BAB II

TINJAUAN PUSTAKA

2.1 Sejarah Ulos

Menurut pandangan orang-orang Batak, ada tiga sumber kehangatan (panas) bagi manusia, yaitu matahari, api, dan ulos. Kain ulos lahir dari kehidupan orang batak yang hidup di daerah pegunungan yang beriklim sejuk. Kain ulos digunakan untuk memberikan kehangatan terutama pada malam hari saat sumber kehangatan matahari sudah tidak ada.

Pada mulanya fungsi Ulos adalah untuk menghangkan badan, tetapi kini Ulos memiliki fungsi simbolik untuk hal-hal lain dalam segala aspek kehidupan orang Batak. Ulos tidak dapat dipisahkan dari kehidupan orang Batak. Bahkan, dahulu kain ini dipercaya sebagai benda yang keramat dan mengandung kekuatan magis dan memiliki daya istimewa yang dapat memberikan perlindungan kepada pemakainya.

Bila kain ini digunakan oleh laki-laki, bagian atasnya disebut dengan *ande-ande* sedangkan bagian bawahnya disebut *singkot*. Sebagai penutup Kepala disebut *tali-tali, bulang-bulang, sabe-sabe,*atau *detar*. Bila kain ini digunakan oleh perempuan batak toba, bagian bawah disebut *hean*, untuk penutup punggung disebut *hoba-hoba*. Dan bila di pakai sebagai selendang disebut *ampe-ampe*. Apabila digunakan sebagai penutup kepala disebut saong, dan untuk menggendong anak disebut *parompa*.Ulos atau yang sering juga disebut kain ulos merupakan busana khas Indonesia. Ulos merupakan pakaian adat dari Sumatera Utara yang secara turun temurun dikembangkan oleh masyarakat batak Sumatera. Cara membuat ulos yaitu menggunakan alat tenun.

2.1.1 Jenis Ulos

Pada dasarnya ulos yang dibuat dengan cara ditenun ini memiliki motif dan bentuk yang berbeda. Ulos terbagi atas jenis dan fungsinya, terdapat beberapa macam jenis ulos, di antaranya adalah:

a. Ulos *Jugia* (*Homitan*)

Ulos Jugia disebut juga ulos *Naso Ra Pippot* atau *Pinunsaan*. Ulos ini mengandung nilai budaya yang tinggi dan harganya sangat mahal. Oleh karenanya, ulos ini biasanya disimpan di *hombung* atau *parmona-monangan* (jenis lemari pada jaman dahulu). Menurut kepercayaan orang batak, ulos ini hanya dapat di pakai oleh orang yang sudah *saur matua* atau *naung gabe* (orang tua yang sudah mempunyai cucu dari anak laki-laki dan perempuan).

Namun walaupun telah mempunyai cucu seseorang belum masuk kategori *saur matua* jika masih ada anaknya yang belum menikah atau mempunyai keturunan, walaupun telah mempunyai cucu dari anaknya yang lain. Sulitnya persyaratan untuk dapat memakai ulos jugia menyebabkan ulos ini menjadi benda yang langka sehingga banyak orang yang tidak mengenalnya. Ulos ini sering menjadi barang warisan orang tua kepada anaknya, dan nilainya sama dengan *sitoppi* (emas yang dipakai oleh istri raja pada waktu pesta).



Gambar 2.1 Ulos Jugia

b. Ulos *Ragidup*

Ulos ini adalah kain tenun yang tertinggi derajatnya. Sebab pembuatannya sangatlah sulit. Kain tenun ulos jenis ini terdiri dari tiga bagian, yaitu 2 sisi yang ditenun sekaligus, dan 1 bagian tengah yang ditenun sendiri dengan motif yang rumit. Motif Ulos Ragidup ini harus terlihat seperti benar-benar lukisan hidup. Karenanya, ulos jenis ini sering

diartikan sebagai ulos yang melambangkan kehidupan dan doa restu untuk kebahagian dalam kehidupan.



Gambar 2.2 Ulos Ragidup

c. Ulos *Ragidup Silinggom*

Ulos inilah yang paling tepat diberikan kepada anak yang punya pangkat dan punya kuasa, dengan maksud, kita bisa (*marlinggom*) berlindung di bawah kebijaksanaannya. Ulos Ini juga diberikan kepada petinggi yang mendatangi kampung kita. *Ragidup Silinggom* tidak diperjual belikan dan hanya akan ditenun bila ada pemesannya.



Gambar 2.3 Ulos ragidup Silinggom

d. Ulos *Ragihotang*

Ulos ini derajatnya 1 tingkat di bawah ulos ragidup. Pembuatannya tidak serumit Ulos Ragidup. Namun, Ulos Ragihotang punya arti dan keistimewaan yang berhubungan dengan pekerjaan. Ulos ini pun sering dipakai dalam upacara adat kematian sebagai pembungkus atau penutup jenazah yang akan dikebumikan. Ulos jenis ini mengartikan bahwa pekerjaan seseorang di dunia ini telah selesai.



Gambar 2.4 Ulos Ragihotang

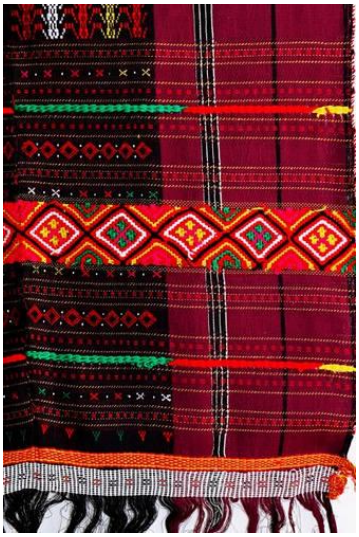
e. Ulos *Godang* atau Ulos *Sadum Angkola*

Disebut juga Sadum atau Sadum Angkola. Ulos ini mungkin yang paling bagus coraknya dan Indah warnanya di antara semua ulos, sehingga Ulos ini sangat mahal. Lebar Ulos ini juga cukup lebar diantara Ulos batak lainnya.

Secara Tingkatan Derajat Ulos Sadum/Ulos Godang masih dibawah Ulos ragidup, tapi kalau masalah harga, Ulos ini jauh lebih mahal dari Ulos ragidup. Ulos godang biasanya diberikan kepada anak kesayangan, yang membawa sukacita dalam keluarga. Inilah yang diharapkan dengan adanya pemberian ulos ini, supaya kelak si anak makin membawa hal-hal kebajikan yang banyak dan besar / *godang* (banyak), mencapai apa yang dicita-citakannya dan mendapat berkat yang banyak dan besar dari *Oppu Debata* (Tuhan). Pengertian *Oppu Debata* disini adalah Tuhan Allah yang Maha Kuasa, termasuk buat Orang Batak yang beragama Non Kristen,

namun perkembangan jaman seringkali pengertian *Oppu Debata* lebih ditujukan ke orang Batak yang beragama Kristen.

Biasanya Ulos Godang ini sering dibuat baju dan selain itu cara memakainya bisa dengan *diabithon* (dipakai) , *dihadang* (dililit di kepala atau bisa juga *ditengteng* atau *ditalitalihon* (dililit di pinggang).



Gambar 2.5 Ulos Sadum Angkola

f. Ulos *Runjat*

Ulos ini biasanya dipakai oleh orang kaya atau orang terpandang ketika *edang-edang* (menghadiri undangan). Setelan itu, ulos ini juga dapat diberikan kepada pengantin oleh keluarga dekat, misalnya oleh *tulang* (paman), dan *pariban* (kakak pengantiun perempuan yang sudah kawin). Ulos ini juga dapat diberikan pada waktu mangupa-upa dalam acara pesta gembira (*ulaon silas ni roha*).



Gambar 2.6 Ulos Runjat

g. Ulos *Sibolang*

Ulos *Sibolang* disebut juga *sibulang* yang diberikan untuk memberikan rasa hormat karena jasanya. Misalkan, Seorang *Ulubalang* yang mengalahkan musuh, atau yang bisa membinasakan binatang pemangsa yang mengganggu ketentraman Manusia. Jaman sekarang, ulos ini diberikan kepada *Amang ni hela* dan ulos ini disebut sebagai "*ulos pansamot na sumintahon*" supaya *Amang ni hela* tadi bisa menjadi tempat bersandar dan berlindung.

Perumpamaannya: " *na gogo mansamot jala parpomparan sibulang bulangan*". "*Marasar sihosari, di tombak ni panggulangan sai halak na gogo ma hamu mansari, jala parpomparan sibulangbulangan*. Ulos sibolang juga sering dipakai untuk menghadiri upacara kematian dan biasanya dililitkan di Kepala yang sudah Janda (Namabalu) saat kondisi suami meninggal.



Gambar 2.7 Ulos Sibolang

h. Ulos *Suri-suri Ganjang*

Disebut Ulos Suri-suri ganjang (biasanya orang batak hanya menyebutnya ulos suri-suri) karena coraknya berbentuk sisir memanjang. Dahulu ulos ini dipergunakan sebagai ampe-ampe/hande-hande. Pada waktu margondang (memukul gendang) Ulos ini dipakai kepada hula-hula untuk menyabut anak *boru*. Ulos ini juga dapat diberikan sebagai "*Ulos Tondi*" kepada pengantin. Keistimewaan ulos ini adalah panjangnya melebihi ulos biasa.



Gambar 2.8 Ulos Suri-suri Ganjang

i. Ulos *Mangiring*

Ulos ini mempunyai corak yang beriringan, yang melambangkan kesuburandan kesepakatan. Ulos ini sering diberikan untuk menggendongan anak, dengan harapan anak yang akan memakai *parompa* ini akan terus dalam iringan orang tuanya, kalau jaman dulu katanya ulos ini sering dihadiahkan kepada dua kekasih ataupun pasangan muda.



Gambar 2.9 Ulos Mangiring

j. Ulos Bintang *Maratur*

Sebagaimana namanya mempunyai arti yang menggambarkan jejerean bintang yang teratur. Biasanya Ulos ini dipakaikan dengan cara dijadikan Selendang (*Sitalihononton*). Jenis ragi ini bermakna kepatuhan dan kerukunan dalam ikatan kekeluargaan.



Gambar 2.10 Ulos Maratur

k. Ulos *Sitolu Tuho*

Keistimewaan dari ulos ini terlihat dalam motif gorganya terdapat *tolu* (tiga) *tuho* (Cabang/Bidang Arsiran). Ulos ini menggambarkan Simbol kekeraban orang batak yaitu *Dalihan Na Tolu* (di Ulos sering ditulis " *Paratur ni Parhundulon* ").



Gambar 2.11 Ulos Sitolu Tuho

l. Ulos *Bolean*

Ulos ini diberikan kepada anak yang kehilangan orangtua nya. Membelai-belai, dimaksudkan untuk menghilangkan rasa sedih (*Mangapuli*) agar hati anak yang sudah kehilangan Orang Tua tabah menghadapinya.[2]



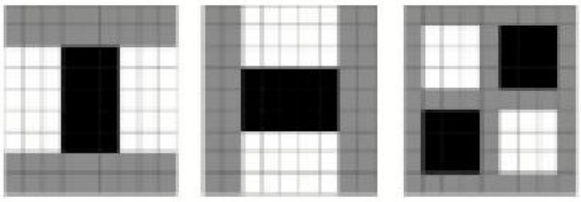
Gambar 2.12 Ulos Bolean

2.2 Surf

Metoda *Speed-Up Robust Features* (SURF) merupakan sebuah metode deteksi fitur yang menggunakan *keypoint* dari sebuah gambar. *Keypoint* itu sendiri adalah bagian-bagian dari sebuah gambar yang nilainya kuat atau tetap ketika mengalami perubahan skala, rotasi, *blurring*, transformasi 3 dimensi, pencahayaan dan juga perubahan bentuk [8]. Perubahan bentuk itu bisa terjadi karena bentuk gambar data awal yang tidak utuh atau tidak sempurna gambar yang ada di dalam gambar sampel tersebut. Gambar uji yang tidak utuh mungkin karena ada objek lain yang menutupi, atau pengambilan gambar yang tidak sempurna, atau keadaan objek itu sendiri yang telah mengalami perubahan.

2.2.1 *Scale-Space Ekstrema Detection* Pada SURF

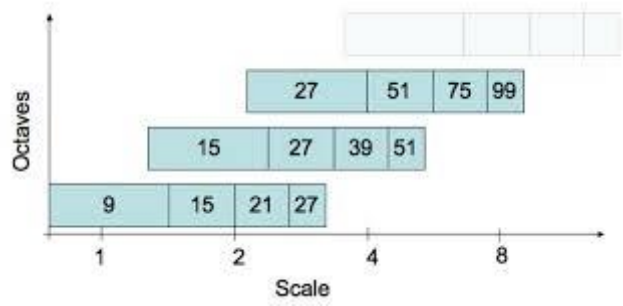
Scale Space dibagi kedalam bilangan octave, dimana satu octave mempresentasikan respon filter yang melakukan proses konvolusi gambar masukan dengan filter yang ukurannya bertambah tinggi. Dalam SURF, level paling rendah dalam *scale space* diperoleh dari output filter 9x9. Filter ini berhubungan dengan nilai *Gaussian* σ =1.2. Ketika membuat filter yang lebih besar, ada beberapa faktor yang harus di perhitungkan terlebih dahulu. Kenaikan ukuran dibatasi oleh panjang lobe yang positif atau negatif dari turunan *Gaussian* orde kedua.



Gambar 2.13 Turunan Orde Dua Gaussian

Pada menunjukan *lobe* positif yang mempresentasikan dengan warna kotak putih, sedangkan *lobe* negatif dipresentasikan dengan warna kotak hitam. menunjukan turunan orde dua *Gaussian* terhadap x,y dan xy.

Karena dalam proses ini membutuhkan posisi *central pixel* tetap, maka kenaikan ukuran *pixel* minimum 2 (satu *pixel* untuk setiap sisi), karena ada tiga *lobe* di masing-masing filter adalah 6. Dengan demikian ukuran filter-filter untuk octave pertama adalah 9x9, 15x15, 21x21, 27x27. Sedangkan untuk setiap octave yang baru, ukuran kenaikan filter adalah *double* (dari 6 menjadi 12, menjadi 24, dan menjadi 48), sehingga ukuran filter octave kedua adalah 15,27,39,51. Octave ketiga 27,51,75,99, dan jika gambar asli masih lebih besar dari ukuran filter, maka octave keempat dibuat yaitu dengan ukuran filter 51,99,147 dan 195. Pada **Error! Reference source not found.** menunjukan contoh scale space.



Gambar 2.14 ***Scala Space*** **pada SURF**

2.2.2 Keypoint Localization Pada SURF

Untuk mencari lokasi *keypoint* yang invarian terhadap rotasi, maka dilakukan proses yang terdiri dari tiga langkah seperti berikut, yaitu:

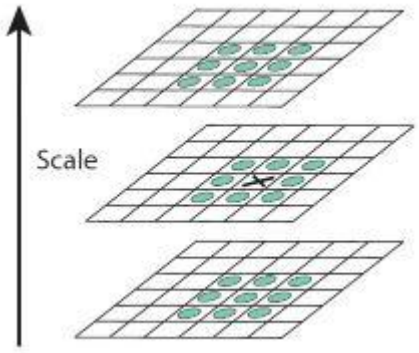
1) Langkah pertama : Menentukan nilai ambang penentuan lokasi apakah *keypoint* atau tidak. Ketika nilai ambang dinaikan, jumlah *keypoint* yang terdeteksi lebih kecil dan sebaliknya. Oleh karena itu, nilai ambang bisa disesuaikan untuk setiap aplikasi.
2) Langkah kedua : Sesudah proses penentuan ambang, selanjutnya proses *non-maxima suppression* dilakukan untuk mencari sekumpulan calon *keypoint*. Proses ini dilakukan dengan membandingkan masing-masing pixel gambar pada *scale space* dengan 26 tetangga, yang terdiri atas 8 titik di skala asli dan 9 di masing-masing skala diatas dan dibawahnya (total keseluruhan 9+8+9=26). Hasil dari *non-maxima suppression* adalah *keypoint* yang dihasilkan gambar.
3) Langkah ketiga : Proses ini adalah proses terakhir dimana dalam mencari *keypoint* yaitu dengan menggunakan interpolasi dengan data yang dekat dengan *keypoint*. Hal tersebut dilakukan dengan mencocokan quadratic 3 dimensi. H(x, y, σ) menyatakan determinan Hessian, yang didefinisikan sebagai persamaan berikut:

$$\boldsymbol{H(x) = H + \frac{\partial H^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 H}{\partial x^2} x} \quad \textbf{(0-1)}$$

Lokasi ekstrim yang diinterpolasi, $x = (x, y, \sigma)$, ditemukan dengan mencari turunan dari fungsi ini dan diberi nilai nol, yang dapat digambarkan dengan persamaan berikut:

$$\boldsymbol{x = -\frac{\partial^2 H^{-1}}{\partial x^2} \frac{\partial H}{\partial x}} \quad \textbf{(0-2)}$$

Jika lebih besar dari 0.5 pada x,y atau arah σ, lokasi perlu disesuaikan lagi dengan melakukan interpolasi lagi. Prosedur ini diulang sampai nilai kurang dari 0.5. Bila interpolasi telah dilakukan beberapa kali tetapi hasilnya tidak memenuhi nilai di bawah 0.5, maka *keypoint* tersebut dihapus. *Keypoint* yang tersisa adalah yang stabil.

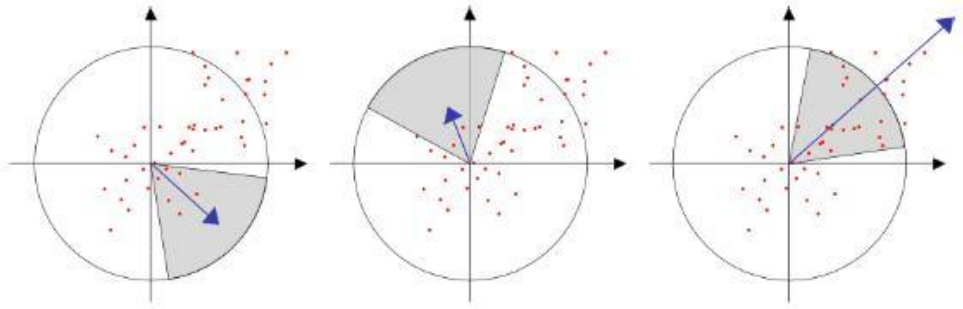


Gambar 2.15 Non-Maxima Suppression

i. Orientation Assignment Pada SURF

Agar *keypoint* invarian terhadap rotasi gambar, maka harus melakukan identifikasi orientasi yang dihasilkan disekitar *keypoint*. Untuk tujuan tersebut, langkah pertama yang dilakukan adalah dengan menghitung respon Haar Wavelet dalam arah x dan arah y pada daerah piksel tetangga disekitar *keypoint* yang berbentuk lingkaran yang dengan jarak 6s, dimana s merupakan skala dari *keypoint* yang terdeteksi. Ukuran wavelet ditentukan menjadi 4s. Filtering dipercepat prosesnya dengan integral gambar, dan keluaran yang dihasilkan direpresentasikan sebagai titik-titik dalam ruang dengan respon horizontal sepanjang sumbu axis dan respon vertikal sepanjang sumbu koordinat. Orientasi

dominan diperkirakan dengan menghitung jumlah semua respon dalam pergeseran orientasi dengan ukuran window π / 3, seperti yang diunjukan pada **Error! Reference source not found.**, dimana respon horizontal dan vertikal dalam window tersebut dijumlahkan. Dari kedua respon yang dijumlahkan akan menghasilkan sebuah vektor orientasi lokal. Vektor terpanjang mendefinisikan orientasi *keypoint*. Ukuran pergeseran window merupakan sebuah parameter yang harus dipilih secara hati-hati.



Gambar 2.16 Pergeseran Orientasi

2.2.3 Keypoint Deskriptor SURF

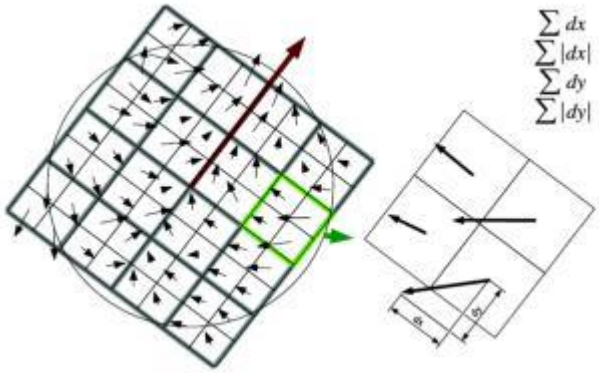
Untuk proses ekstraksi deskriptor, langkah pertama yang dilakukan adalah membuat daerah kotak sekitar *keypoint*, dimana *keypoint* sebagai pusat dari daerah kotak tersebut, dan orientasinya di sekitar orientasi yang ditentukan.



Gambar 2.17 Orientasi Perhitungan pada Skala yang Berbeda

Pada tahap selanjutnya, respon wavelet dx dan dy dijumlahkan masing-masing *sub-region*. Selain memberikan informasi tentang polar dari perubahan intensitas, juga akan dihasilkan jumlah nilai

absolut dari respon |dx| dan |dy|. Masing-masing *sub-region* mempunyai empat dimensi deskriptor vertor v, yaitu dx, dy, |dx| dan |dy|. Untuk 4x4 sub-region, maka panjang vektor deskriptornya adalah 64 seperti pada **Error! Reference source not found.**.



Gambar 2.18 Panjang Vektor Deskriptor SURF

Sementara untuk mendapatkan hasil vektor v dapat dilihat pada persamaan berikut :

$$\boldsymbol{Usubregion} = [\sum \boldsymbol{dx}, \sum \boldsymbol{dy}, \sum |\boldsymbol{dx}|, \sum |\boldsymbol{dy}|] \quad \textbf{(0-3)}$$

Respon wavelet juga invarian terhadap pencahayaan, sedangkan sifat invarian terhadap kontras dicapai melalui pembentukan deskriptor ke dalam satuan vektor.

2.3 Matriks Hessian Affine Region

Matrik adalah susunan bilangan yang diatur berdasarkan baris dan kolom. Bilangan – bilangan tersebut dinamakan entri dalam matrik atau disebut juga elemen (unsur). Matrik Hessian adalah matrik yang setiap elemennya dibentuk dari turunan parsial kedua dari suatu fungsi. Misalkan f(x) fungsi dengan n variabel yang memiliki turunan parsial kedua dan turunannya kontinu, matrik Hessian f(x) ditulis H adalah :

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial_n} \\ \frac{\partial^2 f}{\partial x_2 \partial_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial_n} \\ \cdots & \cdots & \ddots & \cdots \\ \frac{\partial^2 f}{\partial x_n \partial_1} & \frac{\partial^2 f}{\partial x_n \partial_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Matrik Hessian dapat digunakan untuk melakukan uji turunan kedua fungsi lebih dari satu variabel, yaitu untuk mengidentifikasi optimum relatif dari nilai fungsi tersebut. Penggolongan titik stasioner fungsi dua variabel dengan menggunakan matriks Hessian misalkan f(x) = F(x1, …, xn) adalah fungsi bernilai real dimana semua turunan parsialnya kontinu. Misalnya x0 adalah titik stasioner dari F dan didefinisikan H = H(x0) dengan persamaan Hij = Fxi, yj (x0). H (x0) adalah Hessian dari F pada x0

Titik stasioner dapat digolongkan sebagai berikut :

1. x0. Adalah suatu minimum relatif dari F jika jika H(x0.) definit positif

2. x0. Adalah suatu maksimum relatif dari F jika H(x0.) definit negatif

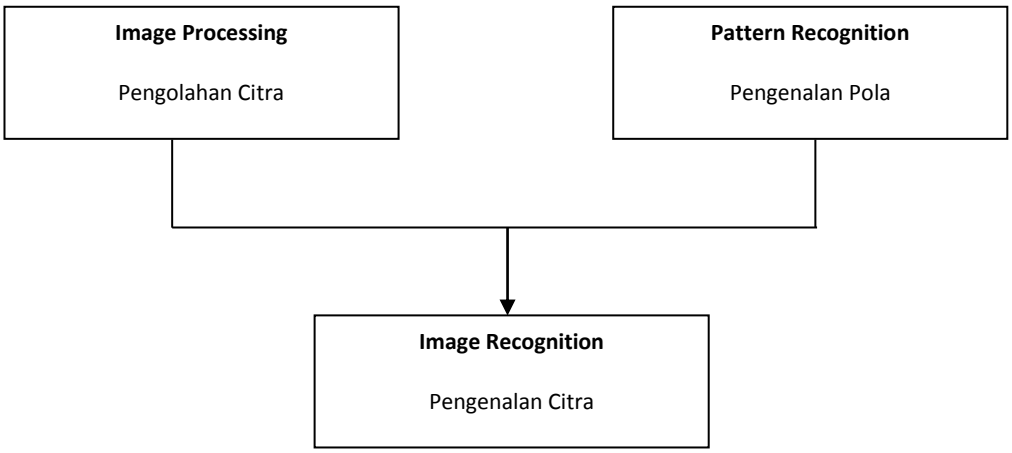
3. x0. Adalah suatu titik pelana dari F jika H(x0.) indefinite

2.4 Pengolahan Citra

Secara harafiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh oleh alat-alat optik, misalnya mata

pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam [4].

Bila pengenalan pola dihubungkan dengan pengolahan citra, diharapkan akan terbentuk suatu sistem yang dapat memproses citra masukan sehingga citra tersebut dapat dikenali polanya. Proses ini disebut pengenalan citra atau *image recognition.* Proses pengenalan citra ini sering diterapkan dalam kehidupan sehari-hari.



Gambar 2.19 Bagan Pengolahan Citra

Pengolahan citra dan pengenalan pola menjadi bagian dari proses pengenalan citra. Kedua aplikasi ini akan saling melengkapi untuk mendapatkan ciri khas dari suatu citra yang hendak dikenali [5]. Secara umum tahapan pengolahan citra digital meliputi akusisi citra, peningkatan kualitas citra, segmentasi citra, representasi dan uraian, pengenalan dan interpretasi.

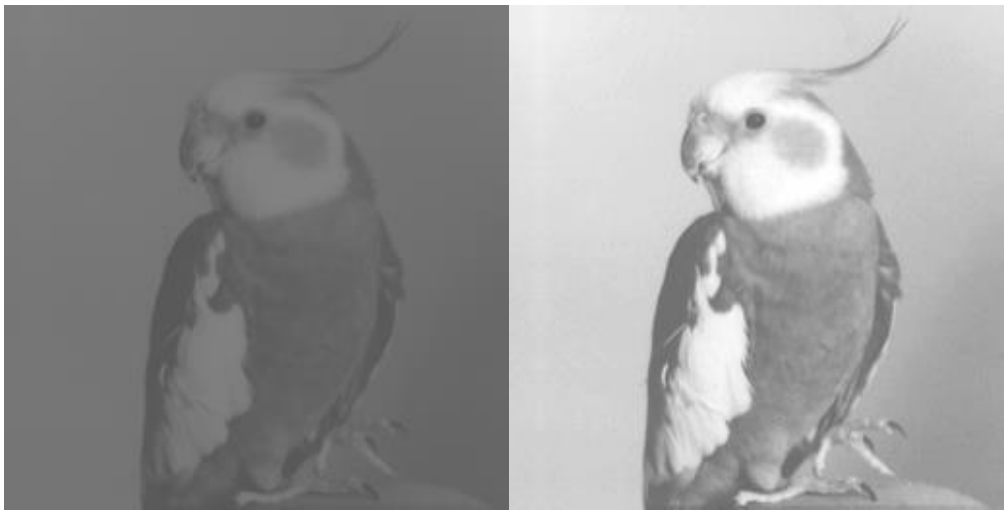
2.4.1 Pengertian Pengolahan Citra

Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang.

Agar citra yang mengalami gangguan mudah diinterpretasi (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain

yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (*image processing*).

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Sebagai contoh, citra burung nuri pada Gambar 2.5(a) tampak agak gelap, lalu dengan operasi pengolahan citra kontrasnya diperbaiki sehingga menjadi lebih terang dan tajam (b).



(a) **(b)**

Gambar 2.20 Citra burung nuri yang agak gelap, (b) Citra burung yang telah diperbaiki kontrasnya.

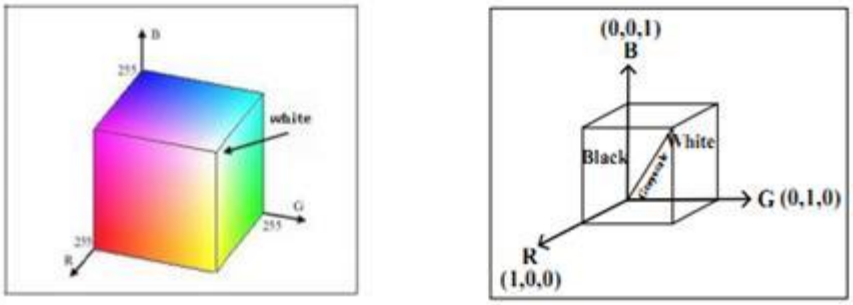
2.4.2 Penganalan Warna Citra Warna RGB

Warna adalah spektrum tertentu yang terdapat di dalam suatu cahaya sempurna (berwarna putih). Nilai warna ditentukan oleh tingkat kecerahan maupun kesuraman warna. Nilai ini dipengaruhi oleh penambahan putih ataupun hitam [5]

Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang paling lebar adalah red (R), green (G) dan blue (B). Ketiga warna tersebut merupakan warna pokok yang biasa disebut RGB. Warna lain dapat diperoleh dengan mencampurkan ketiga warna pokok tersebut dengan perbandingan tertentu. Setiap warna pokok mempunyai intensitas sendiri

dengan nilai maksimum 255 (8-bit). Misal warna kuning merupakan kombinasi warna merah dan hijau sehingga nilai RGB: 255 255 0.

RGB disebut juga ruang warna yang dapat divisualisasikan sebagai sebuah kubus seperti gambar 2.6, dengan tiga sumbunya yang mewakili komponen warna merah (red) R, hijau (green) G, biru (blue) B. Salah satu pojok alasnya yang Sistem Klasifikasi Jenis dan Kematangan Buah Tomat Berdasarkan Bentuk dan Ukuran serta Warna Permukaan Kulit Buah Berbasis Pengolahan Citra Digital berlawanan menyatakan warna hitam ketika R = G = B = 0, sedangkan pojok atasnya yang berlawanan menyatakan warna putih ketika R= G= B= 255 (sistem warna 8 bit bagi setiap komponennya).



Gambar 2.21 Ruang Warna RGB

Kamera yang digunakan pada Handphone ini akan menangkap gambar dari objek dan kemudian merepresentasikannya kedalam bentuk citra warna RGB untuk kemudian dilakukan pengolahan citra tersebut sehingga menghasilkan data-data objek yang dibutuhkan.

2.4.3 Jenis-jenis Citra Digital

Ada banyak cara untuk menyimpan citra digital di dalam memori. Cara penyimpanan menentukan jenis citra digital yang terbentu. Beberapa jenis citra digital yang sering digunakan adalah citra biner, citra grayclale, dan citra warna.

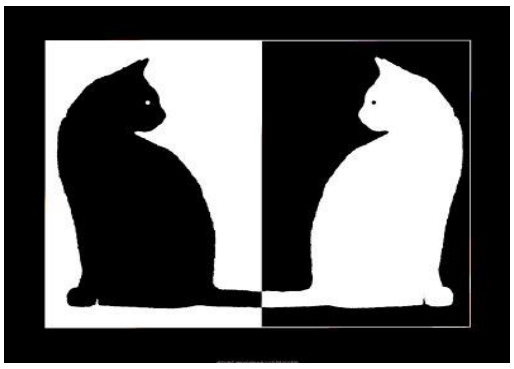
2.4.3.1 Citra Biner (Monokrom)

Citra biner (*binary image*) adalah citra yang hanya mempunyai dua nilai derajat keabuan: hitam dan putih seperti yang ditunjukkan pada gambar 2.7. Meskipun saat ini citra berwarna lebih disukai karena memberi kesan yang lebih kaya daripada citra biner, namun tidak membuat citra biner mati. Pada beberapa aplikasi citra biner masih tetap dibutuhkan, misalnya citra logo instansi (yang hanya terdiri atas warna hitam dan putih), citra kode batang (*bar code*) yang tertera pada label barang, citra hasil pemindaian dokumen teks, dan sebagainya [7]

Proses awal yang dilakukan dalam menganalisis objek di dalam citra biner adalah segmentasi objek. Proses segmentasi bertujuan mengelompokkan *pixel-pixel* objek menjadi wilayah (*region*) yang merepresentasikan objek. Ada dua pendekatan yang digunakan dalam segmentasi objek:

1. Segmentasi berdasarkan batas wilayah (tepi dari objek). *Pixel-pixel* tepi ditelusuri sehingga rangkaian *pixel* yang menjadi batas (*boundary*) antara objek dengan latar belakang dapat diketahui secara keseluruhan (algoritma *boundary following*).

2. Segmentasi ke bentuk-bentuk dasar (misalnya segmentasi huruf menjadi garis-garis vertikal dan horizontal, segmentasi objek menjadi bentuk lingkaran, elips, dan sebagainya).



Gambar 2.22 Citra Biner

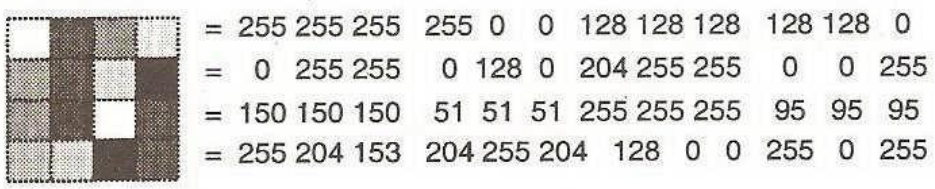
Pada penelitian ini, citra biner digunakan untuk menyimpan hasil pendeteksian objek berdasarkankan warna objeknya. Pemilihan menggunakan citra biner sebagai tempat penyimpanan data sementara untuk mempercepat

proses pendeteksian pola objek yang dilakukan berikutnya dikarenakan ukuran memori yang digunakan pada citra biner sangat kecil, yaitu 1 bit (*true* dan *false*).

2.4.3.2 Citra Warna (*True Color*)

Pada citra warna setiap titik mempunyai warna yang spesifik, yang merupakan kombinasi dari 3 warna dasar, yaitu merah, hijau, dan biru. Format citra ini sering disebut RGB (*red-green-blue*). Setiap warna dasar mempunyai intensitas sendiri dengan nilai maksimum 255 (8 bit). Setiap titik pada citra warna membutuhkan data 3 *byte*.

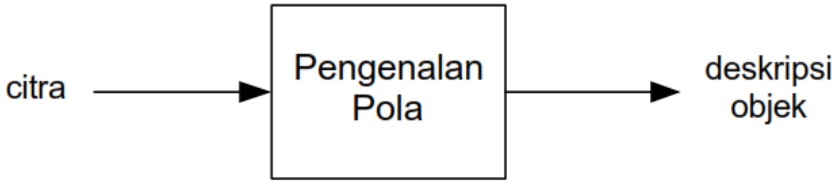
Jumlah kombinasi warna yang memungkinkan untuk format citra ini adalah 2^{24}atau lebih dari 16 juta warna, dengan demikian bisa dianggap mencakup semua warna yang ada. Inilah sebabnya format ini dinamakan *true color*.



Gambar 2.23 Citra warna *(true color)*

2.4.3.3 Pengenalan Pola

Pengenalan pola adalah mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (dalam hal ini komputer). Tujuan pengelompokan adalah untuk mengenali suatu objek di dalam citra. Manusia bisa mengenali objek yang dilihatnya karena otak manusia telah belajar mengklasifikasi objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual manusia inilah yang dicoba ditiru oleh mesin. Komputer menerima masukan berupa citra objek yang akan diidentifikasi, memproses citra tersebut, dan memberikan keluaran berupa deskripsi objek di dalam citra [5].



Gambar 2.24 Pengenalan Pola

Pola adalah entitas yang terdefinisi dan dapat didefinisikan melalui ciri-cirinya (feature). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola yang lain. Pengenalan pola bertujuan untuk menentukan kelompok untuk kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain pengenalan pola membedakan objek dengan objek lain.

Suatu sistem pengenalan pola melakukan:

1. Proses akuisisi data melalui sejumlah alat pengindraan atau sensor,
2. Mengatur bentuk representasi data,
3. Melakukan proses analisis dan klasifikasi data.

Tiga pendekatan pembuatan sistem pengenalan pola adalah Statistik (statistical), Sintaksis (syntactic) dan Jaringan Saraf Tiruan (neural network) (Schalkoff, 1992).

1. Statistik

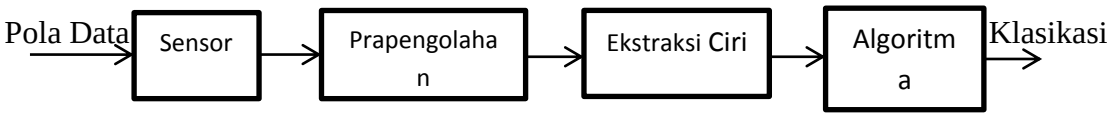
   Semakin banyak pola yang disimpan, maka sistem akan semakin cerdas. Salah satu contoh penerapannya banyak pada pola pengenalan iris scan. Kelemahannya: hanya bergantung pada data yang disimpan saja, tidak memiliki sesuatu struktur yang unik yang dapat menjadi kunci pengenalan pola.
2. Sintaksis (rule)

   Dengan rule/aturan maka sistem yang lebih terstruktur sehingga memiliki sesuatu ciri yang unik. Salah satu contoh penerapannya pada pola pengenalan sidik jari (fingerprint).

3. Jaringan Saraf Tiruan (JST)

Merupakan gabungan dari pendekatan statistik dan pendekatan sintaks. Dengan gabungan dari dua metode maka JST merupakan pengenalan pola yang lebih akurat. Salah satu contoh penerapannya pada pola pengenalan suatu citra JST merupakan suatu sistem yang dapat memproses informasi dengan meniru cara kerja jaringan saraf otak manusia.

Struktur sistem pengenalan pola ditunjukkan pada gambar 2.8. Sistem pengenalan pola ini terdiri dari suatu sensor (misalnya kamera, dan scanner), teknik prapengolahan, suatu algoritma atau mekanisme ekstraksi ciri dan algoritma untuk klasifikasi atau pengenalan (bergantung pada pendekatan yang dilakukan). Sebagai tambahan, biasanya beberapa data yang sudah diklasifikasikan diasumsikan telah tersedia untuk melatih sistem.



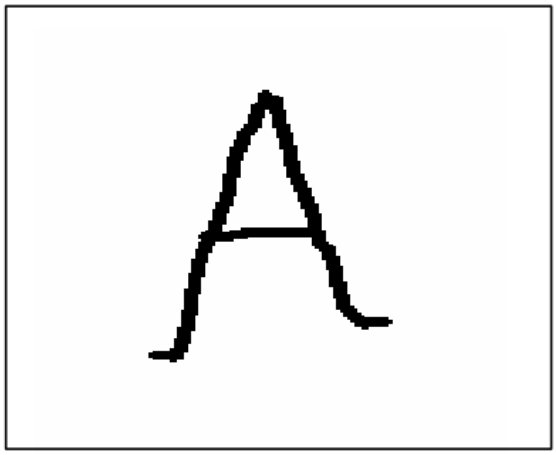
Gambar 2.25 Struktur Sistem Pengenalan Pola

Prapengolahan adalah transformasi input (masukan) data mentah untuk membantu kemampuan komputasional dan pencarian ciri serta untuk mengurangi noise (derau). Pada prapengolahan citra (sinyal) yang ditangkap oleh sensor akan dinormalisasi agar citra menjadi lebih siap untuk diolah pada tahap pemisahan ciri. Kualitas ciri yang dihasilkan pada proses pemisahan ciri sangat bergantung pada hasil prapengolahan.

Klasifikasi merupakan tahap untuk mengelompokkan input data pada satu atau beberapa kelas berdasarkan hasil pencarian beberapa ciri yang signifikan dan pemrosesan atau analisis terhadap ciri itu. Setiap kelas terdiri dari sekumpulan objek yang memiliki kedekatan (kemiripan) ciri. (Munir, 2004; Putra, 2009).

Contoh pengenalan pola misalnya citra pada Gambar 2.11 adalah tulisan tangan yang digunakan sebagai data masukan untuk mengenali karakter ‘*A*’.

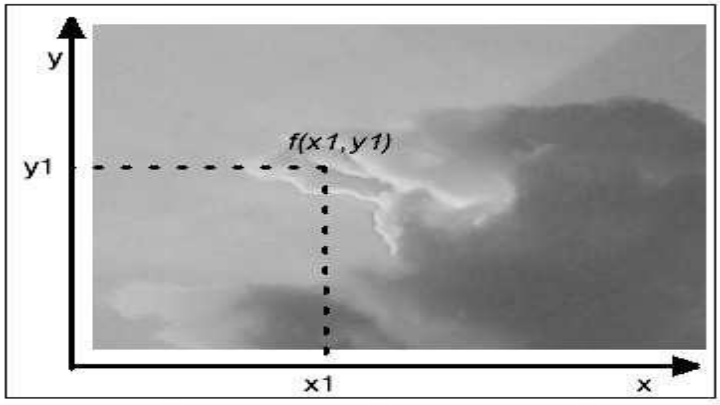
Dengan menggunakan suatu algoritma pengenalan pola, diharapkan komputer dapat mengenali bahwa karakter tersebut adalah '*A*'.



Gambar 2.26 Pola 2 Dimensi

2.5 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, f(x,y), dimana x dan y adalah koordinat spasial dan nilai f(x,y) adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada gambar 2.12.



Gambar 2.27 Citra Digital

2.6 Treshold

Parameter kunci dalam proses thresholding adalah pilihan dari nilai ambang batas. pengguna dapat secara manual memilih nilai ambang, atau algoritma thresholding dapat menghitung nilai secara otomatis, yang dikenal sebagai

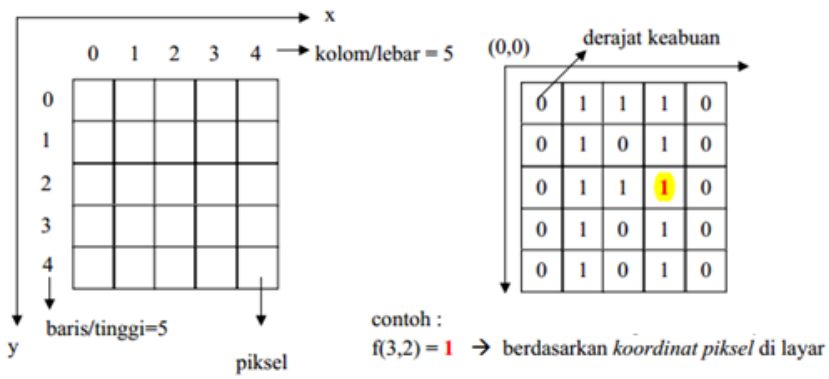
thresholding otomatis. Sebuah metode sederhana akan memilih mean atau median nilai, dasar pemikiran adalah bahwa jika *pixel* objek lebih terang dari latar belakang, juga harus lebih terang dari rata-rata. Sebuah pendekatan yang lebih canggih mungkin untuk membuat histogram dari intensitas *pixel* gambar dan menggunakan jalur lembah sebagai ambang batas. Pendekatan histogram mengasumsikan bahwa ada beberapa nilai rata-rata untuk *pixel* latar belakang dan objek, tetapi bahwa nilai *pixel* yang sebenarnya memiliki beberapa variasi di sekitar nilai rata-rata.

Secara umum proses thresholding terhadap citra grayscale bertujuan menghasilkan citra biner, secara matematis dapat ditulis seperti pada persamaan (2-4).

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

(2-4)

Dengan b(x,y) adalah piksel citra biner dari piksel citra grayscale f(x,y), dan T menyatakan nilai threshold. Nilai T ditentukan dengan menggunakan metode thresholding global dan thresholding local. Thresholding GlobalThresholding global adalah metode dengan seluruh *pixel* pada citra dikonversi menjadi hitam dan putih dengan satu nilai thresholding[12]. Contoh pemetaan piksel citra fungsi f(x,y) dapat dilihat pada gambar 2.13



Gambar 2.28 Contoh pemetaan piksel dalam citra digital

2.7 Elemen- Elemen Citra Digital

Citra digital mengandung sejumlah elemen-elemen dasar.Elemen-elemen dasar tersebut dimanipulasi dalam pengolaha citra dan dieksploitasi lebih lanjut dalam komputer vision . Elemen elemen dasar diantaranya:

2.7.1 Kecerahan (brightness)

Kecerahan adalah kata lain untuk intensitas cahaya. Sebagai mana telah dijelaskan pada bagian sampling, kecerahan pada sebuah titik (pixel) didalam citra bukanlah intensitas yang riil, tetapi sebenarnya adalah intensitas rata-rata dari suatu area yang melingkupinya. Sistem visual manusia mampu menyesuaikan dirinya dengan tingkatan kecerahan (brightness level) mulai dari yang paling rendah sampai yang paling tinggi dengan jangkauan 1010.

2.7.2 Kontras (contrast)

Kontras menyatakan sebaran terang (lightness) dan gelap (darkness) didalam sebuah gambar. Citra dengan kontras rendah dicirikan sebagai besar komposisi citranya adalah terang sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

2.7.3 Kontur (contour)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas cahaya pada pexel pixel yang bertetangga. Karena adanya perubahan intensitas inilah mata kita mampu mendeteksi tepi-tepi (edge) objek didalam citra.

2.7.4 Bentuk (Shape)

Pada umumnya citra yang dibentuk oleh mata merupakan citra dua dimensi, sedangkan objek yang diamati biasanya adalah 3 dimensi telah diproyeksikan kebidang dua dimensi dan kelihatannya sama. Misalnya, suatu ruangan terlihat berbentuk trapezium pada gambar dua dimensi. Didalam hal ini kita tahu apakah hal ini memang disebabkan oleh bentuk ruangan yang panjang ataukah memang ruangan tersebut berbentuk trapesium.

2.7.5 Warna (Color)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang (λ) yang berbeda. Warna merah mempunyai panjang gelombang yang paling tinggi, sedangkan warna ungu (violet) mempunyai panjang gelombang yang paling rendah.Warna-warna yang diterima oleh mata (sistem visual mata) merupakan hasil kombinasi cahaya dengan panjang gelombang yang berbeda.

Penelitian memperlihatkan kombinasi warna yang memberikan rentang warna yang paling lebar adalah merah (red), hijau (green), biru (blue). Penyesuaian warna pada visual kita tidak jarang dapat menimbulkan “cacat” warna (distorsi) yang dilihat. Ada dua jenis distorsi, yakni distorsi warna terhadap ruang (misal bercak abu-abu yang berada disekitar warna hijau akan berkesan ungu), dan distorsi terhadap waktu (misalnya setelah melihat warna hijau kita langsung melihat warna abu-abu, maka warna ungulah yang berkesan pada mata kita).

Pada umumnya citra yang dibentuk oleh mata merupakan citra dua dimensi, sedangkan objek yang diamati biasanya adalah 3 dimensi telah diproyeksikan kebidang dua dimensi dan kelihatannya sama. Misalnya, suatu ruangan terlihat berbentuk trapezium pada gambar dua dimensi. Didalam hal ini kita tahu apakah hal ini memang disebabkan oleh bentuk

ruangan yang panjang ataukah memang ruangan tersebut berbentuk trapesium.

2.7.6 Tekstur (texture)

Pada hakikatnya sistem visual manusia tidak menerima informasi citra secara terpisah pada setiap titik, tetapi sesuatu citra dianggapnya sebagai suatu kesatuan, jadi definisi kesamaan suatu objek perlu dinyatakan dalam bentuk kesamaan dari suatu himpunan parameter citra (brightness, color, size) atau dengan kata lain dua buah citra tidak dapat disamakan dari satu parameter saja.

2.8 Eclipse

Menurut [18] Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse:

a. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
b. Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
c. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in. Eclipse dibuat dari kerja sama antara perusahaan-perusahaan anggota 'Eclipse Foundation' (beserta individu-individu

lain). Banyak nama besar yang ikut dalam 'Eclipse Foundation', termasuk IBM, BEA, Intel, Nokia, Borland. Eclipse bersaing langsung dengan Netbeans IDE. Plugin tambahan pada Eclipse jauh lebih banyak dan bervariasi dibandingkan IDE lainnya.

1. IntelliJ IDEA (commercial, free 30-day trial)

2. Oracle JDeveloper (free)

3. Xinox JCreator (ada versi berbayar maupun free)

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta untuk pengembangannya. Semenjak itu konsursium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya. Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP).Berikut ini adalah komponen yang membentuk RCP:

1. Core platform

2. OSGi

3. SWT (Standard Widget Toolkit)

4. JFace

5. Eclipse Workbench

Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman Java. Konsep Eclipse adalah IDE yang terbuka (open), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang

spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat Diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya. Salah satu situs yang menawarkan plug-in secara gratis seperti Eclipse downloads by project.

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah Deployment dan maintenance untuk sistem enterprise, serta untuk kenyamanan.Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.

2.9 Open Source Computer Visio (Open CV)

Open Source Computer Vision adalah library open source multi-platform yang berlisensi Berkeley Software Distribution (BSD) yang bersifat open source sehingga bebas digunakan untuk hal-hal yang bersifat akademis dan penelitian. Library ini berfungsi untuk mentransformasikan data dari citra diam atau kamera video ke salah satu keputusan atau representasi baru. Semua representasi tersebut dilakukan untuk mencapai beberapa tujuan tertentu.



Gambar 2.29 OpenCV

OpenCV dikembangkan oleh Visual Interactivity Group di Intel's Microprocessor Research Lab pada tahun 1999. Tujuan dikembangkannya OpenCV adalah untuk membangun suatu komunitas open source dan menyediakan situs yang mendistribusikan upaya dari komunitas sehingga dapat dikonsolidasi dan dioptimalkan dari segi kinerjanya.

Telah banyak penambahan kelas OpenCV, sehingga OpenCV sampai saat ini telah memiliki lebih dari 500 fungsi. Berikut ini beberapa fungsi yang dimiliki oleh OpenCV :

1. Metoda Geometrik
2. Pengenalan Objek
3. Pengukuran
4. Segmentasi
5. *Utilitas*
6. Fitur
7. Transformasi Citra
8. Piramida Citra
9. *Tracking*

10. Kalibrasi kamera

11. Pengolahan citra

Alasan penggunaan OpenCV karena kostumisasi yang dimiliki oleh OpenCV yang tinggi dan kinerja yang baik pada program berbasis bahasa native C/C++. Selain itu juga terdapat beberapa kelebihan OpenCV, diantaranya :

1. OpenCV gratis, karena open source library.
2. Memliki dokumen yang cukup banyak.
3. Dapat bekerja lebih cepat pada presesor berbasis Intel.

Selain kelebihan, OpenCV juga memiliki kekurangan, yaitu OpenCV kurang mendukung bahasa berbasis .NET seperti C#, Visual Basic dan lainnya, sehingga dibutuhkan wrappers mendukungnya. Pada penelitian ini menggunakan OpenCV, karena tools yang dimiliki oleh OpenCV dapat mendukung untuk penelitian deteksi makanan dan obat menggunakan metoda String Matching ini.

2.10 UML

UML (Unified Modeling Language) adalah sekumpulan pemodelan konvensi yang digunakan untuk menetukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitannya dengan objek [9]. UML juga diartikan sebuah bahasa grafik standar yang digunakan untuk memodelkan perangkat lunak berbasis objek. UML pertama kali dikembangkan pada pertengahan tahun 1990an dengan kerjasama antara James Rumbaugh, Grady Booch dan Ivan Jacobson, yang masing- masing telah mengembangkan notasi mereka sendiri di awal Tahun 1990an. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

1. Analisis dan desain Berorientasi Objek

Pemrograman berorientasi objek bekerja dengan baik ketika bersamaan dengan *Object-oriented Analysis and Design Process (OOAD).*[8] mengatakan jika kita membuat program berorientasi objek tanpaOOAD,

ibarat membangun rumah tanpa terlebih dahulu menganalisa apa saja yang dibutukan oleh rumah itu, tanpa perencanaan, tanpa *blueprint*, tanpa menganalisis ruangan apa saja yang diperlukan, berapa besar rumah yang akan dibangun dan sebagainya.

a. Objek

Orientasi objek merupakan teknik dalam menyelesaikan masalah yang kerap muncul dalam pengembangan perangkat lunak. Teknik ini merupakan titik dalam menentukan cara yang efektif dalam pembangunan system dengan metode yang paling banyak dipaki oleh para pengembang perangkat lunak saat ini. Orientasi objek merupakan teknik pemodelan sistem riil yang berbasis objek. Intinya dari konsep ini adalah objek yang merupakan model daris sitem nyata. Menurut [13] bahwa objek adalah entitas, karakter *(behavour)* dan kadangkala disertai kondisi *(state).*

b. Kelas

Kelas adalah penggambar satu set objek yang memiliki atribut dan behavior yang sama. Kelas mirip tipe data pada program non objek, tapi lebih komprehensif karena terdapat struktur sekaligus karakteristiknya. Kita dapat membentuk kelas baru yang lebih spesifik dari kelas generalnya. Kelas dan objek merupakan jantung dari pemrograman berorientasi objek.untuk menghasilkan program ini sangat penting untuk selalu berfikirdalam konsep objek.

2. Evolusi UML

[10] menjelaskan bahwa sebelumnya ada UML, para pengembang bahasa pemrograman berorientasi objek sulit unutuk berkomunikasi satu sama lain. Kira-kira memiliki 50 jenis notasi dan grafik yang menggambarkan bahasa pemrograman berorientasi objek pada waktu itu. Pada pengguna yang berlainan ini saling berebut pengaruh agar notasi yang digunakan menjadi notasi standar. Walaupun dijumpai beberapa notasi yang cukup jelas dan sangat cocok untuk menyelesaikan problem dalam pembuatan perangkat lunak *Object Oriented Program*(OOP).

Pada bulan Oktober 1994, Jim Rumbaugh, penemu notasi *Object Modelling Technique (OMT)* dan Graddy Booch, penemu *Booch Method* (Metode Booch) bersama menyamakn notasi mereka. Di tahun yang sama, Ivar Jacobson ikut bergabung. ketiga orang itu sering disebut sebagai "*three Amigos*". Bersama membangun notasi standar OOP untuk software *Rational IBM*.

Ketiga pengembang UML tersebut telah mengalami kesulitan akibat kompleksnya permasalahan OOP yang ada, namun mereka dibantu oleh *Object Management Group (OMG)*. Artinya gabungan dari kurang lebih 800 perusahaan pengembang perangkat lunak berorientasi objek.

Maka sejak tahun 1997, divisi *Revision Task Force (RTF)* milik OMG beberapa kali merevisi UML. Revisi dimaksudkan agar memperkuat konsistensi notasi, meningkatkan kekompakan antar user dan pengembang perangkat lunak. Akan tetapi UML akhirnya mengikuti perkembangan *software-software* bebasis objek yang ada (*Java*) dari sisi pendekatan komponen dan kemampuan *tools software* tersebut. Setelah dilakukan perubahan secara sistematik, akhirnya dihasilkan UML 2.0 pada tahun 2003.

1. Komponen UML

Bagian-bagian atau Komponen utama dari UML adalah view, diagram, model element, dan *general mechanism*.

a. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. View bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis view dalam UML antara lain: use case view, logical view, component view, concurrency view,dan deployment view.

b. *Use case view*

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan external aktors. Aktor yang

berinteraksi dengan sistem dapat berupa user atau sistem lainnya. View ini digambarkan dalam use case diagrams dan kadang-kadang dengan activity diagrams. View ini digunakan terutama untuk pelanggan, perancang (designer), pengembang (developer), dan penguji sistem (tester).

c. Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (class, object,dan relationship) dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke object lain dalam suatu fungsi tertentu.View ini digambarkan dalam class diagrams untuk struktur statis dan dalam state, sequence, collaboration, dan activity diagram untuk model dinamisnya. View ini digunakan untuk perancang (designer) dan pengembang (developer).

d. Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya. View ini digambarkan dalam component view dan digunakan untuk pengembang (developer).

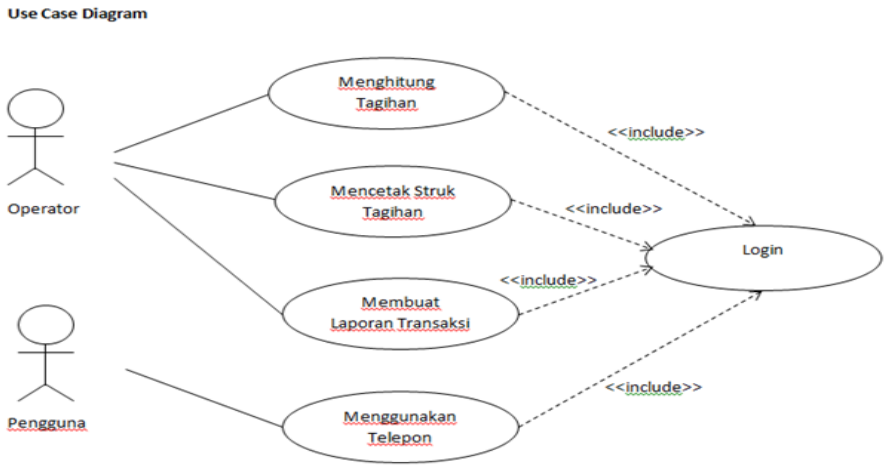
2.11 Use Case Diagram

Menurut [10] use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan [9] mengartikan use case sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomasisasi maupun secara manual, untuk tujuan bisnis tunggal. Salah satu kontributor terhadap diagram use case dalam UML adalah Ivar Jacobsen. Use case menggambarkan external view dari sistem yang akan kita buat modelnya [11] mengatakan bahwa model use case, tetapi yang perlu di ingat, diagram tidak

identik dengan model karena model lebih luas dari diagram. Komponen-komponen pembentuk use case adalah Aktor, use case dan hubungan *(link).* Pada diagram use case, relasi digambarkan sebagai sebuah garis antara symbol. Pemaknaan relasi berbeda-beda tergantung bagaiman garis tersebut digambarkan dan tipe symbol apa yang digunakan untuk menghubungkan garis tersebut. Relasi yang digunakan UML 2.0 adalah generalisasi , inklusi dan ekstensi.

2.12 Aktor

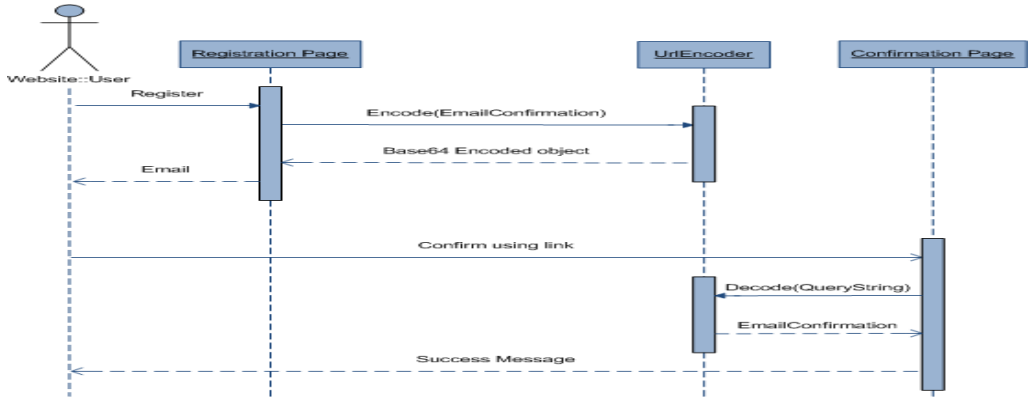
[12] Menyarankan sebelum membuat *use case* dan menentukan aktornya, agar mengidentifikasikan siapa saja yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*, langkah awal yang baik adalah mempertimbangkan kebutuhan klien dan pelanggan dalam membentuk *use case*. Setiap sistem memiliki *stakeholder* potensial yang harus dipertimbangkan karena berpengaruh terhadap kinerja sistem tersebut. Bahkan untuk beberapa jenis sistem tertentu bisa saja *hacker* masuk ke dalam kategori *stakeholder* karena berpengaruh terhadap sistem. Selama proses mencari kebutuhan sistem, hampir sebagian besar waktu dihabiskan untuk menentukan aktor yang dipilih dari *stakeholder* hasil identifikasi awal.



Gambar 2.30 Diagram Use Case

2.13 Sequence Digram

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case. Interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.



Gambar 2.31 *Sequence Diagram*

2.14 Class Diagram

Class adalah suatu diagram yang memperlihatkan atau menampilkan struktur darisebuah sistem, sistem tersebut akan menampilkan sistem kelas, atribut dan hubungan antara kelas ketika suatu sistem telah selesai membuat diagram. *Class diagram* menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Diagram kelas merupakan kumpulan kelas-kelas objek, oleh karena itu pengertian kelas sangat penting sebelum merancang diagram kelas. [4] arti dari kelas sebagai satu set objek yang memiliki atribut dan perilaku yang sama. Kelas kadang-kadang disebut juga kelas objek *(object class).* Objek yang berupa buku analisis disain dan buku pemrograman terstruktur kita kelompokan dalam satu kelas, yaitu kelas buku. Kedua objek memiliki atribut dan perilaku yang serupa. Secara teknis, [12] mengartika sebuah kelas sebagai suatu definisi sumber daya yang termasuk di dalam informasi-informasi yang menggambarkan fitur suatu entitas dan bagaimana penggunaannya. Sedangkan objek adalah entitas yang

bersifat unik yang mengikuti aturan-aturan yang sudah didefinisikan oleh kelasnya.

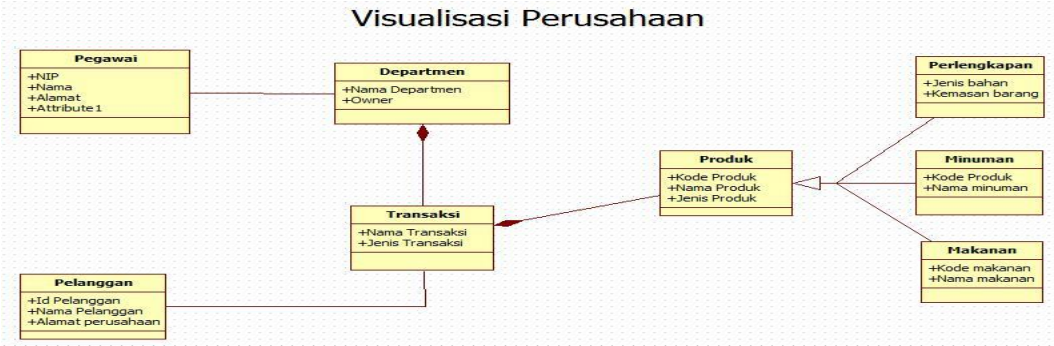
Atribut merupakan suatu kelas yang isinya adalah point-point seperti warna mobil, jumlah sisi bentuk yang disebut sebagai atribut. Atribut bisa saja sederhana (integer, floating-point) bisa juga kompleks. Atribut digambarkan dalam dua notasi yang berbeda yaitu inline atau hubungan antarkelas.

Atribut inline adalah atribut yang dapat menulis atribut kelas disebelah kanan notasi kotak. Tidak ada perbedaan semantic atau atribut inline dan atribut relasi. Keduanya bermaksud merincikan kelas yang akan dibuat.

Atribut relasi adalah atribut yang bisa mempresentasikan sebuah atribut dengan menghubungkan kelas lainnya. Untuk itu kita menggunakan notasi relasi sebagai sarana pembentuk atribut relasi dalam diagram kelas yang sangat besar. Dengan cara itu dapat dibuat atribut yang lebih detail dan kompleks.

Atribut turunan adalah atribut yang diberi notasi slash "/". Symbol ini bermaksud memberi informasi kepada inplementator bahwa atribut tersebut tidak terlalu penting.

Atribut multiplicity merupakan atribut yang menyatakan berapa banyak isi dari atribut yang terbentuk saat kelas diinisialisasi. Misalnya, kelas mobil akan memiliki 4 roda jadi multiplicity atribut roda adalah 4.



Gambar 2.32 Class Diagram

2.15 Perangkat Mobile

Hadirnya perangkat Mobile (ponsel) pada awalnya hanya sebagai sarana berkomunikasi dengan kelebihan yang bisa dibawa kemana- mana. Namun sekarang telah berubah menjadi suatu peralatan yang sangat penting. Dengan adanya internet di dalam ponsel yang telah menjadi kekuatan yang luar biasa. Sehingga dapat diibaratkan ponsel menjadi suatu magnet yang sangat penting bagi para penggunanya.bahkan di dalam ponsel saja sudah banyak fitur yang digunakan sama seperti komputer. Sehingga dengan situasi seperti ini google memanfaatkannya dengan meluncurkan ponsel berbasis android sebagai salah satu andalannya saat ini. Hingga sampai saat ini google berhasil membuat persaingan yang terjadi pada dunia mobile dari segi Sistem Operasi hingga aplikasi- aplikasi pendukungnya.

2.16 Android

Android adalah sebuah sistem operasi untuk berbagai perangkat mobile seperti handphone, netbook dan komputer tablet. Sistem operasi ni dikembangkan oleh google dengan berbasis kernel Linux versi 2.6 dan berbagai perangkat lunak dari GNU yang bersifat open source. Pada awalnya, android dikembangkan oleh perusahaan android Inc, namun kemudian perusahaan tersebut diakuisisi oleh google sehingga menjadi produk google. Sekarang ini pengembangan android ditentukan oleh sebuah konsorsium bernama *Open Handset Alliance (OHA)* yang terdiri atas berbagai vendor perangkat mobile, Komputer, dan telekomunikasi seperti Intel, Nvidia, Google, Samsung, Sprint, T-Mobile, Motorola, LG, Sony Ericsson, Toshiba, Vodafone, serta masih banyak yang lain dan anggotanya terus bertambah. Karena android bersifat Open Source dan mempunyai lisensi apache yang sangat terbuka dan bebas, maka Android menjadi sistem operasi yang sangat popular bagi berbagai produsen perangkat mobile. Berdasarkan persentasenya android mencapai 1,1 Miliar di tahun 2014. Salah satu keunggulan android adalah adanya komunitas para developer dan programmer yang luas untuk mengembangkan berbagai aplikasi yang berjalan diperangkat berbasis Android sehingga mampu memperluas fitur dan kemampuan perangkat tersebut. Saat ini sudah terdapat lebih dari 70000 aplikasi android sehingga berada di peringkat ke

dua di dunia. Android relatif mudah dan cepat karena menggunakan bahasa pemrograman Java dengan dukungan library Java yang dikembangkan oleh google.