

Penggunaan Pencocokan Graf Pada Pengolahan Citra

Fadhil Muhtadin - 13510070
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510070@std.stei.itb.ac.id

Abstrak—*Computer vision* adalah bidang yang mempelajari metode-metode untuk memproses gambar yang direkam dari dunia nyata dengan pengolahan citra agar menjadi data yang dapat diolah oleh komputer. Tujuannya adalah agar komputer dapat mengenali dan bahkan berinteraksi dengan objek yang diamati atau diobservasi. Kegunaan dari *computer vision* sangat banyak, diantaranya adalah *handwriting recognition*, *OCR (Optical Character Recognition)*, *fingerprint classification*, serta *robot vision*. Salah satu aspek yang penting dalam *computer vision* adalah pengenalan pola (*pattern recognition*). Pengenalan pola inilah yang akan menjadi basis dalam pengenalan bentuk, simbol, dan karakter pada pengolahan citra. Makalah ini akan membahas bagaimana melakukan pengenalan pola dalam pengolahan citra menggunakan teori graf, khususnya pencocokan graf.

Kata Kunci—Algoritma Ullmann, *computer vision*, graf isomorfik, pencocokan graf, pengenalan pola.

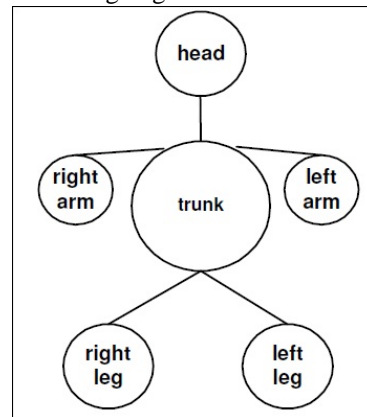
I. PENDAHULUAN

Seiring berkembangnya teknologi, maka berkembang pula kemampuan komputer. Pada awalnya komputer yang dibuat manusia hanya mampu melakukan komputasi sederhana, lama kelamaan komputer dapat melakukan operasi yang rumit dan memecahkan masalah-masalah yang tidak dapat dipecahkan manusia sebelumnya.

Lalu berkembanglah apa yang dinamakan dengan kecerdasan buatan, dimana kecerdasan yang dibuat manusia dimasukkan ke dalam mesin agar ia dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Oleh karena potensi yang dimiliki komputer, tidak heran bahwa manusia berusaha untuk mensimulasikan kemampuan yang dimiliki pada otak manusia agar dapat diimplementasikan pada komputer.

Salah satu aspek kemampuan otak manusia yang dicoba disimulasikan adalah persepsi atau indra. Pada *computer vision*, itulah yang ingin dicapai, agar komputer dapat mengenali objek yang dilihatnya seakan-akan memiliki indra penglihatan selayaknya manusia. Untuk dapat mengenali objek, pertama-tama komputer harus mengenali dahulu bentuk dari objek yang terdapat pada gambar yang diambil. Biasanya terdapat pola tertentu dari gambar yang diambil. Pola ini dapat

direpresentasikan dengan graf.



Gambar 1.1 Ilustrasi representasi bentuk tubuh manusia dengan graf

Gambaran kasar dari proses pengenalan pola ini adalah, pertama dilakukan segmentasi pada gambar sehingga gambar terbagi-bagi dalam beberapa area yang dapat direpresentasikan dengan graf, lalu graf data dari gambar dicocokkan dengan graf dari model bentuk yang ada pada database yang tersimpan dalam memori. Jika ada bentuk yang sesuai dengan graf representasi gambar, maka itulah bentuk yang ada pada gambar.

Oleh karena itu, teori tentang graf yang dipelajari pada kuliah Struktur Diskrit menjadi penting pada pembahasan ini. Teori graf memudahkan kita melakukan pencocokan pola bentuk gambar pada pengolahan citra.

II. DASAR TEORI

2.1 Dasar Teori Graf

Teori graf dalam ilmu matematika merupakan bahasan yang sudah tua namun masih memiliki banyak kegunaan sampai saat ini. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek-objek tersebut. Representasi visual graf memiliki beberapa properti yang berguna, seperti contoh: kita dapat merotasi atau mentranslasi sebuah graf dalam bidang berdimensi 2 tanpa mengubah graf tersebut, sebuah graf juga dapat memiliki atribut pada simpul dan sisinya sehingga informasi yang ditampilkan dalam graf lebih banyak. Karena itu, graf menjadi alat yang *powerful* dalam melakukan pemodelan baik dalam

bidang kimia, fisika, biologi, maupun bidang lainnya.

Definisi formal dari graf adalah sebagai berikut. Sebuah graf G adalah pasangan himpunan V dan E , dimana V adalah himpunan tidak kosong dari simpul (vertices/nodes) dan E adalah himpunan sisi (edges/arcs) yang menghubungkan sepasang simpul.

$$V = \{ v_1, v_2, v_3, \dots, v_n \}$$

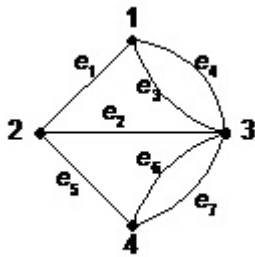
$$E = \{ e_1, e_2, e_3, \dots, e_n \}.$$

Atau secara singkat dapat ditulis $G = (V,E)$. Dari definisi di atas, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, namun harus mempunyai minimal satu buah simpul.

Simpul pada graf dapat dinomori dengan huruf, bilangan asli, atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul v_i dan v_j dinyatakan dengan pasangan (v_i, v_j) atau dengan lambang e_1, e_2, \dots . Dengan kata lain, bila e adalah sisi yang menghubungkan simpul v_i dan v_j , maka e dapat ditulis sebagai

$$e = (v_i, v_j)$$

Secara geometri graf digambarkan sebagai kumpulan noktah dalam bidang berdimensi 2 yang dihubungkan dengan sekumpulan garis. Sebagai contoh



Gambar 2.1 Contoh graf

Graf diatas adalah graf G dengan

$$V = \{ 1, 2, 3, 4 \}$$

$$E = \{ (1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4) \}$$

$$= \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7 \}$$

2.2 Terminologi Dasar Graf

Dalam materi yang akan kita bahas, ada beberapa terminologi graf yang harus dipahami terlebih dahulu.

2.2.1 Graf sederhana dan graf tak sederhana

Graf yang tidak mengandung gelang maupun sisi ganda disebut graf sederhana. Sedangkan untuk graf tak sederhana ada 2 macam, yakni graf ganda dan graf semu. Graf ganda mengandung sisi ganda, sisi ganda yang menghubungkan sepasang simpul bisa lebih dari dua. Graf semu adalah graf yang mengandung gelang, termasuk bila memiliki sisi ganda sekalipun.

2.2.2 Graf berarah dan tak berarah

Graf yang sisinya tidak memiliki orientasi arah disebut graf tak berarah. Pada graf tak berarah, urutan penulisan pasangan simpul pada sisi tidak diperhatikan. Sedangkan pada graf berarah tiap sisinya diberi orientasi arah.

2.2.3 Bertetangga (Adjacency)

Dua buah simpul pada graf tak berarah G dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain, v_i bertetangga dengan v_j jika (v_i, v_j) adalah sebuah sisi pada graf G .

2.2.4 Bersisian (Incidency)

Untuk sembarang sisi $e = (v_i, v_j)$, sisi e dikatakan bersisian dengan simpul v_i dan v_j .

2.2.5 Upagraf

Misalkan $G = (V,E)$ adalah sebuah graf. $G' = (V',E')$ adalah upagraf (subgraph) dari G jika $V' \subseteq V$ dan $E' \subseteq E$. Dengan kata lain, graf $G' \subseteq G$ atau graf G mengandung graf G' . Graf G dinyatakan sebagai *supergraph* dari graf G' . Apabila G' mengandung semua simpul dari G , maka G' dinyatakan sebagai upagraf merentang dari G .

2.2.6 Graf berbobot (Weighted Graph)

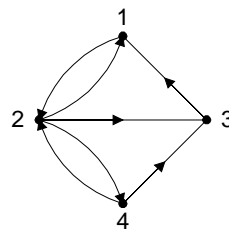
Simpul dan sisi pada sebuah graf dapat memiliki informasi atau yang disebut sebagai atribut. Informasi ini dapat berupa nama atau label, bisa juga berupa harga (bobot). Sebagai contoh, bobot pada sisi dapat menyatakan jarak antar dua buah objek dari gambar yang direkam komputer.

2.2.7 Matriks ketetangaan (Adjacency Matrix)

Matriks ketetangaan adalah bentuk representasi graf yang paling umum. Misal graf $G = (V,E)$ memiliki n simpul, $n \geq 1$. Matriks ketetangaan G adalah matriks berukuran $n \times n$. Misal matriks tersebut dinamakan $A = [a_{ij}]$. Maka

$$a_{ij} = \begin{cases} 1, & \text{jika simpul } i \text{ dan } j \text{ bertetangga} \\ 0, & \text{jika simpul } i \text{ dan } j \text{ tidak bertetangga} \end{cases}$$

Sebagai contoh



	1	2	3	4
1	0	1	0	0
2	1	0	1	1
3	1	0	0	0
4	0	1	1	0

Gambar 2.2 Graf dan matriks ketetanggaannya

2.2.8 Graf isomorfik

Dua graf disebut isomorfik bila kedua graf merupakan graf yang sama namun secara geometri berbeda, serta penamaan simpul dan sisinya juga dapat berbeda. Dua buah graf, G_1 dan G_2 dikatakan isomorfik jika terdapat korespondensi satu-satu antara simpul-simpul kedua graf dan antara sisi-sisi kedua graf sedemikian sehingga hubungan kebersisian tetap terjaga

Dari definisi isomorfik kita dapat simpulkan bahwa dua graf isomorfik memiliki tiga syarat berikut

1. Mempunyai jumlah simpul yang sama.
2. Mempunyai jumlah sisi yang sama.
3. Mempunyai jumlah simpul yang sama berderajat

tertentu.

Namun, ketiga syarat ini ternyata belum cukup untuk menjamin keisomorfikan. Pemeriksaan secara visual masih diperlukan. Nanti akan kita bahas penentuan keisomorfikan dengan algoritma Ullmann.

2.2.9 Graf Planar

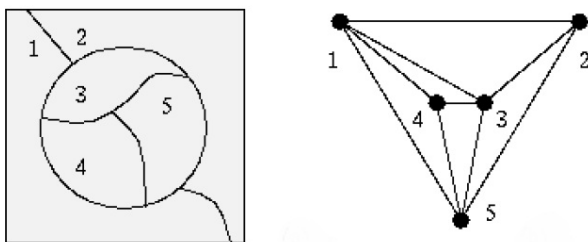
Graf planar adalah graf yang dapat digambarkan pada bidang datar dengan sisi-sisi yang tidak saling memotong. Sebuah graf yang terlihat sisi-sisinya saling berpotongan belum tentu tidak planar, graf tersebut dapat saja digambarkan kembali agar sisinya tidak berpotongan. Graf yang digambarkan dengan sisi-sisinya tidak saling berpotongan disebut juga graf bidang.

2.2.10 Graf Dual

Misal ada sebuah graf bidang G , kita dapat menggambar graf G^* yang secara geometri merupakan dual dari graf G dengan cara :

1. Pada tiap wilayah atau muka (*face*) di G , buat sebuah simpul v^* untuk G^* .

2. Untuk tiap sisi di G , tariklah sisi e^* (untuk G^*) yang memotong sisi tersebut. Sisi e^* akan menghubungkan simpul-simpul v^* pada G^* . Untuk sisi e yang salah satu simpulnya berderajat 1, maka e^* merupakan sebuah gelang. Graf G^* disebut graf dual. Aplikasi dari graf dual ini salah satunya dalam merepresentasikan area (*region*) dalam pemetaan. Pada peta, tiap wilayah atau area direpresentasikan dengan simpul, sedangkan sisi menyatakan hubungan ketetanggaan antar area. Dalam pengolahan citra, sebuah gambar dapat disegmentasi menjadi beberapa wilayah sesuai karakteristik warna dan batasan (*boundary*) lalu dibuat graf pemetaannya yang menggambarkan ketetanggaan antar area. Graf ini biasa disebut juga *Region Adjacency Graph*. Dari sinilah kemudian graf tersebut dicocokkan dengan bentuk yang ada dalam database sehingga bentuk dari objek pada gambar dapat dikenali.



Gambar 2.3 Segmentasi bentuk suatu simbol dan *Region Adjacency Graph*-nya

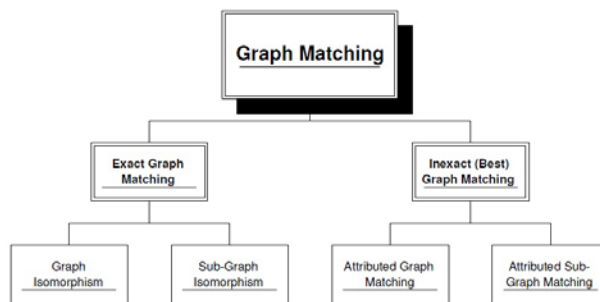
2.2 Pencocokan Graf

Seperti telah dibahas sebelumnya pencocokan graf menjadi penting dalam pengenalan pola pada pengolahan citra. Pola direpresentasikan sebagai graf atau *Region Adjacency Graph*, lalu dicocokkan dengan graf dari pemodelan bentuk (*shape*) yang ada dalam database. Graf dari database ini akan kita sebut sebagai graf model,

sedangkan graf dari gambar yang diamati akan kita sebut graf data. Pencocokan graf dilakukan dengan pencarian korespondensi antar simpul antara graf yang satu dengan yang lainnya sehingga keduanya “terlihat sama”. Graf yang dicocokkan dapat berupa graf beratribut (graf berbobot).

Secara umum, masalah pencocokan graf dapat dilakukan dengan mencari isomorfisme antara dua graf. Cara ini disebut pencocokan graf secara eksak. Dalam hal ini tiap simpul pada graf yang satu memiliki korespondensi satu-satu dengan simpul-simpul pada graf yang lainnya. Namun, terkadang objek yang diamati memiliki graf yang tidak sama dengan graf model karena segmentasi yang tidak sempurna. Bisa saja graf yang satu memiliki simpul yang lebih banyak. Dalam kasus ini kita gunakan pencocokan graf secara tidak eksak. Ini dapat dilakukan dengan mencocokkan dua graf sesuai atributnya (*attributed graph matching*), lalu kita identifikasi model mana yang paling mirip dengan struktur objek yang diamati. Teknik ini menggunakan suatu fungsi objektif yang mengukur tingkat kemiripan dari simpul-simpul dan sisi-sisi yang berkorespondensi. Fungsi ini biasa disebut *fitness function*.

Namun kita tidak akan membahas teknik tidak eksak ini. Dalam kasus dimana jumlah simpul graf model dan graf data berbeda, atau $|V_M| < |V_D|$, kita dapat berusaha untuk mencari pemetaan $f : V_D \rightarrow V_M$ sehingga $(u, v) \in E_D$ jika dan hanya jika $(f(u), f(v)) \in E_M$. Dengan kata lain mencari korespondensi graf kecil didalam graf yang lebih besar. Teknik ini disebut pencocokan upagraf atau pencarian isomorfisme upagraf.



Gambar 2.4 Klasifikasi dari Graph Matching

2.2.1 Algoritma Ullmann

Salah satu teknik yang digunakan dalam pencocokan graf adalah dengan algoritma yang dicanangkan oleh Ullmann pada tahun 1976. Algoritma yang ia ajukan berupa pencarian secara enumerasi dari kemungkinan upagraf yang ada melalui *search-tree*. Dari hasil enumerasi ini dicocokkan apakah ada upagraf yang cocok dengan graf yang dibandingkan, bila ada maka graf tersebut merupakan upagraf dari graf sumber enumerasi tersebut. Metode awal yang ada dilakukan secara *brute-force*, namun kemudian diperbaiki oleh Ullmann dengan *refinement procedure*. Berikut akan dijabarkan langkah-langkahnya.

Misal ada dua graf $G_A = (V_A, E_A)$ dan $G_B = (V_B, E_B)$ dan ingin dicari apakah G_A isomorfik dengan upagraf dari G_B . Jumlah simpul dan sisi pada masing-masing graf adalah p_A, q_A , dan p_B, q_B . Kita buat matriks ketetanggaan dari tiap graf $A = [a_{ij}]$ dan $B = [b_{ij}]$. Lalu kita definisikan matriks M' berukuran $p_A \times p_B$ yang isinya 1 dan 0 sedemikian sehingga tidak ada kolom yang berisi lebih dari satu nilai 1. Dari sini kita hasilkan matriks $C = [c_{ij}] = M'(M'B)^T$. Apabila

$$(\forall i \forall j) (a_{ij} = 1) \rightarrow (c_{ij} = 1) ; 1 \leq i \leq p_A \ \& \ 1 \leq j \leq p_B \quad (1)$$

maka M' menyatakan adanya isomorfisme antara G_A dan upagraf dari G_B . Dalam hal ini $m'_{ij} = 1$ menandakan adanya korespondensi antara simpul ke- i pada G_A dengan simpul ke- j pada G_B .

Pada awal algoritma kita definisikan matriks $M^0 = [m^0_{ij}]$ berukuran $p_A \times p_B$ dan mengisinya dengan

$$m^0_{ij} = \begin{cases} 1 & \text{bila derajat simpul ke-} j \text{ } G_B \geq \text{derajat simpul ke-} i \text{ } G_A \\ 0 & \text{selain dari itu} \end{cases}$$

Matriks M^0 ini akan digunakan untuk mengenumerasi semua kemungkinan matriks korespondensi M' yang mungkin terjadi, lalu matriks M' yang ada akan dicocokkan dengan syarat (1) diatas.

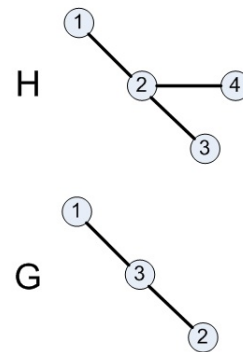
Matriks M' dibentuk dengan mengubah semua 1 menjadi 0 kecuali satu dalam tiap baris dari matriks M^0 . Bila semua langkah dan kemungkinan enumasinya direpresentasikan dalam search-tree maka tiap simpul terminal dari pohon pada kedalaman $d = p_A$ berkorespondensi dengan satu matriks M' unik. Sedangkan tiap simpul dalam dari pohon pada kedalaman $d < p_A$ berkorespondensi dengan suatu matriks M yang berbeda dengan matriks M^0 dalam hal pada sejumlah d baris dari matriks telah dilakukan perubahan nilai 1 menjadi 0 sesuai kriteria diatas.

Algoritma ini menggunakan p_B -bit vektor biner $\{F_1, F_2, \dots, F_{p_B}\}$ untuk merekam kolom mana yang digunakan dalam tahapan yang telah dilewati. $F_i = 1$ jika kolom ke- i telah dipakai. Algoritma ini juga menggunakan $\{H_1, H_2, \dots, H_{p_A}\}$ untuk menyatakan kolom telah dipilih pada kedalaman mana. $H_d = k$ bila kolom ke- k telah dipakai pada kedalaman (baris) ke- d . Berikut ini algoritma enumerasi M'

- Step 1 $M := M^0, d := 1; H_1 = 0;$
for all $i = 1, \dots, p_A$, set $F_i := 0;$
- Step 2 If there is no value of j such that $m_{dj} = 1$ and $F_j = 0$ then go to step 7;
 $M_d := M,$
If $d = 1$ then $k := H_1$ else $k := 0,$
- Step 3 $k := k + 1,$
if $m_{dk} = 0$ or $F_k = 1$ then go to step 3;
for all $j \neq k$ set $m_{dj} := 0,$
- Step 4 If $d < p_A$ then go to step 6 else use condition (1) and give output if an isomorphism is found;

- Step 5 If there is no $j > k$ such that $m_{dj} = 1$ and $F_j = 0$ then go to step 7;
 $M := M_d,$
go to step 3;
- Step 6 $H_d := k, F_k := 1; d := d + 1;$
go to step 2,
- Step 7 If $d = 1$ then terminate algorithm,
 $F_k := 0; d := d - 1, M = M_d, k := H_d,$
go to step 5

Mari kita praktekan algoritma ini dengan contoh. Misalnya kedua graf berikut ini akan kita uji dengan algoritma Ullmann.



Maka pertama kita perlu membuat matriks ketetanggaan kedua graf tersebut.

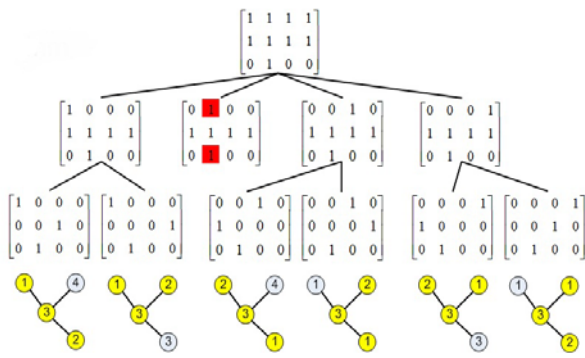
$$A_H = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$A_G = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Lalu kita bentuk matriks M^0 dengan kriteria yang sudah disebutkan.

$$M^0 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Baru dari matriks M^0 tersebut kita enumerasi semua kemungkinan upagraf dari graf H.



Gambar 2.4 Search-tree dari enumerasi kemungkinan upagraf pada algoritma Ullmann

Dari semua hasil tersebut ternyata ada satu yang isomorfis dengan graf G, yakni

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = P \cdot M \cdot (M^T A_H)^T$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^T$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

compared with $A_G = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

Dari sini dapat kita nyatakan bahwa matriks G isomorfis dengan upagraf matriks H.

Hal yang menarik untuk diperhatikan adalah bahwa algoritma Ullmann tidak hanya untuk mencari isomorfisme upagraf, namun juga isomorfisme 2 graf yang memiliki jumlah simpul yang sama.

III. IMPLEMENTASI DALAM PENGOLAHAN CITRA

3.1 Proses Pengenalan Pola

Seperti yang sudah diterangkan sebelumnya, agar sebuah komputer dapat mengenali objek yang diamatinya, perlu dilakukan serangkaian proses untuk menganalisis data gambar agar dapat dikenali pola dan bentuknya pada gambar tersebut. Rangkaian proses ini dimulai dari penangkapan gambar, hingga pencocokan.

3.1.1 Akuisisi data

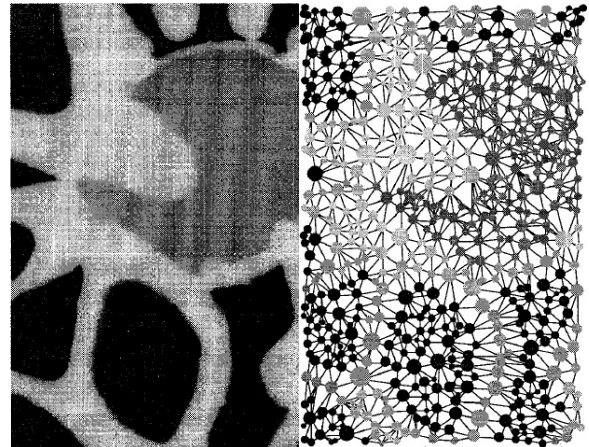
Akuisisi data merupakan proses komputer untuk menangkap dan mendapatkan informasi mengenai tampilan objek yang diamati menggunakan sensor yang tersedia. Dalam hal ini kamera menjadi perangkat yang lazim digunakan.

3.1.2 Segmentasi

Proses segmentasi merupakan proses mengekstrak

struktur geometrik dari objek agar dapat diproses secara kognitif oleh prosesor. Ada dua pendekatan dalam proses ini, berbasis sisi dan berbasis area.

Pendekatan berbasis sisi akan berupaya untuk mendeteksi diskontinuitas pada permukaan objek, sehingga tampak ada sisi yang membatasi dan membagi-bagi objek menjadi beberapa petak permukaan atau area. Sedangkan pendekatan berbasis area akan mendeteksi perbedaan sifat (*property*) dalam jangkauan pixel tertentu. Pengelompokan jangkauan pixel menjadi satu area atau petak permukaan tertentu didasarkan pada kesamaan sifat. Sifat ini dapat berupa karakteristik warna pada gambar.



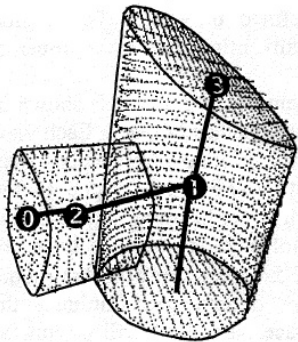
Gambar 3.1 Gambar dan segmentasinya berdasarkan warna pada tiap jangkauan pixel

3.1.3 Pencocokan graf

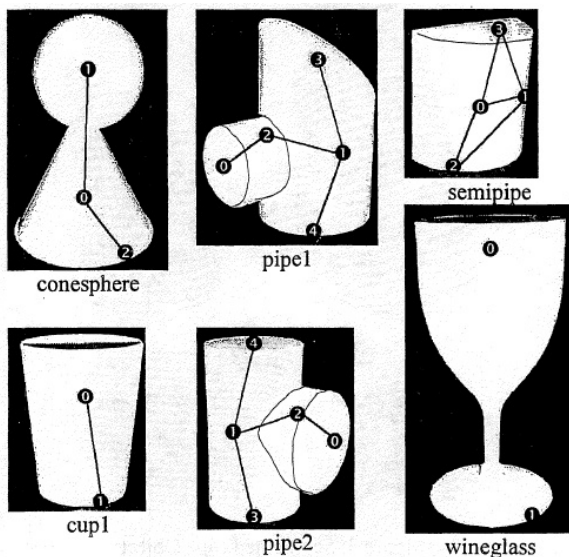
Untuk dapat mengenali bentuk objek, komputer sebelumnya telah menyimpan database berisi model dari bentuk-bentuk lazim yang akan ditemuinya selama ia melakukan pengolahan citra. Sehingga komputer dapat mengenali objek apa yang ia amati dan mengambil keputusan untuk melakukan aktivitas apa berdasarkan objek yang ditemuinya tersebut. Hal ini memang mungkin tidak efisien karena begitu banyak informasi mengenai model benda yang harus disimpan. Tapi bila dilihat dari segi kepraktisannya, penggunaan pengolahan citra ini tidak harus dilakukan secara umum, dapat dikhususkan pada beberapa bidang saja. Misalnya sebuah robot didesain agar ia dapat mengotomatisasi pekerjaan tertentu dalam suatu pabrik. Robot tersebut hanya perlu memiliki database objek yang akan ditemuinya di pabrik. Ia tidak perlu mengingat benda yang hanya ditemui di tempat lain, misalnya peralatan kedokteran atau perabotan rumah tangga.

Seperti telah dibahas sebelumnya bahwa metode yang dapat digunakan untuk pencocokan ini salah satunya adalah dengan algoritma Ullmann untuk pencocokan isomorfisme upagraf. Proses segmentasi tidak selalu berjalan sempurna, ini bisa terjadi karena keterbatasan perangkat kamera ataupun kelemahan dari proses segmentasi yang otomatis itu sendiri. Proses segmentasi

yang kurang baik dapat menimbulkan bentuk graf yang memiliki jumlah simpul yang berlebih, fenomena ini disebut *over-segmentation*. Oleh karena itu, pencocokan graf tidak bisa dilakukan semerta-merta harus dengan isomorfisme graf, namun lebih baik digunakan isomorfisme upagraf seperti pada algoritma Ullmann. Disinilah kelebihan algoritma Ullmann, algoritma ini dapat digunakan untuk isomorfisme graf dan upagraf sehingga lebih fleksibel, namun kompleksitas agoritmanya termasuk tinggi, sehingga algoritma ini bukanlah yang paling mangkus.



Gambar 3.2 Contoh graf data dari sampel yang diamati



Gambar 3.3 Contoh database pemodelan bentuk dalam graf

IV. KESIMPULAN

Computer vision atau Machine vision merupakan bidang yang menarik untuk dipelajari karena memiliki aplikasi dalam berbagai bidang. Dengan berkembangnya kemampuan persepsi/indra komputer, maka semakin banyak pula yang dapat dilakukan oleh komputer. Sehingga komputer dapat membantu manusia dalam berbagai bidang seperti otomatisasi proses manufaktur

dalam pabrik, membantu pekerjaan rumah tangga, dan lain sebagainya.

Dalam hal ini teori graf yang dipelajari dalam bidang matematika diskrit menjadi sangat penting, graf yang memiliki kemampuan ekspresif yang tinggi untuk dapat merepresentasikan bentuk atau pola menjadi esensial dalam pemodelan bentuk objek dan pengenalan pola pada pengolahan citra.

Dengan demikian sesungguhnya aplikasi matematika diskrit sangat banyak dalam kehidupan sehari-hari, mulai dari penggunaan aritmatika modulo dalam teknik enkripsi, teori graf dalam manajemen jaringan komunikasi dan pencarian jalur tercepat, hingga penggunaan pohon keputusan dalam penentuan keputusan. Jadi aplikasi matematika diskrit tidak hanya terbatas pada *computer science* saja tapi juga dalam bidang-bidang lainnya.

V. DAFTAR REFERENSI

- [1] Munir, Rinaldi, "Diktat Kuliah IF2091 Struktur Diskrit", Bandung: Penerbit Informatika, 2008.
- [2] J.R. Ullmann, "An Algorithm for Subgraph Isomorphism", England: National Physical Laboratory, 1974.
- [3] Irving Hoffmann & Ray Jarvis, "Object Recognition via Attributed Graph Matching", Australia: Department of Electrical and Computer Systems Engineering Monash University, 2000.
- [4] "The Graph Matching Problem"
URL: <http://www.sc.ehu.es/acwbecae/ikerkuntza/these/Ch2.pdf> diakses tanggal 10-12-2012 pukul 19.00
- [5] "Computer Vision"
URL : http://en.wikipedia.org/wiki/Computer_vision diakses tanggal 10-12-2012 pukul 19.10

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2011

ttd

Fadhl Muhtadin - 13510070