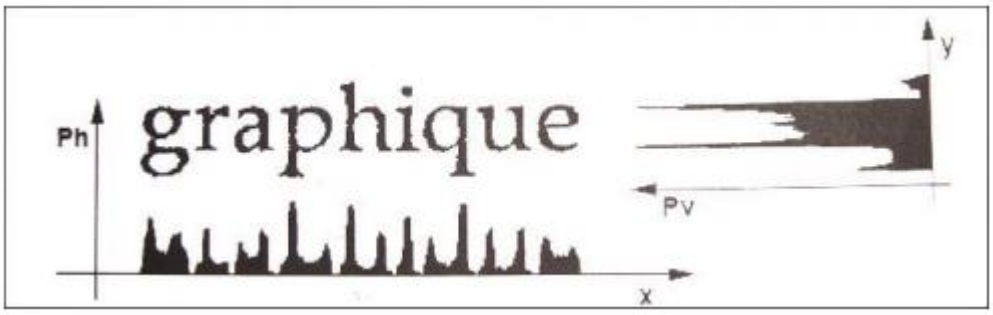
Segmentasi karakter merupakan suatu proses segmentasi yang bertujuan untuk memisahkan dari setiap karakter yang terdapat pada sebuah baris[23]. Keluaran dari proses segmentasi karakter ini berupa kumpulan dari berbagai citra karakter yang telah terpotong menjadi bentuk tunggal atau satu citra per-karakterr. Segmentasi karakter diperlukan agar nantinya ketika pendeteksian karakter bisa menjadi lebih tepat. Karena biasanya algoritma yang digunakan untuk ekstraksi ciri dan pengenalan karakter hanya bisa mengolah per-karakter saja[24].

2.7 Profile Projection

Profile projection adalah salah satu teknik yang digunakan untuk segmentasi karakter yang terdapat pada citra baik itu hasil dari OCR ataupun citra yang dibuat oleh sediri menggunakan komputer[24]. Profile projection ini bertujuan untuk memisahkan karakter untuk tiap baris dan juga kolom dengan secara otomatis dan akurat. Kelebihan dari profile projetion ini berdasarkan dari kemampuannya untuk dapat mendeteksi ruang antar baris dan kolom pada karakter sehingga dapat memisahkan karakter tersebut, dan juga apabila ukuran

dari masing-masing karkater baik pada baris atau kolom berbeda, maka teknik ini masih bisa dapat diandalkan.

Pada profile projection terdapat 2 jenis profil yaitu profil vertikal dan horizontal. Profil vertical merupakan banyakanya pixel hitam yang tegak lurus dengan sumbu *y*, sedangkan profil horisontal merupakan banyaknya pixel hitam yang tegak lurus dengan sumbu *x*[23]. Berikut adalah contoh dari profil vertical dan profil horizontal yang ditunjukkan pada gambar 2.1.



Gambar 2. 1 Contoh Profil Projection Vertical dan Horizontal[25]

Berdasarkan gambar 2.1 diatas, Profil vertical direpresentasikan dengan vector *Pv* berukuran *N,* sedangkan profil horisontal direpresentasikan dengan vector *Ph* berukuran *M.* Profil vertikal didefinisikan sebagai berikut :

$$P_v[i] = \sum_{i=1}^{M} S[i,j] \qquad (2.\ 4)$$

Sedangkan profil horisontal didefinisikan sebagai berikut

$$P_h[J] = \sum_{j=1}^{N} S[i,j] \qquad (2.\ 5)$$

2.8 Resize

Resize adalah suatu proses yang digunakan untuk memperbesar ataupun memerkecil ukuran dari sebuah citra sesuai dengan ukuran yang diinginkan. Proses resize pun bertujuan untuk menyamakan ukuran citra hasil dari segmentasi karakter agar dapat diproses pada saat melakukan ekstraksi ciri regionprops. Berikut adalah persamaan yang digunakan untuk meresize ukuran citra[26]

$$x = \frac{pb * pp}{pa} \quad (2.\ 6)$$

Keterangan:

x = nilai piksel baris baru

pb = Ukuran panjang matriks baru

pp = Posisi piksel baris

pa = Ukuran panjang matriks lama

$$y = \frac{lb * lp}{la} \quad (2.\ 7)$$

Keterangan:

y = Nilai piksel kolom baru

lb = Ukuran lebar matriks baru

lp = Posisi piksel kolom

la = Ukuran lebar matriks lama

2.9 Ekstraksi Fitur

Ektraksi Fitur (Feature Extraction) adalah suatu proses pengambilan ciri (feature) dari suatu bentuk yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Ektraksi fitur (Feature Extraction) bertujuan untuk mencari daerah fitur yang signifikan pada gambar atau citra tergantung pada karakteristik intriksi dan aplikasinya. Niali fitur yang akan digunakan nantinya adalah nilai area, metric, dan eccentricit. Pada penelitian ini, regionprops akan digunakan sebagai ekstraksi fitur pada citra karakter.

2.9.1 Regionprops

Region properties (regionprops) merupakan suatu fungsi yang dimiliki oleh MATLAB untuk mengukur sekumpulan properti-properti dari setiap region yang telah dilabeli dalam matriks label L. Bilangan integer positif yang merupakan elemen dari L berkorespondensi dengan region yang bersesuaian. Area, panjang

major axis, dan panjang minor axis yang digunakan dalam tugas akhir ini merupakan sebagian dari properti yang dihasilkan fungsi regionprops. Dalam fungsi regionprops sebuah obyek direpresentasikan sebagai sebuah regiondengan pendekatan bentuk elips.

Pada bagian regionprops ini, terdapat dua fitur yang akan digunakan untuk membedakan berbagai jenis karakter nantinya, yaitu adalah fitur metric dan eccentricity. fitur metric merupakan rasio antara luas dan keliling dari objek, jika hasil dari metric mendekati angka nol maka bentuk dari objek tersebut dekat dengan bentuk garis panjang, sedangkan metric yang mendekati angka satu maka objek tersebut dekat dengan bentuk yang bulat . sedangkan fitur eccentric merupakan nilai perbandingan antara jarak fokus elips terkecil dengan jarak fokus elips terbesar pada objek. Berikut adalah contoh syntak yang digunakan untuk ekstraksi fitur menggunakan regionprops.

```
contoh = imread('contoh_citra.png');
fitur_eccecntric=regionprops(contoh,'MajorAxisLength','
MinorAxisLength');
fitur_metric = regionprops(contoh, 'Area','Perimeter');
hasil_fitur_eccentric=sqrt(1-
((fitur_eccentric.MajorAxisLength^2)/(fitur_eccentric.M
inorAxisLength^2)));
hasil_fitur_metric=(4*phi*fitur_metric.Area)/(fitur_met
ric.Perimeter^2)a;
```

Berdasarkan syntak diatas, hasil fitur eccentric dan fitur metric dapat dihitung menggunakan persamaan dibawah ini.

$$e = \sqrt{1 - \frac{b^2}{a^2}} \qquad (2.\ 8)$$

Keterangan:

e = nilai fitur eccentricity

b = nilai MinorAxisLength

a = nilai MajorAxisLength

$$M = \frac{4\pi A}{P^2} \quad (2.9)$$

Keterangan:

M = nilai fitur metric

A = nilai Area

P = nilai perimeter

2.10 Klasifikasi

Klasifikasi adalah suatu pengelompokan data dimana data yang digunakan tersebut mempunyai kelas label atau target. Sehingga algoritma-algoritma untuk menyelesaikan masalah klasifikasi dikategorikan ke dalam Supervised Learning atau pembelajaran terawasi. Setelah citra karakter melewati proses ekstraksi fitur, maka nilai fitur tersebut akan diklasifikasikan menggunakan metode Learning Vector Quantization 3 (LVQ3).

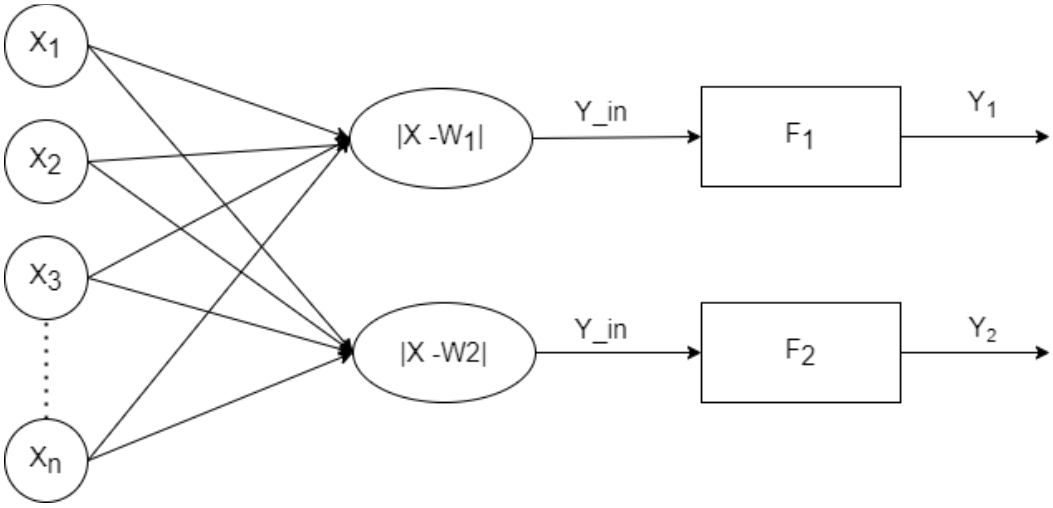
2.10.1 Learning Vector Quantization

Learning Vector Quantization (LVQ) merupakan suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input[27]. Jika dua vektor input mendekati sama, maka lapisan kompetitif akan meletakan kedua vektor input tersebut kedalam kelas yang sama. Vector bobot untuk suatu unit keluaran sering dinyatakan sebagai sebuah vector referensi. Setelah pelatihan, jaringan LVQ mengklasifikasi vector masukan dengan menugaskan ke kelas yang sama sebagai unit keluaran, sedangkan yang mempunyai vector referensi diklasifikasikan sebagai vector masukan.

2.10.1.1 Arsitektur Jaringan Learning Vector Quantization (LVQ)

Menurut Meri[28], arsitektur pada LVQ sama halnya dengan Self Organizing Map (SOM). LVQ juga terdiri dari 2 lapisan yakni input (X) dan output (Y), di mana antar lapisannya dihubungkan oleh bobot tertentu yang sering disebut sebagai vektor pewakil (W). Informasi yang diberikan ke jaringan pada saat pembelajaran bukan hanya vektor data saja melainkan informasi kelas dari data juga ikut dimasukkan. Tujuan dari dimasukkannya informasi ini adalah untuk menambah keakuratan jaringan dalam mengklasifikasikan data.

Berikut adalah bentuk dari arsitektur jaringan LVQ yang ditunjukkan pada Gambar 2.2 di bawah ini:



Gambar 2. 2 Arsitektur Learning Vector Quantization (LVQ)

Keterangan:

X = Vektor Masukan ($X_1 \dots, X_2 \dots, X_{n\dots}$)

$|X - W|$ = Selisih nilai dari jarak Euclidean antara vektor input dengan vektor bobot

Y_in = Masukan kelapisan kompetitif

F = Lapisan Kompetitif

Y = keluaran (output)

Ketika hasil pemrosesan jaringan memberikan hasil klasifikasi yang sama dengan informasi kelas yang diberikan di awal maka vektor pewakil akan disesuaikan agar lebih dekat dengan vektor masukan. Sebaliknya ketika hasil klasifikasi tidak sama dengan informasi kelas yang diberikan di awal, maka vektor pewakil akan disesuaikan agar menjauhi vektor masukan.

2.10.1.2 Algoritma Learning Vector Quantization (LVQ)

Pada algoritma LVQ, terdapat beberapa parameter yang dibutuhkan diantarahnya adalah[29]:

1. X, vektor-vektor pelatihan (X1,…Xi,…Xn).
2. T, kategori atau kelas yg benar untuk vektor-vektor pelatihan.
3. Wj, vektor bobot pada unit keluaran ke-j (W1j,…Wij,…,Wnj).
4. Cj, kategori atau kelas yang merepresentasikan oleh unit keluaran ke-j.
5. learning rate (α), α didefinisikan sebagai tingkat pembelajaran. Jika α terlalu besar, maka algoritma akan menjadi tidak stabil sebaliknya jika α terlalu kecil, maka prosesnya akan terlalu lama. Nilai α adalah $0 < \alpha < 1$.
6. Nilai pengurangan learning rate, yaitu penurunan tingkat pembelajaran. Pengurangan nilai α yang digunakan pada penelitian ini adalah sebesar 0.1*α.
7. Nilai minimal learning rate (Mina), yaitu minimal nilai tingkat pembelajaran yang masih diperbolehkan.
8. Pembaharuan bobot dilakukan dengan kondisi:
   a. Jika T= Cj maka: $Wj(t+1) = Wj(t) + \alpha(t)[x(t) - Wj(t)]$ (2. 10)
   b. Jika $T \neq Cj$ maka: $Wj(t+1) = Wj(t) - \alpha(t)[x(t) - Wj(t)]$ *(2. 11)*

Terdapat beberapa tahapan pelatihan pada algoritma LVQ diantaranya adalah[referensi]:

1. Tetapkan bobot awal variable input ke-j menuju ke kelas ke-i, parameter learning rate (α), nilai pengurangan learning rate, nilai minimal learning rate ($Min\ \alpha$), dan epoch = 0

2. Masukan data input dan target (T)
3. Lakukan apabila $\alpha > Min$ α maka:
    a. Hitung jarak euclidean antara vektor W dan vektor X: $\sqrt{(X - W)}\ 2$
    b. Tentukan jarak terdekat (Dmin)
    c. Lakukan perubahan Wj dengan ketentuan pada persamaan (2.8) dan (2.9)
    d. Kurangi nilai α.

Setelah melakukan pelatihan, maka akan diperoleh bobot-bobot akhir (W). Bobot - bobot ini nantinya akan digunakan untuk melakukan simulasi atau pengujian.

2.10.1.3 Algoritma Learning Vaector Quantizaton 2 (LVQ2)

Algortima LVQ dalam pengembanganya memiliki beberapa variasi, salah satunya adalah LVQ2. Pada algoritma LVQ1 vektor referensi yang paling dekat dengan vektor masukan saja yang diperbaharui. Sedangakan untuk variasi LVQ2, dua vektor (pemenang dan runner-up) diperbaharui jika beberapa kondisi dipenuhi. Modifikasi pertama adalah LVQ2, kondisi dimana kedua vektor akan diperbaharui jika [30]:

1. Unit pemenang dan runner up (vektor terdekat kedua) merepresentasikan kelas yang berbeda.
2. Vektor masukan mempunyai kelas yang sama dengan runner up.
3. Jarak antara vektor masukan ke pemenang dan jarak antara vektor masukan ke runner up kirakira sama.

Kondisi ini diperlihatkan di dalam notasi berikut:

1. X vektor masukan saat ini
2. Yc vektor referensi terdekat dengan X
3. Yr vektor referensi terdekat berikutnya dengan X (runner up)
4. Dc jarak dari X ke Yc
5. Dr jarak dari X ke Yr

Vektor referensi dapat diperbaharui jika masuk ke dalam daerah yang disebut window (ε). Window yang digunakan untuk memperbaharui vektor referensi didefinisikan sebagai berikut:

Vektor masukan X akan masuk ke dalam window bila

$$\frac{d_c}{d_r} > 1 - \varepsilon, \frac{d_c}{d_r} < 1 + \varepsilon \quad (2.12)$$

Dengan nilai ε tergantung dari jumlah data pelatihan. Berdasarkan Kohonen (1990an) dalam Fausett (1994) nilai ε = 0.3 adalah nilai yang disarankan. Vektor Yc dan Yr akan diperbaharui bila kondisi 1,2 dan 3 terpenuhi. Vektor Yc dan Yr diperbaharui dengan menggunakan persamaan:

$$Y_c(t + 1) = Y_c(t) - \alpha(t)[X(t) - Y_c(t)] \quad (2.13)$$

$$Y_r(t + 1) = Y_r(t) + \alpha(t)[X(t) - Y_r(t)] \quad (2.14)$$

Berikut ini merupakan tahapan pelatihan pada algoritma LVQ2 [30]:

1. Lakukan inisialisasi bobot w dan j
2. Input α (learning rate) atau derajat pembelajaran dan ε (window)
3. Untuk setiap pelatihan vektor pelatihan W temukan j sehingga $|Xi - Wj|$ bernilai minimum
4. Perbaiki dengan ketentuan :
   a. Jika $T = C_j$ maka $W_j = W_j + \alpha(X_i - W_j)$
   b. Jika $T = C_j$ maka $D_1 > (1 - \varepsilon) * D_2 \; AND \; D_2 < (1 + \varepsilon) * D_1$
      Jika *True* maka W yang tidak termasuk vektor X diperbahurui dengan menggunakan persamaan (2.11), Sedangkan W yang termasuk vektor X diperbaharui dengan menggunakan persamaan (2.12)
   c. Maka diperoleh W_j baru, jika *False* maka $W_j = W_j - \alpha(X - W_j)$
   d. Lakukan pengurangan α.

2.10.1.4 Algoritma Learning Vector Quantization 2.1 (LVQ2.1)

Modifikasi LVQ2.1 mempertimbangkan dua vektor referensi tederkat, yaitu $Yc1 \; dan \; Yc2$. Kondisi dalam memperbaharui kedua vektor tersebut adalah apabila salah satu dari vektor tersebut (misal, $Yc1$) masuk ke dalam kelas yang sama dengan vektor masukan x, sementara vektor lainnya (misal, $Yc2$) tidak masuk ke dalam kelas yang sama dengan vektor masukan x, maka dalam LVQ2 vektor x harus

masuk ke dalam window agar bisa terjadi pembaharuan. Pendefinisian window sebagai berikut[30]:

$$min\left[\frac{dc1}{dc2}, \frac{dc2}{dc1}\right] > 1 - \varepsilon \qquad (2.\ 15)$$
$$max\left[\frac{dc1}{dc2}, \frac{dc2}{dc1}\right] < 1 + \varepsilon$$

Jika kondisi-kondisi tersebut terpenuhi, maka vektor referensi yang masuk ke dalam kelas yang sama dengan vektor x akan diperbaharui menggunakan persamaan:

$$Y_c1(t + 1) = Y_c1(t) - \alpha(t)[X(t) - Y_c1(t)] \qquad (2.\ 16)$$

Sedangkan untuk vektor referensi yang tidak masuk ke dalam kelas yang sama dengan vektor x akan diperbaharui menggunakan persamaan:

$$Y_c2(t + 1) = Y_c2(t) - \alpha(t)[X(t) - Y_c2(t)] \qquad (2.\ 17)$$

2.10.1.5 Algoritma Learning Vector Quantization 3 (LVQ3)

LVQ 3 merupakan sebuah algoritma hasil pengembangan dari LVQ sebelumnya yaitu LVQ1, LVQ2, dan LVQ2.1. Pada LVQ3 vector referensi dapat diperbaharui jika masuk kedalam daerah yang disebut window (ε). Window yang digunakan untuk memperbaharui vektor referensi dapat didefiniskan sebagai berikut[30]:

$$min\left(\frac{dc}{dr}.\frac{dr}{dc}\right) > (1 - \varepsilon)(1 + \varepsilon) \qquad (2.\ 18)$$

Vector Yc akan diperbaharui bila kondisi persamaan window terpenuhi atau bernilai *True* dan salah satu dari 2 vector referensi terdekat berada dalam kelas yang sama dengan vector masukan. Maka vektor referensi yang masuk ke dalam kelas yang sama dengan vektor masukan akan diperbaharui menggunakan persamaan (2.14) dan vektor referensi yang tidak masuk ke dalam kelas yang sama dengan vektor x akan diperbaharui menggunakan persamaan (2.15). Apabila persamaan window tidak terpenuhi atau bernilai *False* maka vector Yc dan Yr diperbaharui dengan menggunakan persamaan sebagai berikut:

$$Y(t + 1) = Y(t) + \varepsilon\, a(t)[X(t) - Y(t)] \qquad (2.\ 19)$$

Berikut adalah Parameter yang dibutuhkan untuk proses pembelajaran LVQ3 diantaranya adalah[31]:

1. X, vektor-vektor pelatihan (X1,…Xi,…Xn)..
2. T, kategori atau kelas yg benar untuk vektor-vektor pelatihan.
3. Wj, vektor bobot pada unit keluaran ke-j (W1j,…Wij,…,Wnj).
4. Cj, kategori atau kelas yang merepresentasikan oleh unit keluaran ke-j.
5. learning rate (α), α didefinisikan sebagai tingkat pembelajaran. Jika α terlalu besar, maka algoritma akan menjadi tidak stabil sebaliknya jika α terlalu kecil, maka prosesnya akan terlalu lama. Nilai α adalah $0 < \alpha < 1$.
6. Nilai pengurangan learning rate, yaitu penurunan tingkat pembelajaran.
7. Nilai minimal learning rate (Mina), yaitu minimal nilai tingkat pembelajaran yang masih diperbolehkan. Pengurangan nilai α yang digunakan pada penelitian ini adalah sebesar $0.1 * \alpha$.
8. Nilai window (ε), yaitu nilai yang digunakan sebagai daerah yang harus dipenuhi untuk memperbaharui vektor referensi pemenang (Yc1) dan runner-up (Yc2) jika berada dikelas yang berbeda. Dengan menggunakan persamaan (2.16).
9. Jika memenuhi kondisi window (ε) , maka vektor referensi yang masuk ke dalam kelas yang sama dengan vektor x akan diperbaharui menggunakan persamaan 2.14.
10. Sedangkan vektor refrensi yang tidak masuk ke dalam kelas yang sama dengan vektor x akan diperbaharui menggunakan persamaan 2.15.

2.11 Sertifikat

Tanda atau surat keterangan (pernyataan) tertulis maupun tercetak yang dibuat oleh orang yang berwenang dan dapat digunakan sebagai bukti pemilikan dari suatu kejadian disebut dengan Sertifikat[32]. Sertifikat pun dapat digunakan sebagai bukti dari kepemilikan tanah, kelahiran, dll.

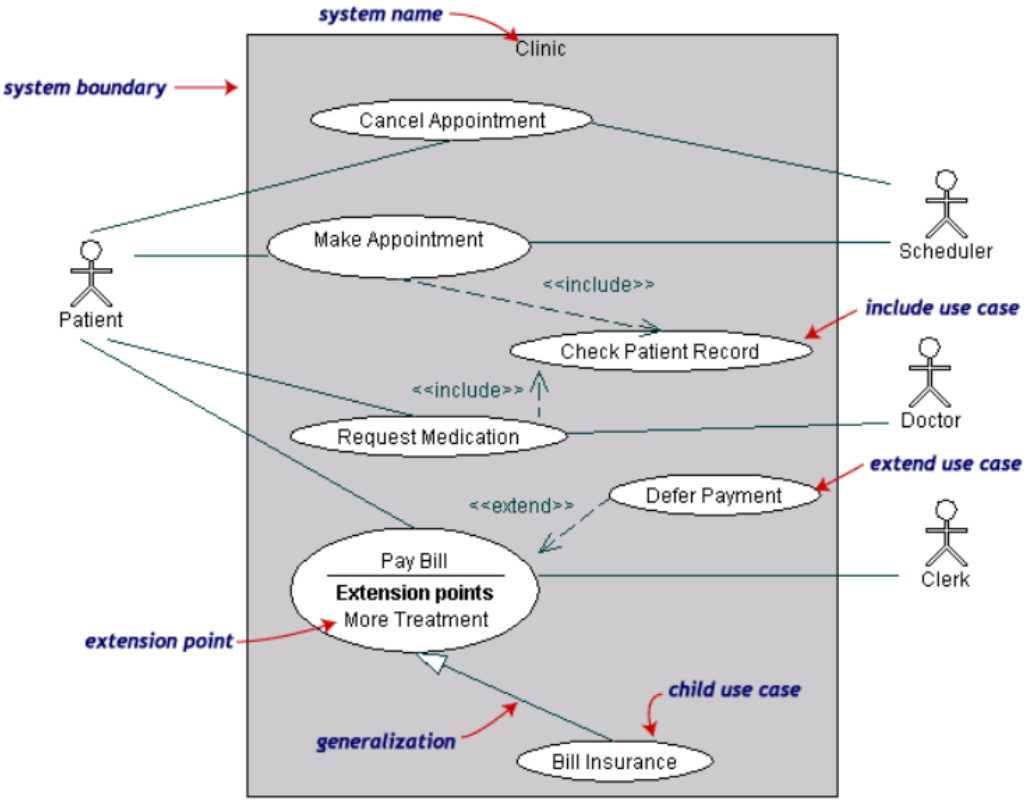
2.12 Unified Modelling Languange (UML)

Unified Modelling Languange merupakan sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi , merancang, dan mendokumentasikan sistem piranti lunak, dengan menawarkan sebuah standar untuk merancang model sebuah sistem[35]. UML versi 2.0 mencakup 13 macam diagram dan perangkat yang berfungsi untuk menggambarkan sistem informasi berorientasi objek dengan sangat lengkap dan rinci.

Meski demikan, tidak selalu 13 diagram tersebut digunakan saat para pengembang sedang mengembangkan perangkat lunak yang berorientasi objek Beberapa diagram UML yang digunakan dalam penelitian ini, antara lain ialah *Use Case Diagram, Activity Diagram, Sequance Diagram, dan Class Diagram*.

2.12.1 Use Case Diagram

Use Case diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, use case diagram juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya. *Use case diagram* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari use case *symbol* namun dapat juga dilakukan dalam *activity diagrams*. Use case digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem. Berikut adalah contoh *Use Case Diagram* yang ditunjukan pada Gambar 2.4.

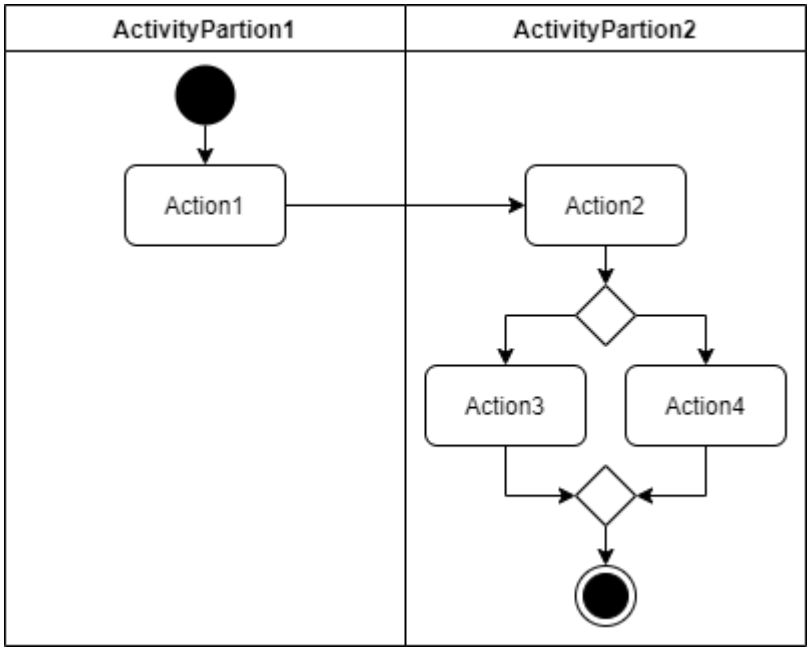


Gambar 2. 3 ***Use Case Diagram***

2.12.2 Activity Diagram

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum[35]. Suatu aktivitas dapat direalisasikan oleh satu *use case a*tau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Berikut adalah contoh activity diagram yang ditunjukan pada Gambar 2.5 dibawah ini.

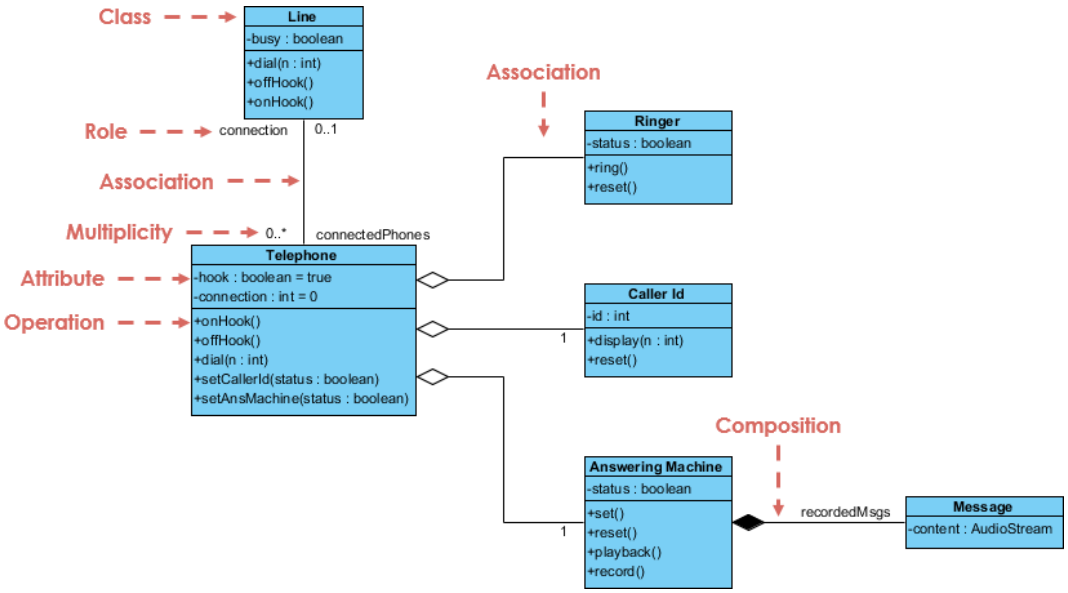


Gambar 2. 4 Activity Diagram

2.12.3 Class Diagram

*Clas*s adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Sedangkan *class diagram* merupakan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain[35]. Jadi *class diagram* ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.

Berikut adalah contoh dari *class diagram* yang dapat dilihat pada Gambar 2.6 dibawah ini.



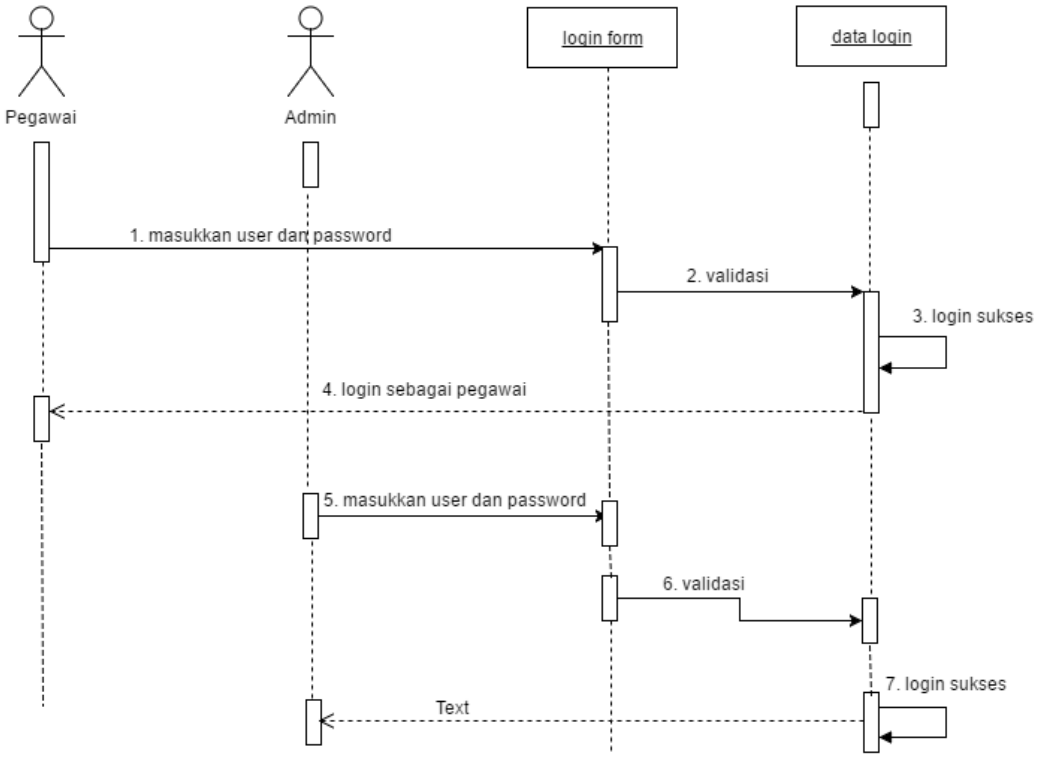
Gambar 2. 5 *Class Diagram*

2.12.4 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait)[35]. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan *output* tertentu.

Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari *class. Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Berikut dibawah ini adalah contoh dari *sequence diagram* yang ditunjukan pada Gambar 2.7



Gambar 2. 6 *Sequence Diagram*

2.13 MATLAB

MATLAB merupakan singkatan dari *MATrix LABoratory* dikarenakan setiap data pada MATLAB menggunakan dasar matriks. MATLAB adalah bahasa pemrograman tinggi, tertutup, dan case sensitive dalam lingkungan komputasi numerik yang dikembangkan oleh MathWorks. Salah satu kelebihannya yang paling populer adalah kemampuan membuat grafik dengan visualisasi terbaik. MATLAB mempunyai banyak tools yang dapat membantu berbagai disiplin ilmu. Ini merupakan salah satu penyebab industri menggunakan MATLAB. Selain itu MATLAB mempunyai banyak library yang sangat membantu untuk menyelesaikan permasalahan matematika seperti membuat simulasi fungsi, pemodelan matematika dan perancangan GUI. Saat ini matlab memiliki ratusan fungsi yang dapat digunakan sebagai problem solver mulai dari simple sampai masalah-masalah yang kompleks dari berbagai disiplin ilmu[36].

2.14 Pengujian Akurasi

kurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (true value atau reference value). Untuk dapat mengukur tingkat akurasi yang diperoleh dapat dihitung dengan persamaan sebagai berikut.

$$Akurasi = \frac{Jumlah\ Karakter\ Yang\ Dideteksi}{Jumlah\ Karakter\ Pada\ Sertifikat}\ X\ 100\% \qquad (2.\ 20)$$