

Penerapan Teori Graf dalam Metode *Elastic Bunch Graph Matching* pada Sistem Pengenalan Wajah

Wilson Tandya - 13519209¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13519209@std.stei.itb.ac.id

Abstrak—Sistem pengenalan wajah atau yang lebih dikenal dengan *face recognition* sudah sangat lazim digunakan terutama pada gawai-gawai terbaru. Sistem pengenalan wajah dapat dilakukan dengan berbagai metode, pada makalah ini dibahas metode *Elastic Bunch Graph Matching (EBGM)* karena dinilai efektif untuk mengurangi kesalahan pada pengenalan wajah dari faktor pose, penerangan, dan ekspresi wajah. Pada makalah ini akan dibahas tahapan penggunaan metode *Elastic Bunch Graph Matching (EBGM)*, mulai dari pembuatan *Gabor wavelet* dari gambar wajah, pembuatan *jets* dan *bunch graphs* sampai proses mencari kesamaan dengan persamaan *similarity*.

Kata Kunci—*Elastic Bunch Graph Matching, Gabor wavelet, graf, sistem pengenalan wajah*

I. PENDAHULUAN

Zaman sekarang, gawai yang kita miliki sering kali telah dilengkapi dengan berbagai fitur canggih, salah satu yang menarik adalah pemindai biometrik. Biometrik sendiri berasal dari bahasa Yunani *bios* yang berarti hidup dan *metron* yang berarti ukur, secara harfiah biometrik dapat diartikan sebagai studi tentang karakteristik biologi yang dapat diukur. Pada kamus *Merriam-Webster*, *biometrics* diartikan sebagai pengukuran dan analisis karakteristik fisik atau perilaku yang unik terutama sebagai alat untuk memverifikasi identitas pribadi.

Bentuk dan jenis dari pemindai biometrik sendiri sangat beragam, mulai dari wajah, sidik jari, suara, iris, dan yang paling ekstrem DNA. Biometrik sendiri telah digunakan sejak era perang dunia kedua, penelitian militer Amerika menemukan penggunaan *voice recognition* pada pesawat tempur pada saat itu, selain itu metode pengenalan sidik jari *Automate Fingerprint Identification System (AFIS)* untuk menganalisis sidik jari ditemukan pada tahun 1960 oleh FBI di Amerika. Di Indonesia, penggunaan biometrik dapat terlihat pada e-KTP yang dicetak oleh pemerintah. Pada e-KTP tersebut tersimpan data biometrik sidik jari dan iris pemilik.

Penggunaan dari sistem biometrik ini tidak hanya terbatas pada bidang keamanan, perusahaan seperti *Google* telah berhasil mengembangkan *voice recognition* yang hanya menerima perintah dari pemiliknya saja. Namun, perkembangan paling pesat dari kegunaan biometrik ini ditujukan untuk sistem keamanan.

Pada saat ini, *smartphone* mulai mengimplementasikan fitur

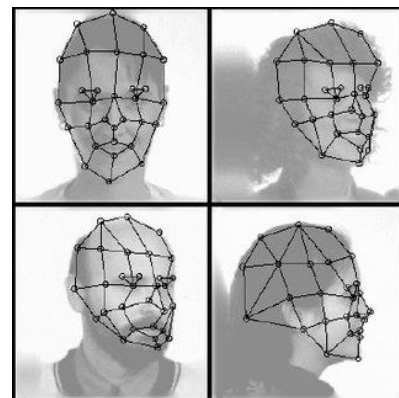
face recognition sebagai pengganti dari biometrik sidik jari. Sistem pengenalan wajah (*face recognition*) telah diaplikasikan dengan berbagai metode, antara lain, metode *Principal Component Analysis (PCA)*, metode *Independent Component Analysis (ICA)*, metode *Linear Discriminant Analysis (LDA)*, metode *Local Binary Patterns Histograms (LBPH)*, metode *Kernel*, metode *Bayesian Framework*, metode *Hidden Markov Model (HMM)*, metode *Support Vector Machine (SVM)*, metode *Elastic Bunch Graph Matching (EBGM)*, dan masih banyak lagi.

Metode *Elastic Bunch Graph Matching (EBGM)* dipilih pada pembahasan ini karena memiliki tingkat akurasi tinggi. Permasalahan yang biasa muncul pada berbagai metode *face recognition* adalah *pose, illumination, dan expression*. Hal tersebut dapat dikurangi dengan memanfaatkan metode *Elastic Bunch Graph Matching (EBGM)*.

II. LANDASAN TEORI

A. Sistem Pengenalan Wajah

Sistem pengenalan wajah dapat menentukan atau mencari kesamaan pada wajah-wajah manusia dengan menyimpan data dari hasil pemetaan wajah manusia dalam bentuk graf. Banyak metode yang diterapkan pada sistem pengenalan wajah ini, salah satunya adalah *Elastic Bunch Graph Matching (EBGM)*, dengan menggunakan metode ini, kelemahan seperti perbedaan pose, keterangan, ekspresi dapat ditangani lebih baik ketimbang metode-metode lain, diantaranya *Principal Component Analysis (PCA)*, *Support Vector Machine (SVM)*, *Hidden Markov Model (HMM)*, dan masih banyak lagi.

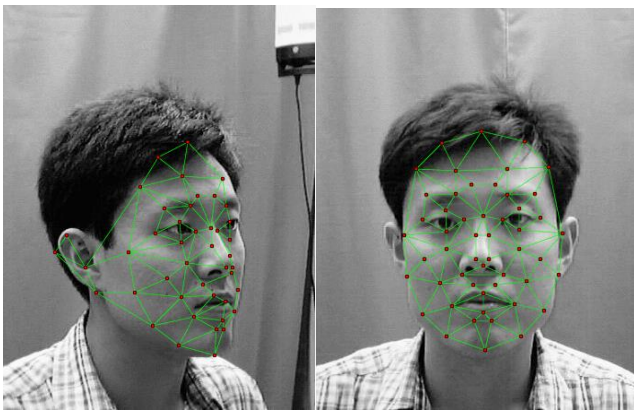


Gambar 2.1 Kisi untuk Pengenalan Wajah, diambil dari [2]

B. Elastic Bunch Graph Matching (EBGM)

Elastic Bunch Graph Matching (EBGM) merupakan algoritma komputer untuk mengenali objek atau kelas objek dalam suatu gambar berdasarkan representasi grafik yang dipetakan dari gambar lain. Algoritma ini sudah sangat sering ditemukan pada sistem pengenalan dan analisis wajah dan mengenali gerak dari objek lainnya.

Metode Elastic Bunch Graph Matching (EBGM) ini merupakan ekstensi dari Elastic Graph Matching (EGM). Metode Elastic Graph Matching (EGM) merupakan algoritma yang terinspirasi secara biologis dikarenakan cara kerjanya yang menyerupai model dari otak. Objek pada EGM akan direpresentasikan dengan graf berlabel, dengan simpul menandakan tekstur daerah lokal berdasarkan Gabor wavelets dan sisi menandakan jarak antar simpul pada suatu gambar. Dengan kumpulan simpul dan sisi tersebut akan didapat kumpulan tekstur lokal serta jarak antar tekstur tersebut pada suatu objek.



Gambar 2.2 Pemetaan Graf pada Wajah Manusia, diambil dari [3]

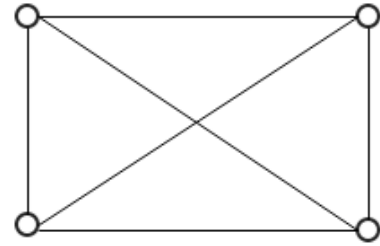
Dengan ekstensi dari metode Elastic Graph Matching (EGM), kesamaan pose, pencahayaan, dan ekspresi dapat dibedakan, karena mengambil sampel pada bagian-bagian tertentu dari objek, tidak hanya mengklasifikasi sebuah objek dengan sebuah graf namun banyak potongan graf yang dikumpulkan dan membuat objek baru ketimbang dengan hanya melihat kesamaan struktur dari objek tersebut.

C. Graf

Graf adalah himpunan dari verteks (*node*) yang terhubung oleh sisi (*edge*). Pada umumnya graf digambarkan dengan kumpulan titik untuk merepresentasikan verteks dan kumpulan garis yang menghubungkan titik untuk merepresentasikan sisi. Dalam notasi matematika sebuah graf dapat dimodelkan sebagai $G = (V, E)$, dengan V sebagai himpunan tak kosong dan berhingga dari simpul simpul $\{v_1, v_2, \dots, v_n\}$ dan E sebagai himpunan sisi yang menghubungkan sepasang simpul $\{e_1, e_2, \dots, e_n\}$

Graf memiliki beragam jenis, berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, terdapat :

a. Graf Sederhana (Simple Graph)

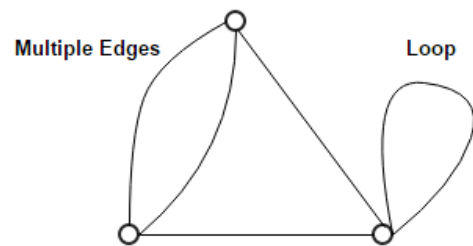


Simple Graph

Gambar 2.3 Graf Sederhana, diambil dari [4]

Graf sederhana adalah graf yang tidak memiliki gelang maupun sisi ganda.

b. Graf Tidak Sederhana (Unsimple Graph)



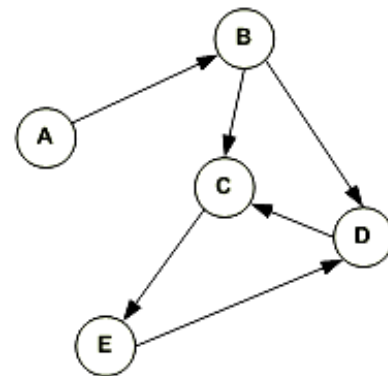
Not a Simple Graph

Gambar 2.4 Graf Tidak Sederhana, diambil dari [4]

Graf Tidak Sederhana adalah graf yang memiliki gelang atau sisi ganda.

Berdasarkan orientasi arah pada sisi, secara umum graf dibedakan menjadi 2 jenis:

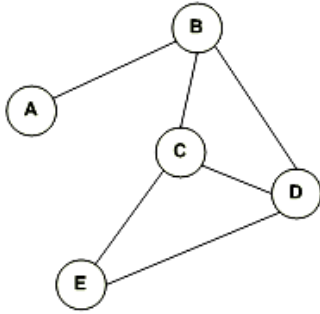
c. Graf Berarah (Directed Graph)



Gambar 2.5 Graf Berarah, diambil dari [4]

Graf Berarah adalah graf yang sisinya memiliki orientasi arah, baik keluar maupun masuk simpul.

d. Graf Tidak Berarah (*Undirected Graph*)

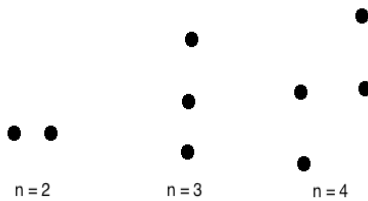


Gambar 2.6 Graf Tidak Berarah, diambil dari [4]

Graf Tidak Berarah adalah graf yang sisinya tidak memiliki orientasi arah.

Beberapa jenis graf yang tidak dikategorikan diatas lainnya antara lain:

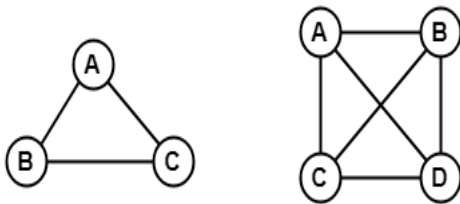
e. Graf Kosong (*Null Graph*)



Gambar 2.7 Graf Kosong, diambil dari [4]

Graf kosong adalah graf yang tidak memiliki sisi antar simpulnya, biasa ditulis dengan notasi N_n .

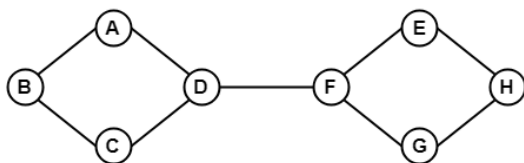
f. Graf Lengkap (*Complete Graph*)



Gambar 2.8 Graf Lengkap, diambil dari [4]

Graf lengkap adalah graf sederhana yang setiap simpulnya memiliki sisi ke semua simpul lainnya. Graf ini biasa ditulis dengan notasi K_n . Graf lengkap dengan n simpul memiliki $\frac{n(n-1)}{2}$ sisi.

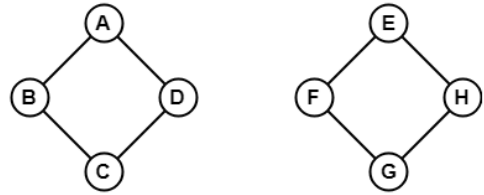
g. Graf Terhubung (*Connected Graph*)



Gambar 2.9 Graf Terhubung, diambil dari [4]

Graf terhubung adalah graf yang seluruh simpulnya terhubung baik secara langsung maupun tidak langsung ke seluruh simpul lainnya.

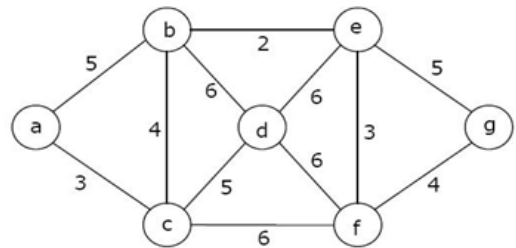
h. Graf Tidak Terhubung (*Disconnected Graph*)



Gambar 2.10 Graf Tidak Terhubung, diambil dari [4]

Graf tidak terhubung adalah graf yang tidak seluruh simpulnya memiliki sisi yang secara langsung maupun tidak langsung menghubungkan ke seluruh simpul lainnya.

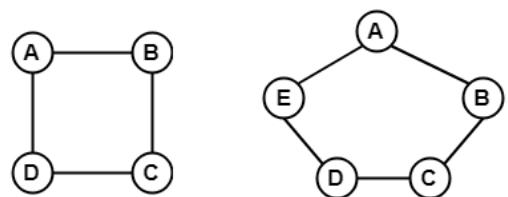
i. Graf Berbobot (*Weighted Graph*)



Gambar 2.11 Graf Berbobot, diambil dari [4]

Graf berbobot adalah graf yang setiap sisinya diberi sebuah nilai atau bobot.

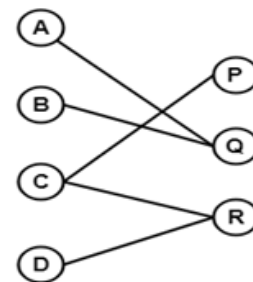
j. Graf Lingkaran (*Cyclic Graph*)



Gambar 2.12 Graf Lingkaran, diambil dari [4]

Graf lingkaran adalah graf yang setiap simpulnya memiliki derajat dua, biasa ditulis dengan notasi C_n .

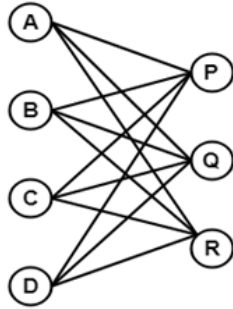
k. Graf Bipartite (*Bipartite Graph*)



Gambar 2.13 Graf Bipartite, diambil dari [4]

Graf bipartite adalah graf yang kumpulan simpulnya dapat dipisah menjadi dua bagian. Setiap sisi pada graf hanya dapat menghubungkan simpul pada dua bagian simpul yang berbeda.

1. Graf Bipartite Lengkap (*Complete Bipartite Graph*)



Gambar 2.14 Graf Bipartite Lengkap, diambil dari [4]

Graf bipartite lengkap merupakan graf bipartite dengan syarat seluruh bagian dari simpul satu terhubung tepat satu kali dengan seluruh simpul bagian dua

D. Terminologi Graf

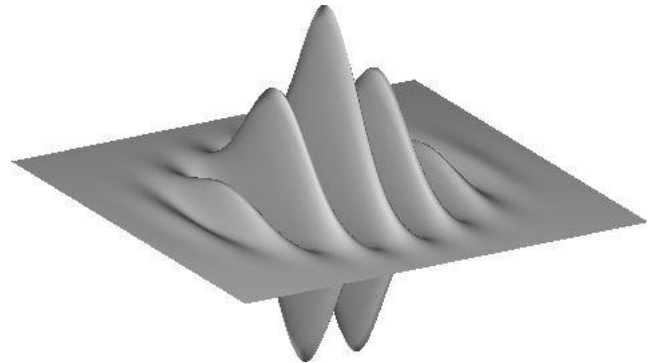
- a. **Ketetanggaan (*Adjacent*)**
Dua simpul bertetangga bila keduanya terhubung oleh sebuah sisi.
- b. **Bersisian (*Incidency*)**
Pada sisi $e = (v_1, v_2)$, dapat dinyatakan bahwa e bersisian dengan simpul v_1 dan bersisian dengan simpul v_2 .
- c. **Simpul Terpencil (*Isolated Vertex*)**
Simpul terpencil adalah simpul yang tidak memiliki sisi yang bersisian dengannya.
- d. **Derajat (*Degree*)**
Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut, dituliskan dengan notasi $d(v)$.
- e. **Lintasan (*Path*)**
Lintasan adalah jejak tanpa simpul berulang.
- f. **Sirkuit (*Circuit*)**
Sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama.
- g. **Keterhubungan (*Connectivity*)**
Dua simpul v_1 dan v_2 dapat dikatakan terhubung bila terdapat lintasan dari simpul v_1 ke simpul v_2 .
- h. **Upagraf (*Subgraph*)**
 $G_1 = (V_1, E_1)$ dikatakan upagraf dari $G = (V, E)$ jika $V_1 \subseteq V$ dan $E_1 \subseteq E$.
- i. **Upagraf Merentang (*Spanning Subgraph*)**
Sebuah upagraf dikatakan upagraf merentang bila seluruh simpul pada graf utama terdapat pada graf upagraf.
- j. **Cut-set**
Cut-set dari graf terhubung G adalah himpunan sisi yang bila dihilangkan dari G menyebabkan G tidak terhubung, atau menghasilkan dua komponen.

III. METODE *ELASTIC BUNCH GRAPH MATCHING*

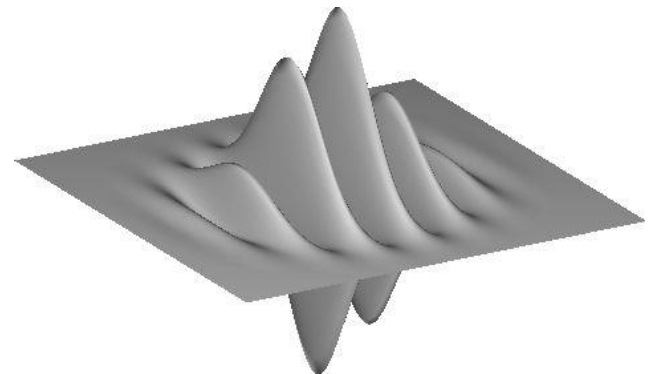
A. *Gabor Wavelets*

Fitur paling dasar yang merupakan dasar dari metode *Elastic Bunch Graph Matching (EBGM)* adalah *Gabor wavelets*. *Gabor wavelets* memiliki bentuk grafik kosinus atau sinus yang dikali dengan fungsi *Gaussian envelope*, dituliskan dengan persamaan:

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2} \exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right) \left[\exp(i\vec{k}_j \vec{x}) - \exp\left(-\frac{\sigma^2}{2}\right) \right]$$



Gambar 3.1 *Cosine Gabor Wavelet (real part of the complex wavelet)*, diambil dari [5]



Gambar 3.2 *Sine Gabor Wavelet (imaginary part of the complex wavelet)*, diambil dari [6]

Pada persamaan *Gaussian envelope* tersebut, bagian eksponensial pertama menentukan lebar, bagian eksponensial kedua menggabungkan gelombang kosinus pada kuadran riil dan gelombang sinus pada kuadran imajiner untuk menentukan orientasi dan frekuensi dari gelombang, dan bagian eksponensial terakhir merupakan komponen koreksi agar lipatan memiliki mean nol. Bagian k^2/σ^2 menormalkan lipatan.

Pada umumnya, fungsi tersebut didefinisikan secara kontigu, namun pada metode *Elastic Bunch Graph Matching (EBGM)*, \vec{x} dibuat secara diskrit mengikuti resolusi gambar serta \vec{k} juga dibuat secara diskrit dengan persamaan:

$$\vec{k}_{m,l} = k_{\max} \alpha^{-m} \begin{pmatrix} \cos\left(\frac{\pi l}{L}\right) \\ \sin\left(\frac{\pi l}{L}\right) \end{pmatrix} \quad m = 0, \dots, M-1, \quad l = 0, \dots, L-1.$$

Parameter yang sering digunakan adalah $L = 8, M = 5, \alpha = \sqrt{2}$

B. Transformasi Gabor Wavelet

Pada sebuah gambar atau objek, sebuah *Gabor wavelet* bertanggung jawab pada satu titik atau lokasi tertentu saja, sedangkan bila ingin ditemukan kesamaan yang optimal, perlu banyak titik, lebih banyak lebih baik. Oleh karena itu, digunakan operasi konvolusi (*spatial filter*) pada sebuah gambar yang akan menghasilkan respon pada seluruh lokasi, dirumuskan dalam:

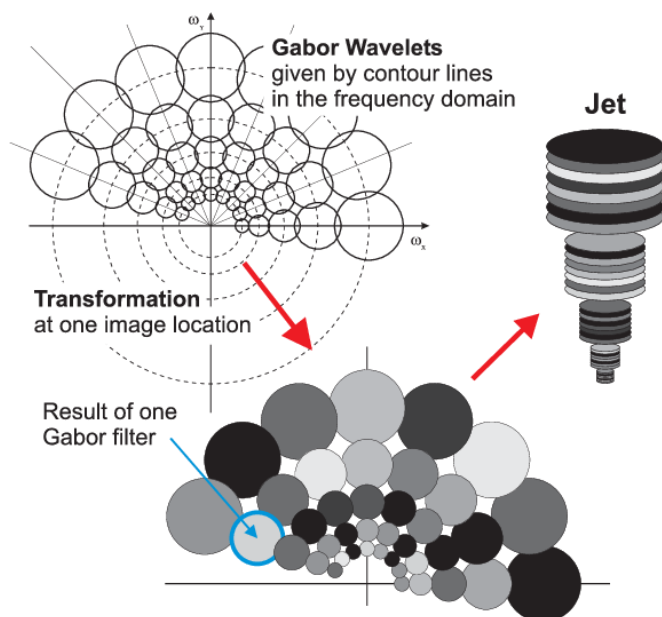
$$WI(\vec{x}_0, \vec{k}) = (\psi_{\vec{k}} * I)(\vec{x}_0) = \int \psi_{\vec{k}}(\vec{x}_0 - \vec{x}) I(\vec{x}) d^2x.$$

Untuk perhitungan dari operasi konvolusi tersebut, digunakan Transformasi *Fourier* yang akan membuat perhitungan lebih mudah. Transformasi *Fourier* dari *Gabor wavelets* dapat dihitung dengan persamaan:

$$(\mathcal{F}\psi_{\vec{k}})(\vec{\omega}) = \exp\left(-\frac{\sigma^2(\vec{\omega} - \vec{k})^2}{2k^2}\right) - \exp\left(-\frac{\sigma^2(\vec{\omega}^2 + \vec{k}^2)}{2k^2}\right)$$

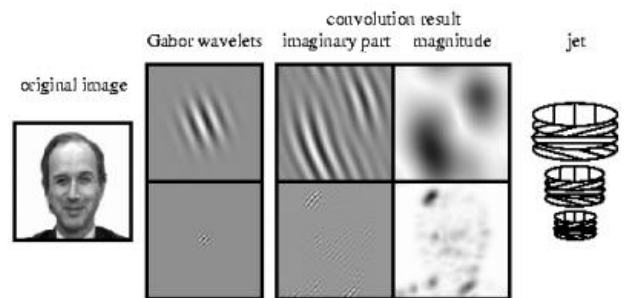
C. Jets

Pada objek maupun gambar yang melalui transformasi *Gabor Wavelet*, akan dihasilkan nilai untuk setiap *wavelet* pada seluruh lokasi objek atau gambar tersebut. Dengan parameter yang telah didiskritkan pada pembahasan bagian B, akan terbentuk 80 nilai yang terdiri atas 40 nilai riil dan 40 nilai imajiner pada seluruh posisi *pixel*. Kumpulan nilai ini pada sebuah *pixel* disebut dengan *jet*.



Gambar 3.3 Pembuatan *Jet* Melalui Transformasi *Gabor Wavelet*, diambil dari [8]

Dari *jet* yang berisi nilai dengan frekuensi dan orientasi yang berbeda-beda, *jet* dapat menjadi representasi dari tekstur lokal yang pada awal pembahasan sudah disinggung, digunakan pada metode EGM dan ekstensinya, metode EBGM.



Gambar 3.4 Tahapan Pengubahan Gambar Menjadi *Gabor Wavelet* Dilanjutkan dengan Pengubahan dalam Bentuk *Jets*, diambil dari [2]

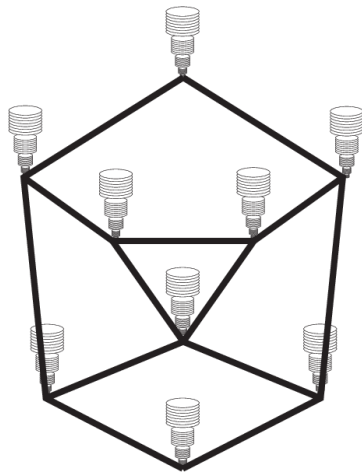
Terdapat pasangan kosinus yang melambangkan bagian riil dan sinus yang melambangkan bagian imajiner pada *Gabor wavelet*. Karena setiap filter masing-masing (riil dan imajiner) sensitif terhadap pergerakan kecil maka dari 80 nilai yang didapat pada proses transformasi, akan dilakukan penjumlahan kuadrat dari pasangan kosinus dan sinus untuk mendapatkan 40 nilai yang tetap menggambarkan frekuensi dan orientasi dari suatu *pixel* namun tidak sensitif terhadap pergeseran kecil, hal ini berkaitan dengan $\cos^2x + \sin^2x = 1$. Secara rinci, pada persamaan *jet* terdapat amplitudo dan fase yang menghasilkan vektor amplitudo dan vektor fase. Dengan hanya mengambil bagian dari amplitudo pada *jet* akan lebih menguntungkan untuk sistem *face recognition* dikarenakan hasil tidak akan sensitif terhadap sedikit pergerakan atau perpindahan. Namun, dengan hanya mengambil amplitudo dari *jet* saja, tidak dapat dilakukan rekonstruksi tekstur lokal. Namun, karena tujuan utama pembentukan *jet* pada masalah ini adalah untuk keperluan sistem pengenalan wajah, maka dengan mengambil data vektor amplitudo *jet* akan mempercepat pengolahan data.

IV. APLIKASI GRAF PADA METODE *ELASTIC BUNCH GRAPH MATCHING*

A. Pembentukan Graf

Graf diperlukan pada metode *Elastic Bunch Graph Matching* (*EBGM*) karena sebuah data *jet* hanya mengandung nilai dari sebuah titik atau lokasi saja, namun pada sebuah gambar atau objek yang akan dicari kesamaannya tentu terdapat sangat banyak titik. Karena itu, graf digunakan untuk menghubungkan nilai-nilai dari *jet* yang telah ditentukan nilainya.

Tentu dapat dilakukan pengambilan nilai *jet* pada setiap *pixel* di gambar, namun hal itu akan sangat redundan dan tidak efisien. Dapat ditentukan titik-titik menonjol atau titik-titik penting saja saat pengambilan nilai dari *jet* dan disimpan pada simpul-simpul graf, dengan sisinya merupakan jarak antar titik yang nilai *jet* nya diambil seperti pada Gambar 4.1.



Gambar 4.1 Graf Model Terdiri dari *Jets* pada Simpul, diambil dari [9]

Dari graf G^I dan G^M yang telah dibentuk, dapat diukur kesamaan untuk kepentingan pengenalan wajah melalui persamaan *similarity*, yang akan memberikan nilai lebih besar bila semakin mirip dua graf yang di bandingkan, dan nilai yang lebih rendah semakin berbeda dua graf yang di bandingkan.

$$S_v(J^I, J^M) := \frac{|J^I| \cdot |J^M|}{\|J^I\| \|J^M\|}$$

Pada persamaan *similarity* ini, untuk *jet* informasi dari fase diabaikan dan hanya nilai mutlak dari *wavelet* yang digunakan, hal ini akan membuat kesamaan lebih baik karena variasi kecil tidak akan merubah nilai kesamaanya secara drastis. Dalam praktiknya persamaan ini tidak digunakan begitu saja, namun dimodifikasi sesuai kebutuhan objek yang ingin dicari kesamaannya untuk mengoptimalkan pengenalan.

$$S_e(\vec{\Delta}_{ij}^{-I}, \vec{\Delta}_{ij}^{-M}) := - \frac{\|\vec{\Delta}_{ij}^{-I} - \vec{\Delta}_{ij}^{-M}\|^2}{\|\vec{\Delta}_{ij}^{-M}\|^2}$$

Pada persamaan *similarity* kedua ini, menggunakan perbedaan jarak antar *node*.

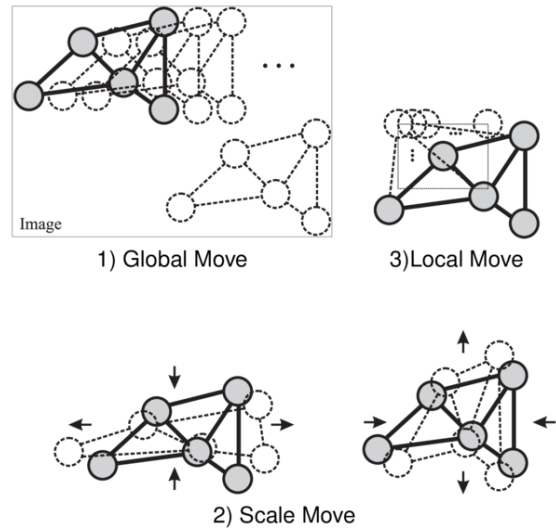
$$S_G(G^I, G^M) := \sum_{i \in V} S_v(J_i^I, J_i^M) + \lambda \sum_{(i,j) \in E} S_e(\vec{\Delta}_{ij}^{-I}, \vec{\Delta}_{ij}^{-M})$$

Pada persamaan *similarity* ketiga ini, digabungkan kesamaan *jet* dan kesamaan jarak antar *node* dengan λ sebagai konstanta untuk membobotkan penekanan persamaan *similarity*. Bagian pertama persamaan ini disebut sebagai *texture term* dan bagian akhir dari persamaan ini disebut *geometry term*. Kesamaan antar dua graf dapat dicapai bila tekstur dan geometri sama.

B. Elastic Graph Matching

Graf yang sudah dibentuk dengan simpul-simpul *jets* harus

dicocokkan pada gambar dengan melakukan *elastic graph matching*. Hal ini diperlukan, karena pada sistem *face recognition* walau wajah sama, belum tentu hasil graf yang dibentuk sama karena berbagai faktor. Terdapat tiga gerakan pada proses *elastic graph matching* untuk dapat menentukan kesamaan dari sebuah graf dengan sebuah gambar, antara lain:



Gambar 4.2 Elastic Graph Matching, diambil dari [10]

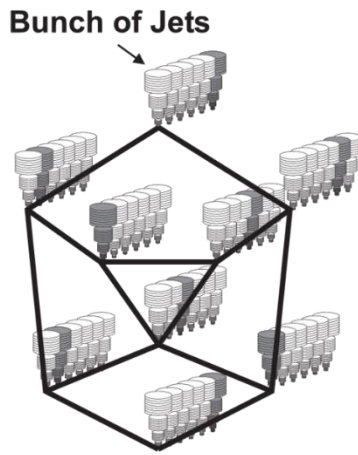
Pertama, *global move*, gerakan ini memindahkan seluruh graf (jarak antar *node* tidak diubah) untuk mendapatkan lokasi terbaik dengan *matching* terbesar. Biasanya, gerakan ini dilakukan dengan bantuan *grid* untuk melihat kesamaan pada potongan gambar dengan graf yang diberikan.

Kedua, *scale move*, gerakan ini mengubah besar dan rasio gambar. Graf dapat dibesarkan secara horisontal dan vertikal dan diubah rasio secara keseluruhan.

Ketiga, *local move*, gerakan ini memindahkan simpul-simpul dalam jangkauan lokal untuk menghadapi masalah distorsi lokal maupun perbedaan minim. Hal ini dilakukan secara random pada seluruh *node*, dan dilakukan berulang-kali sampai mendapatkan posisi optimal.

C. Bunch Graphs

Elastic graph matching yang dibahas pada bagian sebelumnya hanya menyelesaikan permasalahan untuk menemukan objek yang diketahui pada gambar dan dipasangkan atau dicocokkan dengan graf. Dengan *bunch graph* dapat diselesaikan permasalahan menemukan objek yang tidak diketahui dalam sebuah gambar dan membuat graf untuk objek tersebut. Dengan hal ini, *bunch graph* dapat membuat graf secara sendirinya bila ada gambar baru yang ingin dicari kesamaannya, dan *bunch graph* akan mempercepat komputasi karena tidak perlu mencocokkan seluruh *database* gambar yang dimiliki untuk mencari graf yang sesuai.



Gambar 4.3 Skema *Bunch Graphs*, diambil dari [11]

Pada pembuatan graf, harus ditentukan letak menaruh simpul-simpul kedalam graf dan mengeliminasi latar (*background*). Namun, pada saat pemrosesan wajah untuk *face recognition*, sudah diketahui struktur umum dari bentuk wajah manusia, dari letak mata, hidung, mulut, dan lainnya. Pengetahuan tentang struktur benda secara umum ini dapat digambarkan dengan *bunch graphs*.

Cara kerja dari *bunch graphs* ini adalah, dari data *jets* yang banyak dari berbagai objek dengan struktur yang serupa dikumpulkan pada satu simpul dan sisi pada *bunch graphs* melambangkan jarak rata-rata dari antar *jets*.

$$S_{BG}(G^I, G^B) := \frac{1}{|V|} \sum_{i \in V} \max_{m \in G} S_v(J_i^I, J_{i,m}^B) + \frac{\lambda}{|E|} \sum_{(i,j) \in E} S_e(\Delta_{ij}^I, \Delta_{ij}^B)$$

Kumpulan *jets* yang terdapat pada *bunch graphs* ini di bandingkan dengan objek yang ingin dicari dan akan ditemukan *node - node* yang bersesuaian. Dalam mencari *node* yang bersesuaian tersebut digunakan persamaan yang tertera di atas.

V. TAHAPAN DALAM SISTEM PENGENALAN WAJAH MENGGUNAKAN METODE EBGM

Dalam proses *face recognition* menggunakan metode *Elastic Bunch Graph Matching (EBGM)*, terdapat enam tahapan, yakni:

Pertama, membuat graf wajah, pada tahapan pertama ini kita harus menentukan secara manual lokasi dari simpul wajah, seperti ujung mata atau mulut, mata, ujung hidung, dan titik lainnya. Hal ini dilakukan karena ini merupakan gambar pertama yang akan diubah menjadi graf.

Kedua, membuat *face bunch graph*, pada tahapan pertama (membuat graf wajah) dapat dipandang sebagai *face bunch graph* dengan satu komponen. Oleh karena itu, tahap pertama diulang terus menerus dan disatukan satu sama lain. Dalam proses ini akan dilakukan pula penggeseran simpul secara manual agar didapat titik-titik yang lebih general.

Ketiga, membuat galeri model dari graf, karena telah dibuat *face bunch graph* sebagai dasar, kita dapat memindai sisa dari *database* foto yang dimiliki untuk dibentuk grafnya pula didasari *face bunch graph* pada tahapan kedua.

Keempat, membuat *probe graph*, saat diberi wajah untuk diselidiki, maka dari gambar tersebut akan dibentuk sebuah

probe graph yang nantinya digunakan untuk perbandingan dengan *face bunch graph* yang telah dibuat.

Kelima, perbandingan dengan seluruh graf model, graf dari gambar di bandingkan dengan semua model dari graf yang telah dibuat, dan akan dihasilkan n banyak nilai kesamaan melalui Persamaan berikut, yang telah dibahas pada Bab 4 pula.

$$S_G(G^I, G^M) := \sum_{i \in V} S_v(J_i^I, J_i^M) + \lambda \sum_{(i,j) \in E} S_e(\Delta_{ij}^I, \Delta_{ij}^M)$$

Terakhir, tahap pengenalan, secara intuisi bila kesamaan dengan nilai tertinggi pada model dari graf, maka graf gambar tersebut sama dengan wajah yang dicari. Namun, bila nilai tertinggi dari graf model relatif rendah, dapat ditentukan bahwa wajah yang dicari tidak ditemukan sama sekali. Bila nilai tertinggi pada model dari graf terdapat beberapa buah, maka kemungkinan nilai kesamaan tidak terlalu tinggi sehingga ada beberapa gambar dari *database* yang cocok dengan gambar yang dicari. Sistem hanya dapat menjamin kesamaan suatu wajah, bila kesamaan pada satu gambar di model graf tinggi dan sisanya rendah.

Sistem *face recognition* ini bekerja 98% secara benar pada percobaan galeri dengan 250 foto tampak depan. Selain *face recognition* pengenalan wajah dengan metode *Elastic Bunch Graph Matching (EBGM)* ini dapat digunakan untuk menganalisis wajah seperti ekspresi wajah yang berbeda-beda serta analisis di bidang medis. Tahapan yang sama juga dapat dilakukan dengan objek lain hanya dengan pengubahan struktur dari graf.

VI. KESIMPULAN

Kemajuan teknologi zaman sekarang telah membuat *face recognition* sebagai hal yang biasa. Seiring berkembangnya teknologi dasar dari suatu sistem seringkali terlupakan. Proses sistem pengenalan wajah (*face recognition*) ini memiliki banyak metode, salah satunya adalah metode *Elastic Bunch Graph Matching (EBGM)* yang dinilai lebih akurat dikarenakan dapat menangani kasus-kasus terkait *pose*, *illumination*, dan *expression* yang kadang membuat pencarian kesamaan pada wajah tidak akurat.

Dengan memanfaatkan teori dasar graf, proses pada metode *Elastic Bunch Graph Matching (EBGM)* meliputi 6 langkah, pertama, pembuatan graf wajah, diikuti dengan, pembuatan *face bunch graph*, pembuatan galeri model dari graf, pembuatan *probe graph*, perbandingan dengan seluruh graf model, dan terakhir pengenalan graf.

VII. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa dan kepada Dr. Ir. Rinaldi Munir, MT. sebagai dosen pengajar mata kuliah IF2120 Matematika Diskrit K-01. Penulis juga mengucapkan terima kasih kepada teman-teman dan rekan-rekan lain yang telah mendukung saya dalam keberjalanan menulis makalah ini.

REFERENSI

- [1] Jaiswal, S., Bhadauria, S.S., & Jadon, R.S. (2011). COMPARISON BETWEEN FACE RECOGNITION ALGORITHM-EIGENFACES, FISHERFACES AND ELASTIC BUNCH GRAPH MATCHING. *Journal of Global Research in Computer Sciences*, 2, 187-193.
- [2] L. Wiskott, J. -. Fellous, N. Kruger and C. von der Malsburg. "Face recognition by elastic bunch graph matching," *Proceedings of International Conference on Image Processing*, Santa Barbara, CA, 1997, pp. 129-132 vol.1.
- [3] Laurenz Wiskott et al. (2014) Elastic Bunch Graph Matching. *Scholarpedia*, 9(3):10587.
- [4] <https://www.javatpoint.com/graph-theory-types-of-graphs>, diakses pada 5 Desember 2020.
- [5] <http://www.scholarpedia.org/w/images/8/8c/SpaceKernelRealPart.jpg>, diakses pada 5 Desember 2020.
- [6] <http://www.scholarpedia.org/w/images/5/54/SpaceKernelImaginaryPart.jpg>, diakses pada 5 Desember 2020.
- [7] Luo, Bing & Hao, Yu-Jie & Zhang, Wei-Hua & Liu, Zhi-Shen. (2008). Comparison of PCA and ICA in Face Recognition. 10.1109/ICACIA.2008.4770014.
- [8] <http://www.scholarpedia.org/article/File:CreationOfJetsFigure.png>, diakses pada 7 Desember 2020.
- [9] <http://www.scholarpedia.org/article/File:ModelGraphPure.png>, diakses pada 7 Desember 2020.
- [10] <http://www.scholarpedia.org/article/File:MovesEGM.png>, diakses pada 7 Desember 2020.
- [11] <http://www.scholarpedia.org/article/File:BunchGraph.png>, diakses pada 7 Desember 2020.
- [12] <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/matdis.htm>, diakses pada 7 Desember 2020.
- [13] Rajput N., Jain P., Shrivastava S. (2012) Face Detection Using HMM – SVM Method. In: Wyld D., Zizka J., Nagamalai D. (eds) *Advances in Computer Science, Engineering & Applications. Advances in Intelligent Systems and Computing*, vol 167. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30111-7_80
- [14] Wiskott, Laurenz & Fellous, Jean-Marc & Krüger, Norbert & von der Malsburg, Christoph. (1999). Face recognition by elastic bunch graph matching. 10.1007/3-540-63460-6_150.
- [15] <https://ai.stanford.edu/~nilsson/QAI/qai.pdf>, diakses pada 7 Desember 2020.
- [16] <https://www.sciencedirect.com/topics/engineering/gabor-wavelet>, diakses pada 7 Desember 2020.
- [17] Trnovszky, Tibor & Kamencay, Patrik & Benco, Miroslav & Hudec, Robert & Radilova, Martina & Sykora, Peter. (2014). Comparison of PCA, LDA and LBPH for Face Recognition on the AT&T Data Set.
- [18] Sarsoh J. T., Hashem K. M., Al-Hadi daykh D. (2012). Classifying of human face images based on the graph theory concepts. *Glob. J. Comput. Sci. Technol.* 12 23–27.
- [19] M. Kafai, L. An and B. Bhanu, "Reference Face Graph for Face Recognition," in *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2132-2143, Dec. 2014, doi: 10.1109/TIFS.2014.2359548.

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.

Tangerang, 9 Desember 2020



Wilson Tandya - 13519209