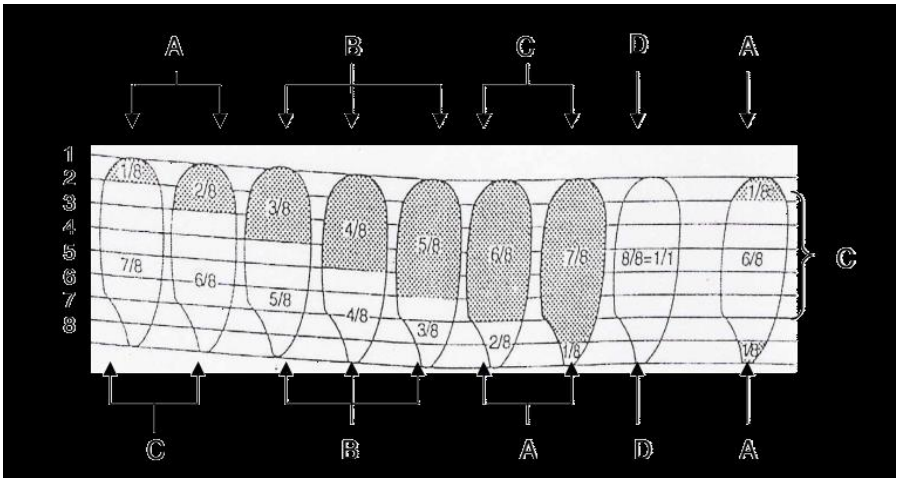
BAB 2
TINJAUAN PUSTAKA

2.1 Pengenalan Citra Beras

Beras adalah bagian bulir padi yang telah dipisah dari sekam. Sekam (Jawa merang) secara anatomi disebut 'palea' (bagian yang ditutupi) dan 'lemma' (bagian yang menutupi). Sebagaimana bulir serealia lain, bagian terbesar beras didominasi oleh pati (sekitar 80-85%). Beras juga mengandung protein, vitamin (terutama pada bagian aleuron), mineral, dan air. tetapi beras memiliki kualitas yang berbeda berdasarkan oleh karena itu pengawasan dalam menentukan kualitas beras yang akan beredar dipasaran merupakan hal penting karena menyangkut dengan tingkat penjualan dan konsumsi.

Berdasarkan produksi beras yang ada di Indonesia, beras dibagi kedalam beberapa jenis pada tugas akhir kali ini hanya menggunakan beras IR64/Setra Ramos adalah beras yang paling banyak beredar di pasaran, karena harganya yang terjangkau dan relatif cocok dengan selera masyarakat perkotaan. Normalnya beras jenis ini pulen jika dimasak menjadi nasi, namun jika telah berumur terlalu lama (lebih dari 3 bulan) maka beras ini menjadi sedikit pera, dan mudah basi ketika menjadi nasi. Beras ini memiliki ciri fisik agak panjang / lonjong, tidak bulat[11].

Kualitas beras memiliki tingkatan yang berbeda-beda spesifikasinya. Butir utuh, butir kepala, butir patah, butir menir merupakan parameter-parameter utama dari seluruh parameter kualitas dari SNI pada Gambar 2.1.



Gambar 2.1 Bagian-Bagian Beras : (A) Patahan Kecil, (B) Patahan Besar, (C) Beras Kepala, (D) Beras Utuh

2.2 Pengolahan Citra Digital

Pengolahan citra digital artinya melakukan pemrosesan gambar berdimensi dua melalui computer digital. Pengolahan citra adalah istilah umum untuk berbagai Teknik yang keberadaannya untuk memanipulasi dan dan memodifikasi citra dengan berbagai cara[12].

Suatu citra dapat didefinisikan sebagai fungsi f(x,y) berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitude f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut.

Citra digital dapat ditulis dalam bentuk matriks seperti pada persamaan (2.1)

$$f(x,y) = \begin{matrix} f(0,0) & \dots & f(0,N-1) \\ \dots & \dots & \dots \\ f(M-1,0) & \dots & f(M-1,N-1) \end{matrix} \quad (2.1)$$

Nilai pada suatu irisan baris dan kolom (pada posisi x,y) disebut dengan picture elements, image elements, pels, atau pixels. Istilah terakhir (pixel) paling sering digunakan pada citra digital.

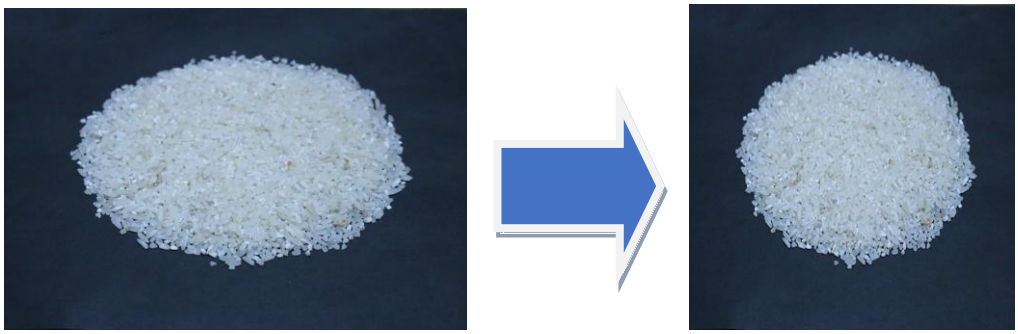
2.2.1 Operasi Pengolahan Citra

2.2.1.1 RGB

RGB adalah suatu model warna yang terdiri atas 3 buah warna, yaitu: merah (red), hijau (green), dan biru (blue) yang ditambahkan sebagai cara untuk menghasilkan bermacam-macam warna[12].

2.2.1.2 Resize

Resize atau penskalaan adalah operasi geometri yang digunakan untuk memperbesar atau memperkecil ukuran dari sebuah citra sesuai dengan ukuran yang dibutuhkan. Proses resize citra dapat mempengaruhi size citra lebih besar atau lebih kecil dan juga kualitas citra menjadi lebih baik atau lebih buruk. Resize dapat dilihat pada gambar 2.2.



Gambar 2.2 Resize

Contoh resize di atas mengubah citra yang awalnya memiliki ukuran 2736x1536px menjadi 512x512px. Proses penskalaan dapat dilakukan dengan rumus[6]:

$$x = \frac{pb * pp}{pa} \tag{2.2}$$

Keterangan:
x = nilai piksel baris matriks baru
pb = ukuran Panjang baru
pp = posisi piksel baris
pa = ukuran Panjang matriks lama

$$y = \frac{lb * lp}{la} \tag{2.3}$$

Keterangan:

y = nilai piksel kolom baru

lb = ukuran lebar matriks baru

lp = posisi posisi kolom

la =ukuran lebar matriks lama

2.2.1.3 Grayscale

Grayscale merupakan proses untuk mengubah warna menjadi keabu-abuan. Dengan mengubah nilai RGB setiap pixel gambar menjadi satu nilai yang sama sehingga setiap pixel memiliki nilai yang sama untuk ketiga unsur warna serta didapatkan nilai matriks *grayscale*.

Setelah melalui tahap resize, maka tahap selanjutnya adalah mengubah citra mode RGB menjadi *Grayscale*. Proses ini bertujuan untuk menyederhanakan nilai pixel pada sebuah citra, yang awalnya pada setiap pixel memiliki 3 nilai yaitu RGB menjadi hanya 1 nilai keabuan. Rumus yang digunakan untuk proses grayscale adalah sebagai berikut[13]:

$$X = 0.299*R + 0.587*G + 0.114*B \quad (2.4)$$

Dimana

X = nilai grayscale

R = nilai red

G = nilai green

B = nilai blue

Persamaan diatas merupakan salah satu rumus yang digunakan untuk mengonversi citra berwarna menjadi grayscale. Persamaan (2.2) dipilih dalam penelitian ini karena mata manusia secara alami lebih sensitive terhadap cahaya merah dan hijau. Maka dari itu, warna-warna ini diberi bobot yang lebih tinggi untuk memastikan bahwa keseimbangan intensitas relatif dalam citra grayscale yang dihasilkan mirip dengan citra warna RGB.

2.3 Feature Extraction

Feature extraction adalah proses transformasi data masukkan menjadi kumpulan fitur untuk mengambil representasi minimal dari data masukkan. Feature extraction merupakan proses mengambil ciri-ciri yang terdapat pada objek di dalam citra. Pada proses ini objek di dalam citra perlu dideteksi seluruh tepinya, lalu menghitung properti-properti objek yang berkaitan sebagai ciri.

Karakteristik fitur yang baik sebisa mungkin memenuhi persyaratan sebagai berikut[11]:

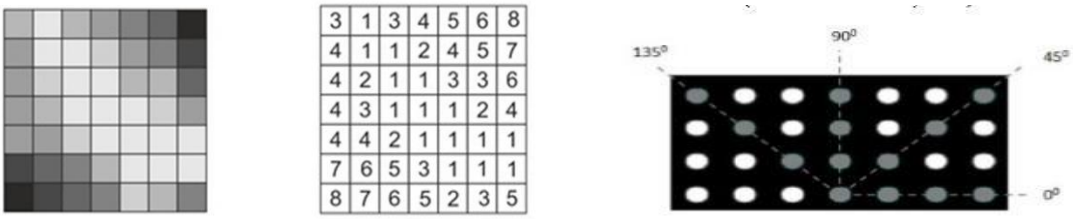
a. Dapat membedakan suatu objek dengan yang lainnya.
b. Kompleksitas komputasi yang tidak terlalu rumit.
c. Tidak terikat invariant terhadap transformasi.
d. Jumlahnya sedikit.

2.3.1 Gray Level Co-occurrence Matrix (GLCM)

Metode GLCM atau analisis tekstur merupakan salah satu metode untuk melakukan klasifikasi citra, GLCM adalah matriks persegi yang dapat menjelaskan sifat-sifat tertentu dengan distribusi spasial. GLCM menggunakan perhitungan tekstur pada orde kedua. Pada orde pertama, pengukuran tekstur menggunakan perhitungan statistik didasarkan pada nilai piksel citra asli semata, seperti varians, dan tidak memperhatikan hubungan ketetanggaan piksel. Sedangkan sehubungan antar pasangan dua piksel citra asli diperhitungkan pada orde kedua. Secara matematis, untuk mendapatkan nilai piksel kookurensi seperti pada persamaan (2.3) dimana d adalah jarak antara dua piksel yaitu (x1,y1) dan (x2,y2)[13].

$$P = \sum_{x-1}^{k} \sum_{y-1}^{k} \begin{Bmatrix} 1, if I(x,y) = i \ and \ I(x+d_x, y+d_y) = j \\ 0, lainnya \end{Bmatrix} \quad (2.5)$$

Untuk memperoleh ciri statistic orde dua adalah dengan menghitung probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Setiap elemen dari GLCM adalah hubungan antara dua piksel dengan warna abu-abu i dan j dengan jarak d dan arah θ. Fungsi sudut orientasi adalah menetukan arah arah hubungan tetangga dari setiap piksel pada citra. Pada gambar 2.3 (a) dijelaskan intensitas keabuan dari suatu citra, (b) merupakan area kerja matriks dari citra dan (c) menggambarkan orientasi sudut citra dianggap jarak antar piksel (d = 1) dengan orientasi sepanjang empat arah yaitu arah orientasi 0^o berarti acuan dalam arah horizontal atau sumbu x positif dari piksel-piksel referensi, 45 ^{o} dan 135 ^{o} acuan dalam arah diagonal, dan 90 ^{o} acuan dalam arah vertical[13].



Gambar 2.3 sudut yang digunakan pada matriks

Untuk kepentingan ilustrasi, ketetanggaan piksel dapat dipilih ke arah timur (kanan). Salah satu cara untuk merepresentasikan hubungan ini yaitu berupa (1,0), yang menyatakan hubungan dua piksel yang berjajar horizontal dengan piksel bernilai 1 diikuti dengan piksel bernilai 0. Berdasarkan komposisi tersebut, jumlah kelompok piksel yang memenuhi hubungan tersebut dihitung. Hal ini diilustrasikan pada Tabel 2.1 sebagai berikut[13].

Tabel 2.1 Ilustrasi: a.citra asli, b.framework, c.jumlah ketetanggaan

pixel	0	1	2	3
0	0	0	1	1
1	0	0	1	1
2	0	2	2	2
3	2	2	3	3

a. Citra asli

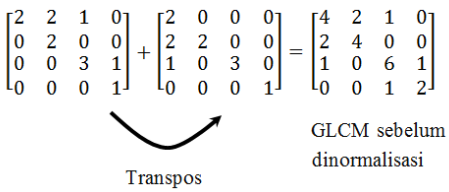
pixel	0	1	2	3
0	(0,0)	(0,1)	(0,2)	(0,3)
1	(1,0)	(1,1)	(1,2)	(1,3)
2	(2,0)	(2,1)	(2,2)	(2,3)
3	(3,0)	(3,1)	(3,2)	(3,3)

b. Framework

pixel	0	1	2	3
0	2	2	1	0
1	0	2	0	0
2	0	0	3	1
3	0	0	0	1

c. Jumlah ketetanggaan

Matriks pada Tabel di atas disebut dengan matriks *framework*. Matriks ini perlu diolah menjadi matriks yang simetris dengan cara menambahkan dengan hasil transposnya, sebagaimana diperlihatkan pada Gambar 2.4.



Gambar 2.4 Pembentukan matriks GLCM simetris

Nilai-nilai elemen GLCM perlu dinormalisasi sehingga jumlahnya bernilai 1 untuk menghilangkan ketergantungan pada ukuran citra. Dengan demikian, contoh di atas menjadi seperti berikut.

Gambar 2.5 Rumus matriks normalisasi

Hanya beberapa besaran yang diusulkan Haralick yang dipakai untuk mendapatkan fitur GLCM. Sebagai contoh, hanya menggunakan lima besaran untuk GLCM yakni *angular second moment* (ASM), *contras, inverse different moment* (IDM), entropi, dan korelasi[13].

a. ASM merupakan ukuran homogenitas citra dihitung dengan cara sebagai berikut:

$$\sum_{i=1}^{L}\sum_{j=1}^{L}(P(i,j))^2 \tag{2.6}$$

Dalam hal ini, L menyatakan jumlah level yang digunakan untuk komputasi.

b. Kontras merupakan ukuran keberadaan variasi atas keabuan piksel citra, dihitung dengan cara sebagai berikut:

$$\sum_{i=1}^{L}\sum_{j=1}^{L}(i-j)^2\,P(i,j) \tag{2.7}$$

c. IDM digunakan untuk mengukur homogenitas, dihitung dengan cara sebagai berikut:

$$\sum_{i=1}^{L}\sum_{j=1}^{L}\frac{P(i,j)}{1+(i-j)^2} \tag{2.8}$$

d. Entropy menyatakan ukuran ketidakteraturan aras keabuan di dalam citra. Nilainya tinggi jika elemen-elemen GLCM mempunyai nilai yang relatif

sama. Nilai rendah jika elemen-elemen GLCM dekat dengan nilai 0 atau 1. Rumus untuk menghitung entropy adalah sebagai berikut:

$$-\sum_{i=1}^{L}\sum_{j=1}^{L} P(i,j)\log\ P(i,j) \tag{2.9}$$

e. Korelasi yang merupakan ukuran ketergantungan linear antar nilai aras keabuan dalam citra dihitung menggunakan rumus sebagai berikut:

$$\sum_{i=1}^{L}\sum_{j=1}^{L}\frac{(i-\mu_i')*(j-\mu_j')*\ P(i,j)}{\sigma_i'*\ \sigma_j'} \tag{2.10}$$

Dengan

$$\mu_i' = \sum_{i=1}^{L}\sum_{j=1}^{L} i * P(i,j) \tag{2.11}$$

$$\mu_j' = \sum_{i=1}^{L}\sum_{j=1}^{L} j * P(i,j) \tag{2.12}$$

$$\sigma^2{}_i' = \sum_{i=1}^{L}\sum_{j=1}^{L} P(i,j)\,(i-\mu_i')^2 \tag{2.13}$$

$$\sigma^2{}_j' = \sum_{i=1}^{L}\sum_{j=1}^{L} P(i,j)\,(j-\mu_j')^2 \tag{2.14}$$

2.4 Machine Learning

Machine Learning, cabang dari kecerdasan buatan, adalah disiplin ilmu yang mencakup perancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengembangkan perilaku yang didasarkan pada data empiris, seperti dari sensor basis data. Sistem pembelajaran dapat memanfaatkan contoh (data) untuk menangkap ciri yang diperlukan dari probabilitas yang mendasarinya (yang tidak diketahui).

Data dapat dilihat sebagai contoh yang menggambarkan hubungan antara variable yang diamati. Focus besar penelitian pembelajaran mesin adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data. Kesukarannya terjadi karena himpunan semua perilaku yang mungkin, dari semua masukkan yang dimungkinkan, terlalu besar untuk diliput oleh himpunan contoh pengamatan (data pelatihan). Karena itu pembelajaran harus merapatkan (generalisasi) perilaku dari contoh yang ada untuk menghasilkan keluaran yang berguna dalam kasus-kasus baru[7].

2.4.1 Smooth Support Vector Machine (SSVM)

SSVM adalah pengembangan SVM dengan menggunakan teknik smoothing. Metode ini pertama kali diperkenalkan oleh Lee[14]. SVM memanfaatkan optimasi dengan quadratic programming, sehingga untuk data berdimensi tinggi dan data jumlah besar SVM menjadi kurang efisien. Oleh karena itu dikembangkan smoothing technique yang menggantikan plus function SVM dengan integral dari fungsi sigmoid neural network yang selanjutnya dikenal dengan *Smooth Support Vector Machine* (SSVM). Diberikan masalah klasifikasi dari *n* objek dalam ruang dimensi *R* sehingga susunan data berupa matriks *A* berukuran *n x p* dan keanggotaan tiap titik terhadap kelas {+1} atau {-1} yang didefinisikan pada diagonal matriks D berukuran ,problem optimasinya adalah

$$min_{w,b,\xi} \frac{c}{2}\xi'\xi + \frac{1}{2}(w'w + b^2) \tag{2.47}$$

Dengan kendala

$$D(Aw + eb) + \xi \geq e, \xi \geq 0 \tag{2.48}$$

Solusi problem

$$\xi = (e - D(Aw + eb)) \tag{2.49}$$

Dimana ξ adalah variable slack yang mengukur kesalahan klasifikasi. Kemudian dilakukan substitusi dan konversi, sehingga persamaan dapat ditulis sebagai berikut:

$$min_{w,b} \frac{c}{2}||e - D(Aw - eb)||_2^2 + \frac{1}{2}(w'w + b^2) \tag{2.50}$$

Fungsi objektif dalam persamaan tidak memiliki turunan kedua. Teknik smoothing yang diusulkan dilakukan dengan mengganti fungsi plus dengan p(x,a) yaitu integral dari fungsi sigmoid neural network atau dapat dituliskan sebagai berikut:

$$p(x,a) = x + \frac{1}{2}\log(1+\varepsilon^{-ax}), a > 0 \quad (2.51)$$

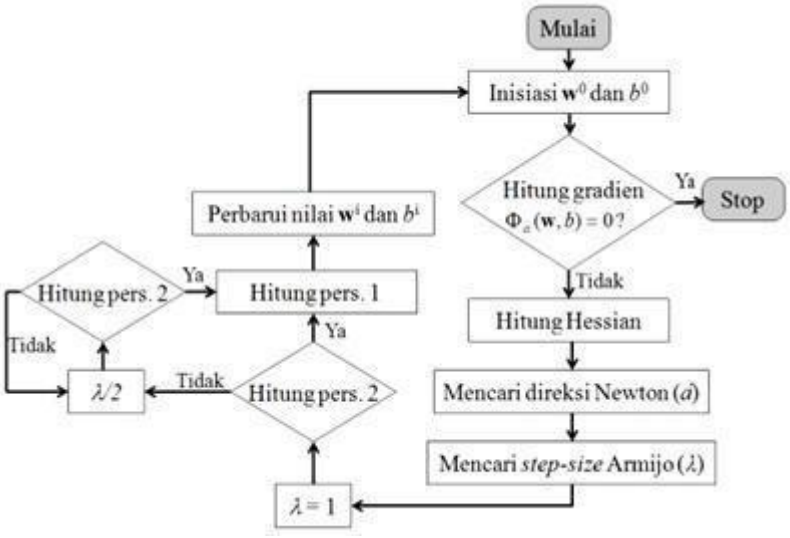
Dimana a adalah parameter smoothing. Dengan menggantikan fungsi plus dengan p(x,a) maka diperoleh model ssvm sebagai berikut:

$$min_{(w,b)\in Rp+1\phi a}(w,b) = min_{(w,b)\in Rp+1}\frac{c}{2}||P(e-D(Aw-eb)|\Big|_2^2 + \frac{1}{2}(w'w+b^2) \quad (2.52)$$

Secara umum, problem optimasi SSVM dapat dituliskan sebagai berikut:

$$min_{(w,b)\in Rp+1\phi a}(w,b) = min_{(w,b)\in Rp+1}\frac{c}{2}\left|\left|P\left(e-D\left(K\left(x_i,x_j\right)Dw-eb\right),a\right)\right|\right|\frac{2}{2}+\frac{1}{2}(w'w+b^2) \quad (2.53)$$

Yang diselesaikan dengan iterasi newton Armijo(Gambar 2.6) dan K(Xi,Xj) merupakan fungsi kernel yang dalam penelitian ini digunakan kernel gaussian atau bias dirumuskan berikut K(Xi,Xj) = exp(-y(||Xi-Xj||²) dengan parameter kernel y.



Gambar 2.6 Diagram Alir Algoritma Newton Armijo

Persamaan 1:

$$\phi_a(w_i, b_i) - \phi_a((w_i, b_i) + (\lambda_i d_i) \geq -\delta\lambda_i \nabla \phi_a(w_i, b_i) d_i \quad (2.54)$$

Persamaan 2:

$$w_{i+1}, b_{i+1} = (w_i, b_i) + (\lambda_i d_i) \quad (2.55)$$

Saat iterasi pada algoritma newton-armijo berhenti, diperoleh nilai w dan b yang konvergen. Dengan demikian fungsi pemisah yang diperoleh untuk kasus klasifikasi linear sebagai berikut:

$$F(x) = sign(w'x + b) \quad (2.56)$$

Sedangkan fungsi pemisah untuk kasus klasifikasi non-linear adalah sebagai berikut:

$$F(x) = sign(w'x + b) = sign(u'D'K(Xi, Xj) + b) \quad (2.57)$$

Perumusan program linear SVM 1 -norm adalah salah satu cara untuk memilih atribut (feature selection) di antara varian-varian norm SVM, problem linear tersebut adalah sebagai berikut:

$$\min_{(w,b,s,\xi) \in R^{(2p)+1+n}} Ce'\xi + e's \quad (2.58)$$

Dengan kendala

$$D(Aw + eb) + \xi \geq e \quad (2.59)$$

$$-s \leq w \leq s \quad (2.60)$$

$$\xi \geq 0 \quad (2.61)$$

Solusi dari w mampu menghasilkan model yang parsimony dan bersifat sparsity. Jika nilai dari elemen vector wp=0, maka variable p tidak berkontribusi dalam penentuan kelas. Kontribusi atribut atau variable predictor dapat dinilai dari besarnya nilai wi untuk masing-masing atribut, dengan l=1,1,…,p.

Support vector machine (SVM) adalah suatu sistem pembelajaran yang menggunakan ruang hipotesis dari suatu fungsi linear dalam suatu ruang dimensi berfitur tinggi yang dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di Annual Workshop on Computational Learning Theory.

2.5 Pengujian Sistem

Pengujian sistem adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaaan-perbedaan antara hasil yang diterapkan dengan hasil yang terjadi. Pengujian seharusnya meliputi tiga konsep berikut.

1. Demonstrasi validitas perangkat lunak pada masing-masing tahap disiklus pengembangan sistem.
2. Penentuan validitas sistem akhir dikaitkan dengan kebutuhan pemakai.
3. Pemeriksaan perilaku sistem dengan mengeksekusi sistem pada data sampel pengujian.

Pada dasarnya pengujian diartikan sebagai aktiitas yang dapat atau hanya dilakukan setelah pengkodean (kode program selesai). Namun, pengujian seharusnya dilakukan dalam skala lebih luas. Pengujian dapat dilakukan begitu spesifikasi kebutuhan telah dapat didefinisikan. Evaluasi terhadap spesifikasi dan perancangan juga merupakan teknik dipengujian. Kategori pengujian dapat dikategorikan menjadi dua, yaitu :

1. Berdasarkan ketersediaan logik sistem, terdiri dari *Black box testing* dan *White box testing*.
2. Berdasarkan arah pengujian, terdiri dari Pengujian *top down* dan Pengujian *bottom up*

2.5.2 Pengujian Black Box

Konsep *black box* digunakan untuk merepresentasikan sistem yang cara kerja di dalamnya tidak tersedia untuk diinspeksi. Di dalam *black box*, item-item yang diuji dianggap “gelap” karena logiknya tidak diketahui, yang diketahui hanya apa yang masuk dan apa yang keluar dari *black box*. Pada pengujian *black box*, kasus-kasus pengujian berdasarkan pada spesifikasi sistem. Rencana pengujian dapat dimulai sedini mungkin di proses pengembangan perangkat lunak. Teknik pengujian konvensional yang termasuk pengujian “black box” adalah sebagai berikut.

1. *Graph-based testing*
2. *Equivalence partitioning*
3. *Comparison testing*
4. *Orthogonal array testing*

Pada pengujian *black box*, kita mencoba beragam masukan dan memeriksa keluaran yang dihasilkan. Kita dapat mempelajari apa yang dilakukan kotak, tapi tidak mengetahui sama sekali mengenai cara konversi dilakukan. Teknik pengujian *black box* juga dapat digunakan untuk pengujian berbasis skenario, dimana isi dalam sistem mungkin tidak tersedia untuk diinspeksi tapi masukan dan keluaran yang didefinisikan dengan *use case* dan informasi analisis yang lain.

2.5.3 Pengujian Akurasi

Pengukuran kinerja klasifikasi pada data asli dan data hasil dari model klasifikasi dilakukan dengan menggunakan tabulasi silang (matriks konfusi) yang berisi informasi tentang kelas data asli yang direpresentasikan pada kolom matriks dan kelas data hasil prediksi suatu algoritma direpresentasikan pada baris klasifikasi[3].

Tabel 2.2 Matriks Konfusi

F_{ij}		Kelas asli ($_i$)		
		Kelas 1	Kelas 2	Kelas 3
Output Klasifikasi ($_j$)	Kelas 1	F_{11}	F_{21}	F_{31}
	Kelas 2	F_{12}	F_{22}	F_{32}
	Kelas 3	F_{13}	F_{23}	F_{33}

Ketepatan klasifikasi dapat dilihat dari akurasi:

$$akurasi = \frac{F_{11} + F_{22} + F_{33}}{F_{11} + F_{21} + F_{31} + F_{12} + F_{22} + F_{32} + F_{13} + F_{23} + F_{33}} \quad (2.62)$$

2.6 Matlab

MATLAB merupakan perangkat lunak yang digunakan untuk pemrograman, analisis, serta komputasi teknis dan matematis berbasis matriks. MATLAB adalah singkatan dari Matrix Laboratory karena mampu menyelesaikan masalah perhitungan dalam bentuk matriks. MATLAB versi pertama dirilis pada tahun

1970 oleh Cleve Moler. Pada awalnya, MATLAB didesain untuk menyelesaikan masalah-masalah persamaan aljabar linear. Seiring berjalannya waktu, program ini terus mengalami perkembangan dari segi fungsi dan performa komputasi.

Bahasa pemrograman yang kini dikembangkan oleh MathWorks Inc. menggabungkan proses pemrograman, komputasi, dan visualisasi melalui lingkungan kerja yang mudah digunakan. MATLAB juga memiliki keunggulan umum lainnya, seperti analisis dan eksplorasi data, pengembangan algoritma, pemodelan dan simulasi, visualisasi plot dalam bentuk 2D dan 3D, hingga pengembangan aplikasi antar muka grafis. Dalam ruang lingkup perguruan tinggi, MATLAB digunakan sebagai alat pembelajaran pemrograman matematika, teknik, dan sains pada level pengenalan dan lanjutan, sedangkan dalam dunia industri, MATLAB dipilih sebagai alat penelitian, pengembangan, dan analisis produk industri.

MATLAB dapat dioperasikan pada sistem operasi Windows, Linux, maupun macOS. Selain itu, MATLAB juga bisa dihubungkan dengan aplikasi atau bahasa pemrograman eksternal lainnya, seperti C, Java, .NET, dan Microsoft Excel. Dalam MATLAB tersedia pula kotak kakas (toolbox) yang dapat digunakan untuk aplikasi-aplikasi khusus, seperti pengolahan sinyal, sistem kontrol, logika fuzzy, jaringan saraf tiruan, optimasi, pengolahan citra digital, bioinformatika, simulasi, dan berbagai teknologi lainnya.

2.7 UML

UML *(Uniefied Modeling Language)* adalah notasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem berisi informasi dan fungsi, tetapi secara normal digunakan untuk memodelkan sistem komputer. Di dalam pemodelan objek guna menyajikan sistem yang berorientasi objek kepada orang lain, akan sangat sulit dilakukan dalam bentuk kode bahasa pemrograman.

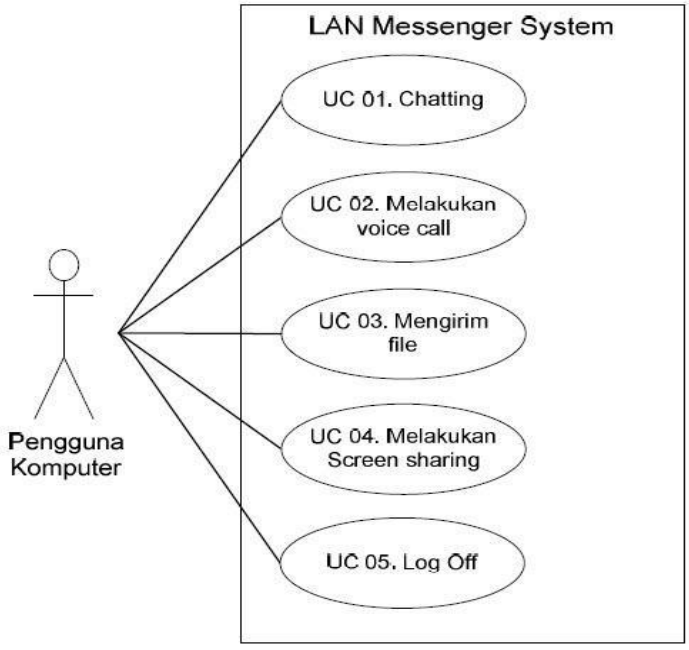
UML disebut sebagai bahasa pemodelan bukan metode. Bahasa pemodelan (sebagian besar grafik) merupakan notasi model yang digunakan untuk mendesain secara cepat. Bahasa pemodelan merupakan bagian terpenting dari metode. UML merupakan bahasa standar untuk penulisan *blueprint software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian. alat-alat dari

sistem perangkat lunak. UML biasanya disajikan dalam bentuk diagram atau gambar yang meliputi *class* beserta atribut dan operasinya, serta hubunganantar kelas. UML terdiri dari banyak diagram diantaranya *use case, diagram, activity diagram, class diagram, dan sequence diagram.*

2.7.1 Use Case Diagram

Dalam konteks UML, tahap konseptualisasi dilakuak dengan pembuatan *use case diagram* yang sesungguhnya merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunannya.

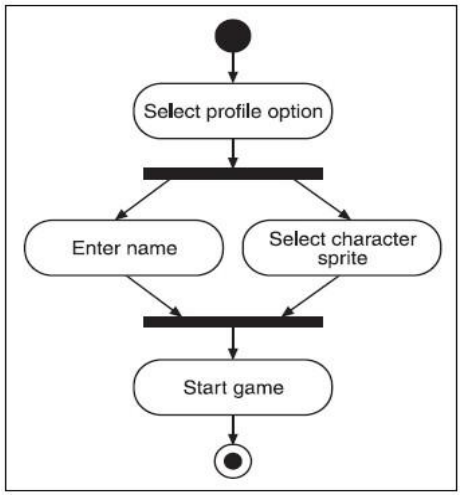
Use case diagram merupakan deskripsi lengkap tentang interaksi yang terjadi antara para aktor dengan sistem. Dalam hal ini, setiap objek yang berinteraksi dengan sistem merupakan aktor untuk sistem, sementara *use case* merupakan deskripsi lengkap tentang bagaimana sistem berprilaku kepada aktornya. Aktor dalam *use case diagram* digambarkan sebagai ikon yang berbentuk manusia, sementara *use case* digambarkan elips yang berisi nama *use case* yang bersangkutan. Untuk mempermudah pemahaman, aktor biasanya dituliskan sebagai kata benda, sementara *use case* biasanya ditulisakan dengan kata kerja.



Gambar 2.7 Use Case Diagram

2.7.2 Activity Diagram

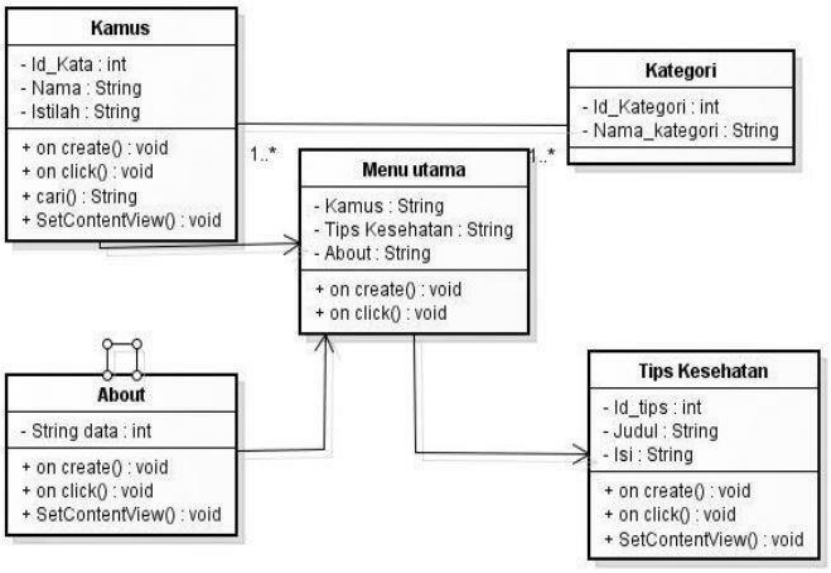
Activity Diagram pada dasarnya menggambarkan skenario secara grafis. A*ctivity diagram* cukup serupa dengan diagram alur (*flow chart),* yang membedakan hanya *swimlane* yang menunjukkan suatu *state* berada pada objek/kelas tertentu. Keunggulan dari *activity diagram* adalah lebih mudah dipahami dibanding skenario. Selain itu, dengan menggunakan *activity diagram,* kita dapat melihat dibagian manakah sistem dari suatu skenario akan berjalan.



Gambar 2.8 Activity Diagram

2.7.3 Class Diagram

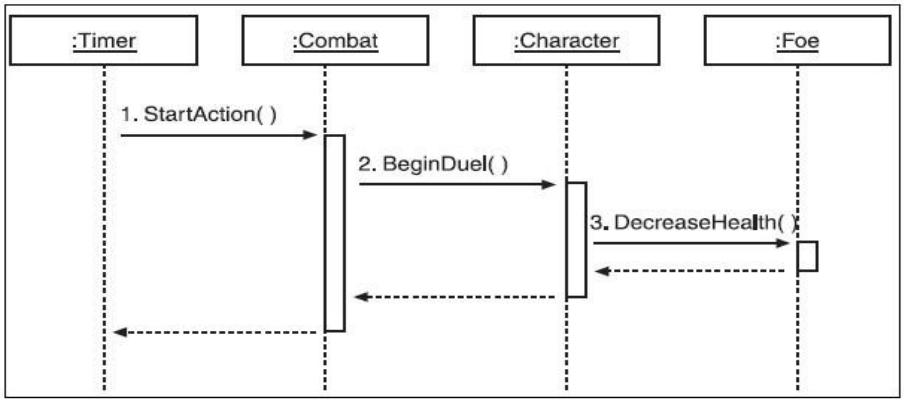
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendeinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* menggambarkan struktur dan deskripsi kelas, package, dan objek beserta hubungan satu sama lain seperti *containment,* pewarisan, asosiasi dan sebagainya.



Gambar 2.9 Class Diagram

2.7.4 Sequence Diagram

Diagram sekuen menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* berserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.



Gambar 2.10 Sequence Diagram