

BAB II

LANDASAN TEORI

2.1 Citra

Prasetyo (2011,p.1) menguraikan bahwa sebuah citra dapat di definisikan sebagai fungsi dua dimensi $f(x,y)$,dimana x dan y adalah koordinat spasial,dan amplitudo dari f pada sembarang kordinat (x,y) disebut *intensity* (intensitas) atau *gray level* (level keabuan) dari citra pada titik tersebut. Ketika x,y dan nilai intensitas dari f adalah semua terbatas, *discrete quantities*, kita sebut citra tersebut *digital image* (citra digital). Citra digital terdiri dari sejumlah elemen tertentu, setiap elemen mempunyai lokasi dan nilai tertentu.

Lestari (2012,p.2) menguraikan bahwa Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal video seperti gambar pada monitor televisi, dan bersifat digital yang dapat langsung disimpan pada suatu pita magnetik. Citra sebagai salah satu komponen multimedia memegang peranan penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi.

Arifin (2009,p.15) menjelaskan bahwa citra merupakan suatu keluaran dari suatu sistem perekaman data yang bersifat optik, analog ataupun digital. Perekaman data citra dapat dibagi menjadi dua yaitu:

a) Citra Analog

Citra analog yaitu terdiri dari sinyal - sinyal elektromagnetik yang tidak dapat dibedakan sehingga pada umumnya tidak dapat ditentukan ukurannya. Citra analog mempunyai fungsi yang kontinu. Hasil perekaman citra analog dapat bersifat optik yakni berupa foto (film foto konvensional) dan bersifat sinyal video seperti gambar pada monitor televisi.

b) Citra Digital

Citra digital terdiri dari sinyal - sinyal yang dapat dibedakan dan mempunyai fungsi yang tidak kontinu yakni berupa titik - titik warna pembentuk citra. Hasil perekaman citra digital dapat disimpan pada suatu media magnetik.

2.1.1 Citra Warna

Arifin (2009,p.16) menguraikan bahwa citra warna adalah citra dengan *system* grafik yang memiliki satu set nilai tersusun (*aset of ordered values*) yang menyatakan berbagai tingkat warna. Citra warna bukanlah seperti citra *grayscale*. Dimana setiap set nilai tersusun mewakili satu '*scale*' warna atau '*hue*'.

Agus (2013,p.309) menjelaskan bahwa Citra yang memiliki warna *grayscale* cenderung kurang menarik untuk dilihat dibandingkan dengan citra berwarna, karena kamera pada jaman dahulu hanya mampu menghasilkan citra dengan format warna *greyscale*, sehingga hasil citra tersebut menjadi kurang menarik untuk dilihat. Padahal, banyak citra zaman dahulu memiliki nilai sejarah yang cukup tinggi yang semestinya disampaikan dari generasi kegenerasi.

Gonzales yang diterjemahkan oleh Arifin (2009,p.17) menguraikan bahwa *system* yang dipakai untuk mewakili warna yaitu sistem *RGB* (*Red, Green,Blue*). Sistem RGB adalah *system* penggabungan antara warna - warna primer (*additiveprimary colours*) yaitu merah (*Red*), hijau (*Green*) dan biru (*Blue*) untuk memperoleh warna tertentu. Misalnya warna putih diperoleh dari hasil gabungan warna merah =255, hijau = 255, dan biru = 255. Dalam *system RGB*, warna putih cerah dinyatakan dengan *RGB* (255, 255, 255). *Range* nilai dari setiap warna primer adalah 0 sampai 255. Sehingga kemungkinan warna yang dapat terbentuk dengan sistem RGB adalah 256 x 256 x 256 yakni kurang lebih 16.7 juta warna. Tabel kode warna seperti tertampil pada tabel 2.1 berikut diperlihatkan beberapa kode warna hasil gabungan warna *RGB*.

Tabel 2.1 Kode Warna dalam Nilai Integer

Warna	Merah	Hijau	Biru
Hitam	0	0	0
Biru	0	0	255
Hijau	0	255	0
Cyan (Biru+Hijau)	0	255	255
Merah	255	0	0
Magenta (Merah+Biru)	255	0	255
Kuning (Merah+Hijau)	255	255	0
Putih (Merah+Hijau+Biru)	255	255	255
Abu-abu	128	128	128

2.1.2 Representasi Citra Digital

Prasetyo (2011,p.14) menguraikan bahwa sebuah citra dapat didefinisikan sebagai fungsi dua dimensi $f(x,y)$, dimana x dan y adalah kordinat spesial, dan amplitudo dari f pada sembarang pasangan kordinat (x,y) disebut intensitas citra (level keabuan) pada titik tersebut. Warna citra di bentuk oleh kombinasi citra 2-D individual. Misalnya, dalam sistem warna RGB, warna citra terdiri dari tiga komponen individu (*red, green, blue*). Untuk alasan ini, banyak cara di kembangkan untuk citra monokrom dapat di perluas ke citra berwarna oleh pemrosesan tiga komponen citra.

Arifin (2009,p.21) menjelaskan bahwa resolusi gambar dikatakan sebagai jumlah *pixel* yang terkandung di dalam suatu citra. Pada resolusi rendah keterperincian dan kedalaman citra akan hilang sama sekali dimana *pixel - pixel* individu jelas kelihatan, pada resolusi tinggi keterperincian data lebih nyata dan tajam. *Aspect Ratio* adalah suatu bilangan yang dapat diperoleh bila bilangan *pixel* mendatar dibagi dengan bilangan *pixel* tegak. Dalam pembesaran resolusi citra *Aspect Ratio* perlu sama agar citra tidak kelihatan *distorted* (menyimpang) dan alami.

Eka (2012,p.2) menjelaskan bahwa pembesaran citra adalah suatu proses yang dilakukan untuk memperbesar suatu citra digital dari ukuran semula menjadi ukuran yang berbeda sesuai dengan faktor pembesaran yang diinginkan. Proses ini memiliki dua langkah yaitu pembuatan lokasi *pixel* yang baru dan penempatan warna yang berdasarkan kepada nilai *gray* level terhadap lokasi baru yang dibuat sebelumnya.

2.2 Dasar - Dasar Warna

Prestyo (2011,p.177) menjelaskan bahwa penggunaan warna dalam pemrosesan citra dimotivasi oleh dua faktor utama. Pertama, warna adalah deskriptor yang *powerful* yang sering menyederhanakan indentifikasi objek dan pengestrakan dari *scene*. Kedua, manusia dapat melihat dengan jelas ribuan bentuk warna dan intensitas, dibandingkan dengan hanya dua lusin bentuk *gray*. Faktor kedua ini penting dalam tuntunan analisis citra. Walaupun proses di dalam merasakan dan menginterpretasikan warna adalah fenomena *physiopsychological* yang belum sepenuhnya dimengerti, kelainan fisik warna dapat diekpresikan pada basis formal yang didukung oleh eksperimen dan teori.

2.2.1 Model Warna

Prasetyo (2011,p.181) menjelaskan bahwa tujuan dari model warna adalah untuk memfasilitasi spesifikasi warna dalam beberapa standar. Esensinya, model warna adalah representasi sistem kordinat dan sub - space di dalam sistem di mana setiap warna direpresentasikan oleh titik tunggal.

2.3 Digital

Digital berasal dari kata Digitus, dalam bahasa Yunani yang berarti jari – jari. Apabila jari – jari seseorang dihitung, maka akan berjumlah sepuluh (10). Nilai sepuluh tersebut terdiri dari 2 radix, yaitu 1 dan θ . Oleh karena itu digital merupakan penggambaran dari suatu kondisi bilangan yang terdiri dari angka θ dan 1 atau *OFF* dan *ON* (*system* bilangan biner), dapat juga disebut dengan istilah bit, (*BINARY DIGIT*). Semua *system* komputer menggunakan *system* digital sebagai basis data nya.

2.4 Format *File* Gambar

Andreswari (2001,p.22) menjelaskan bahwa pada umumnya *file* gambar digunakan untuk menyimpan gambar yang ditampilkan dilayar ke dalam suatu media penyimpanan data. Untuk menyimpan sebuah *file* gambar ini digunakan salah satu format *file*. Ada banyak format *file* gambar yang dapat digunakan untuk menyimpan file gambar, salah satunya adalah *BMP*.

2.4.1 Format *File BMP (Bitmap)*

Format *file BMP* merupakan format grafis yang flexibel untuk *Platform Windows* sehingga dapat dibaca oleh program grafis manapun. Hal tersebut didukung kuat oleh Christian (2012,p.24) dengan berpendapat bahwa, format ini mampu menyimpan informasi dengan kualitas tingkat 1 *byte* sampai 24 *byte*. Format *file* ini mampu menyimpan gambar dalam mode warna *RGB*, *Grayscale*, *Indexed Color*, dan *Bitmap*.

Kelemahan format *file* ini adalah tidak mampu menyimpan *alpha channel* serta ada kendala dalam pertukaran *platform*. Kelebihan *tipe file BMP* adalah dapat dibuka oleh hampir semua program pengolah gambar. Baik *file BMP* yang terkompresi maupun tidak terkompresi, *file BMP* memiliki ukuran yang jauh lebih besar daripada tipe - tipe yang lain.

2.4.2 Format *File JPG/JPEG (Joint Photographic Experts Group)*

Christian (2012 , p.25) menjelaskan bahwa *Joint Photographic Experts (JPEG* dibaca *jay - peg*) di rancang untuk kompresi beberapa *full - color* atau *gray - scale* dari suatu gambar yang asli, seperti pemandangan asli di dunia ini. *JPEG* bekerja dengan baik pada *continous tone images* seperti *photographs* tetapi tidak terlalu bagus pada ketajaman gambar dan seni pewarnaan seperti penulisan, kartun yang sederhana atau gambar yang menggunakan banyak garis.*JPEG* sudah mendukung untuk *24-byte color depth* atau sama dengan 16,7 juta warna ($2^{24} = 16.777.216$ warna).

2.5 Penggunaan Pengolahan Citra Digital

Prasetyo (2011,P.2) menjelaskan bahwa cara paling mudah untuk mengembangkan pemahaman dasar tingkat aplikasi pengolahan citra digital adalah dengan mengklasifikasikan citra menurut sumbernya (misalnya, visual). Sumber energi citra yang digunakan saat ini adalah energi spektrum elektromagnetik.

Achmad (2013,p.4) menjelaskan bahwa suatu citra digital melalui pengolahan citra digital (*digital image processing*) dapat menghasilkan citra digital yang baru, termasuk didalamnya perbaikan citra (*image restoration*) dan peningkatan kualitas citra (*image enhancement*).

2.5.1 Citra Berskala Grey (Keabu-abuan)

Kadir (2013, p.23) menjelaskan bahwa sesuai dengan nama yang melekat, citra jenis ini menangani gradasi warna hitam dan putih, yang tentu saja menghasilkan efek warna abu - abu. Pada jenis gambar ini, warna dinyatakan dengan intensitas. Dalam hal ini, intensitas berkisar antara θ sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih.

2.5.2 Citra Biner

Kadir (2013,p.23) menjelaskan bahwa citra biner adalah citra dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dua kemungkinan (yaitu nilai θ dan 1). Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Citra jenis ini banyak dipakai dalam pemrosesan citra, misalnya untuk kepentingan memperoleh tepi bentuk suatu objek.

2.6 Pengertian *Template Matching*

Template matching adalah sebuah teknik dalam pengolahan citra digital untuk menemukan bagian - bagian kecil dari gambar yang cocok dengan *template* gambar. *Template matching* merupakan salah satu ide yang digunakan untuk menjelaskan bagaimana otak kita mengenali kembali bentuk - bentuk atau pola - pola. *Template* dalam konteks rekognisi pola menunjuk pada konstruk internal yang jika cocok (*match*) dengan stimulus penginderaan mengantar pada rekognisi suatu objek. Atau pengenalan pola terjadi jika terjadi kesesuaian antara stimulus

indera dengan bentuk mental internal. Gagasan ini mendukung bahwa sejumlah besar template telah tercipta melalui pengalaman hidup kita. Tiap - tiap *template* berhubungan dengan suatu makna tertentu.

Teori *Template matching* memiliki keunggulan dan kelemahan, yaitu :

Keunggulan :

- a. Jelas bahwa untuk mengenal bentuk, huruf atau bentuk - bentuk visual lainnya diperlukan kontak dengan bentuk - bentuk internal.
- b. *Template matching* adalah prosedur pengenalan pola yang sederhana yang didasarkan pada ketepatan konfigurasi informasi penginderaan dengan konfigurasi.

Kelemahan :

Jika perbandingan eksternal objek dengan internal objek 1:1, maka objek yang berbeda sedikit saja dengan template tidak akan dikenali. Oleh karena itu, jutaan template yang spesifik perlu dibuat agar cocok dengan berbagai bentuk geometri yang kita lihat dan kenal. Jika memang penyimpanan memori tentu seharusnya sangat kewalahan dan pencarian informasi akan memakan waktu, padahal pada kenyataannya tidak demikian.

Template Matching dapat dibagi antara dua pendekatan, yaitu : pendekatan berbasis fitur dan pendekatan berbasis *template*. Pendekatan berbasis fitur menggunakan fitur pencarian dan *template* gambar seperti tepi atau sudut, sebagai pembanding pengukuran matrik untuk menemukan lokasi *template matching* yang terbagus di sumber gambar.

2.6.1 Pendekatan Berbasis Fitur

Sebuah pendekatan berbasis fitur dapat dianggap; pendekatan dapat membuktikan lebih berguna, jika *template* gambar memiliki fitur yang kuat jika pencocokan di pencarian gambar bisa diubah dengan cara tertentu. Karena pendekatan ini tidak mempertimbangkan keseluruhan dari *template* gambar, komputasi dapat lebih efisien ketika bekerja dengan sumber gambar beresolusi lebih besar, sebagai pendekatan alternatif, berbasis *template*, mungkin memerlukan pencarian titik – titik yang berpotensi untuk menentukan lokasi pencocokan yang terbaik.

2.6.2 Pendekatan Berbasis *Template*

Untuk *template* tanpa fitur yang kuat, atau ketika sebagian besar *template* gambar merupakan gambar yang cocok, sebuah pendekatan berbasis *template* mungkin efektif, *template matching* berpotensi memerlukan sampling dari sejumlah besar poin, untuk mengurangi jumlah sampling poin dengan mengurangi resolusi pencarian dan *template* gambar oleh faktor yang sama dan melakukan operasi pada ketelitian gambar yang dihasilkan (multi resolusi, atau piramida, pengolahan citra), menyediakan pencarian titik data dalam pencarian gambar sehingga *template* tidak harus mempunyai pencarian titik data, atau kombinasi keduanya.

2.7 Pengertian Konvolusi (*convolution*)

Sebuah proses dimana citra dimanipulasi dengan menggunakan eksternal *mask* / *subwindows* untuk menghasilkan citra yang baru. Sedangkan Filtering tanpa menggunakan *ekternal mask* tetapi hanya menggunakan *pixel* tetangga untuk mendapatkan *pixel* yang baru. Konvolusi sangat banyak dipergunakan dalam pengolahan citra untuk memperhalus (*smoothing*), menajamkan (*crispening*), mendeteksi tepi (*edge detection*), serta efek lainnya.

1. *Embossing*

Embossing yaitu membuat citra seolah diukir pada permukaan selembar nikel. Kofisien jendela konvolusi memiliki bobot tengah bernilai θ & jumlah seluruh bobot = θ .

2. *Blurring*

Blurring (Pengaburan) yaitu filter *spasial low-pass* yang melenyapkan detail halus dari suatu citra. Pengaburan dicapai melalui konvolusi dari seluruh kofisien *mask* bernilai sama. *Blurring* ini perataan nilai *pixel - pixel* tetangga, makin besar ukuran *mask* maka makin besar efek pengaburan.

3. *Sharpening*

Sharpening (Penajaman) yaitu memperjelas detil suatu citra (menambah kontras) dengan penjumlahan atas citra tepi dengan citra aslinya maka bagian tepi objek akan terlihat berbeda dengan latarnya, sehingga citra terkesan lebih tajam.

4. *Edge Detection*

Deteksi tepi yaitu proses menentukan lokasi titik - titik yang merupakan tepi objek.

2.8 Pencocokan *Template* dan Konvolusi

Sebuah metode dasar *template matching* menggunakan konvolusi bayangan (*template*), disesuaikan dengan fitur tertentu dari *template matching*, yang ingin kita deteksi. Teknik ini dapat dengan mudah dilakukan pada gambar abu - abu atau tepi gambar. Hasil konvolusi akan di tempat tertinggi di mana struktur gambar sesuai dengan struktur bayangan, di mana nilai - nilai gambar besar dapat dikalikan dengan nilai - nilai bayangan besar.

Metode ini biasanya diimplementasi dengan terlebih dahulu memilih sebuah bagian dari pencarian gambar untuk digunakan sebagai *template*: Kita akan memanggil pencarian gambar $S(x, y)$, dimana (x, y) mewakili koordinat setiap pixel dalam pencarian gambar. Kita akan memanggil *template* $T(x_t, y_t)$, dimana (x_t, y_t) merupakan koordinat dari setiap *pixel* dalam *template*. Kemudian kita hanya memindahkan pusat (atau asal) dari *template* $T(x_t, y_t)$ atas setiap titik (x, y) dalam pencarian gambar dan menghitung jumlah produk antara koefisien dalam $S(x, y)$ dan $T(x_t, y_t)$ atas seluruh wilayah dari *template*. Karena semua kemungkinan posisi dari *template* yang berkenaan dengan pencarian gambar dianggap posisi terbaik. Metode ini kadang - kadang disebut sebagai '*Linear Spasial Filtering*' dan *template* disebut masker penyaring.

2.8.1 Mempercepat Proses

Di masa lalu, tipe spasial *filtering* biasanya hanya digunakan dalam solusi *hardware* khusus karena kompleksitas komputasi operasi, namun kita dapat mengurangi kompleksitas ini dengan penyaringan dalam domain frekuensi dari gambar itu, disebut sebagai 'frekuensi domain *filtering*', hal ini dilakukan melalui penggunaan teori konvolusi.

Cara lain untuk mempercepat proses pencocokan adalah melalui penggunaan dari suatu gambar piramida. Ini adalah serangkaian gambar, pada skala yang berbeda, yang terbentuk dengan berulang kali menyaring dan sub sampling gambar asli agar menghasilkan gambar resolusi berkurang berurutan. Gambar resolusi lebih rendah dapat dicari untuk *template* (dengan mengurangi resolusi yang sama), untuk menghasilkan posisi semula yang memungkinkan untuk mencari pada skala yang lebih besar. Foto yang lebih besar kemudian dapat dicari dalam jendela kecil di sekitar posisi mulai menemukan lokasi *template* terbaik. Metode lain yang dapat menangani masalah seperti ini antara lain terjemahan, skala dan rotasi gambar.

2.8.2 Penjelasan Tentang Kaleng Minuman

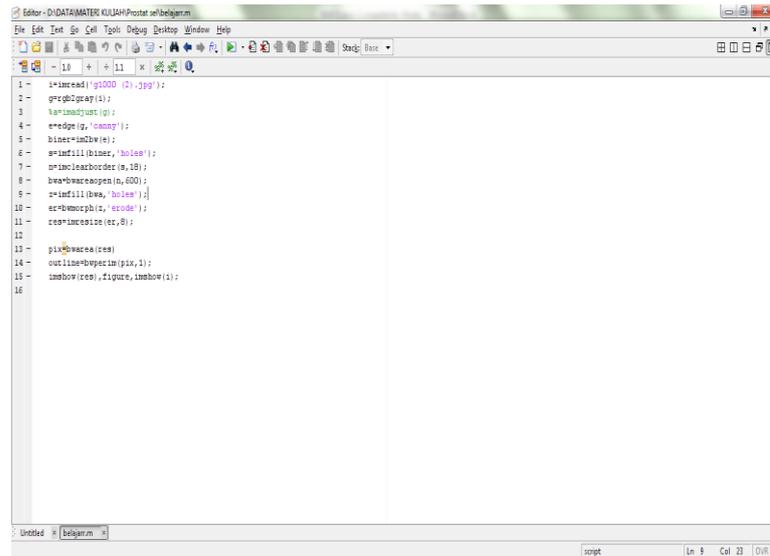
Kaleng merupakan lembaran baja yang dilapisi timah atau berupa wadah yang dibuat dari baja dan dilapisi timah putih tipis dengan kadar tidak lebih dari 1,00-1,25% dari berat kaleng itu sendiri. Terkadang lapisan ini dilapisi lagi oleh lapisan bukan metal yaitu untuk mencegah reaksi dengan makanan ataupun minuman di dalamnya. Kelebihan menonjol dari kemasan ini adalah bisa dilakukannya proses sterilisasi, sehingga makanan yang disimpan di dalamnya menjadi steril, tidak mudah rusak, dan awet. Dan pengertian dari baja adalah logam *alloy* yang komponen utamanya adalah besi, dengan karbon sebagai material *pengalloy* utama. Baja dengan peningkatan jumlah karbon dapat memperkeras dan memperkuat besi, tetapi juga lebih rapuh. Definisi klasik, baja adalah besi karbon *alloy* dengan kadar karbon sampai 5,1 persen, *alloy* dengan kadar karbon lebih tinggi dari ini dikenal dengan besi. Definisi yang lebih baru, baja adalah *alloy* berdasar besi yang dapat dibentuk secara plastik pada kaleng, daya ketahanan timah terhadap korosi juga tidak

sempurna, akan tetapi terhadap reaksi dengan makanan di dalamnya lebih lambat dibandingkan dengan baja. Bagi orang awam, kaleng sering diartikan sebagai tempat penyimpanan atau wadah yang terbuat dari logam dan digunakan untuk mengemas makanan, minuman, atau produk lain. Dalam pengertian ini, kaleng juga termasuk wadah yang terbuat dari aluminium. Kaleng timah merupakan pengembangan dari penemuan Nicolas Francois Appert pada dasawarsa 1800-an. Produk ini dipatenkan oleh seorang berkebangsaan Inggris, Peter Durand (1810). Berkat penemuan produksi massal, pada akhir abad ke-19, kaleng yang berbahan dasar timah menjadi standar produk konsumen.

2.9 Matlab

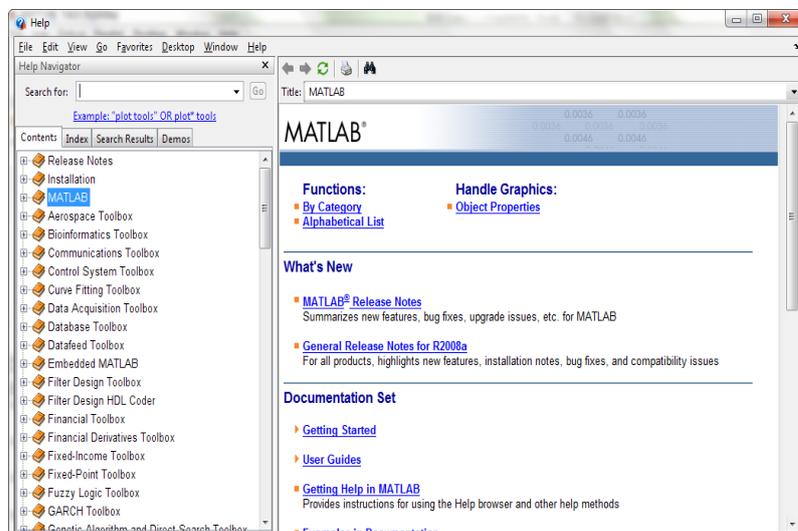
Tim Wahana Komputer (2013, p.21) menguraikan *MATLAB* merupakan sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat. Dikembangkan oleh *The Math Worrrks*, *MATLAB* memungkinkan manipulasi matriks, pemplotan fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan pengantarmukaan dengan program dalam bahasa lainnya. Tipe data dasar adalah matriks (*array*). *Matlab* tidak membutuhkan dimensi, oleh karena itu penggunaan memori dapat dihemat.

Semua data dianggap sebagai matrik. Pada suatu bilangan akan dianggap sebagai matrik 1x1. Untuk pengembangan algoritma *MATLAB* menyediakan antarmuka *command line*, sebuah interpreter untuk menangani bahasa pemrograman *MATLAB*, fungsi manipulasi string dan bilangan, *plotting function* dan kemampuan untuk membuat tampilan *GUI (Graphical User Interface)*. Pemrograman *MATLAB* menginterpretasikan perintah yang mempersingkat waktu pemrograman. Salah satu aplikasi pendukung matlab adalah *MCR (Matlab Compiler)*. *MCR* atau biasa di sebut *Matlab Compiler* adalah aplikasi bahasa C++ untuk menjalankan *exe portable* di laptop atau computer yang tidak terinstal *matlab*. Penjelasan tentang tampilan awal apabila program di jalankan pertama kali. Dapat dilihat di gambar 2.1 di bawah ini :



Gambar 2.3 *MATLAB Editor*

Penjelasan menu *help* yang ada di matlab, Menu *help* berfungsi untuk mengetahui koding yang terdapat di *matlab*. Dapat dilihat di gambar 2.4 di bawah ini :

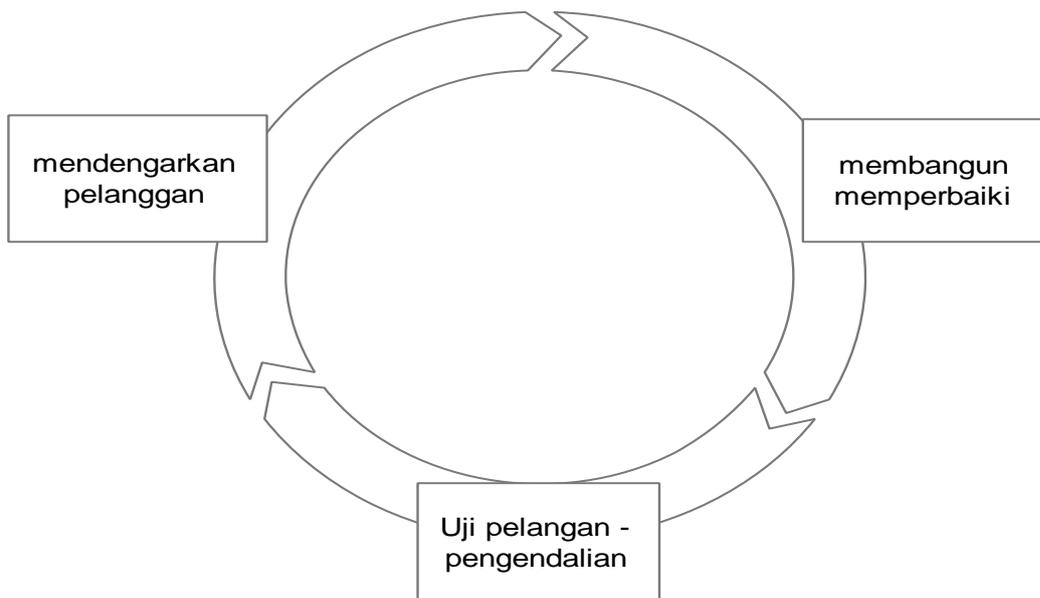


Gambar 2.4 *MATLAB Help System*

2.10 Metode Pengembangan *Prototype*

Pressman (2002,p.56) menguraikan bahwa *prototype* merupakan metode yang efektif dalam merancang perangkat lunak. *Prototype* dimulai dengan mengumpulkan kebutuhan. Pengembang dan pelanggan bertemu dan

mendefinisikan objek keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui dan kemudian melakukan perancangan kilat. Perancangan kilat berfokus pada penyajian dari aspek - aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan input dan format *output*). Perancangan kilat membawa kepada konstruksi sebuah *prototype*. *Prototype* tersebut dievaluasi oleh pelanggan dan dipakai untuk kebutuhan pengembangan perangkat lunak.



Gambar 2.5 *prototype*

Prototype juga dapat didefinisikan sebagai proses pengembangan suatu *prototype* secara cepat untuk digunakan terlebih dahulu dan ditingkatkan terus menerus sampai didapatkan *system* yang utuh. Merupakan proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk *prototype* dari perangkat lunak yang harus dibuat. Proses pada model *prototyping* dapat dijelaskan sebagai berikut :

- 1) Pengumpulan kebutuhan : *Developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian - bagian yang akan dibutuhkan berikutnya.

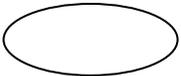
- 2) Perancangan : Perancangan dilakukan cepat dan rancangan mewakili semua aspek perangkat lunak yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
- 3) Evaluasi *prototype* : Klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan perangkat lunak.

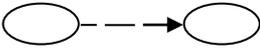
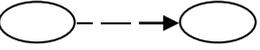
Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype - prototype* dibuat untuk memuaskan kebutuhan klien dan untuk membangun perangkat lunak lebih cepat, namun tidak semua *prototype* bisa dimanfaatkan. Demi kebutuhan klien lebih baik *prototype* yang dibuat diusahakan dapat dimanfaatkan.

2.11 Use Case Diagram

Rosa dan Salahudin (2011,p.131) menguraikan bahwa *use case* diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi - fungsi itu. Simbol - simbol yang digunakan untuk pembuatan *use case* diagram dapat dilihat pada tabel 2.2 berikut:

Tabel 2.2 Simbol *Use Case Diagram*

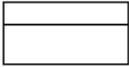
Simbol	Keterangan
<i>Use Case</i> 	Menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem.
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
Asosiasi 	Komunikasi antara <i>use case</i> dan aktor yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

Generalisasi 	Sebagai penghubung antara aktor- <i>use case</i> atau <i>use case-use case</i> .
<<Include>> 	<i>Include Relationship</i> (relasi cakupan) : Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.
<<Extend>> 	<i>Extend Relationship</i> : Memungkinkan relasi <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.

2.11.1 Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain - lain. Simbol - simbol yang digunakan untuk pembuatan *class diagram* dapat dilihat pada tabel 2.3 berikut:

Tabel 2.3 *Simbol Class Diagram*

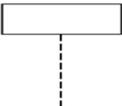
SIMBOL	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.11.2 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Simbol - simbol yang digunakan untuk pembuatan *class diagram* dapat dilihat pada tabel 2.4 berikut :

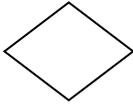
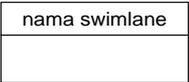
Tabel 2.4 Simbol Sequence Diagram

GAMBAR	NAMA	KETERANGAN
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas.

2.11.3 Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh *system*. (Rosa A.S dan Salahudin (2011,p.134). Simbol - simbol yang digunakan untuk pembuatan *activity diagram* dapat dilihat pada tabel 2.5 di bawah ini :

Tabel 2.5 *Simbol Activity Diagram*

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana ada pilihan aktivitas lebih dari satu.
Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.