

DETEKSI WAJAH MENGGUNAKAN METODE VIOLA JONES PADA GRAPHICS PROCESSING UNIT FACE DETECTION USING VIOLA JONES METHOD ON GRAPHICS PROCESSING UNIT

M .yogi Septian¹, Fitriyani²

¹Ilmu Komputasi, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Deteksi wajah adalah deteksi objek berupa wajah yang didalamnya terdapat fitur - fitur khusus yang merepresentasikan bentuk wajah pada umumnya. Salah satu metode deteksi wajah adalah dengan metode Viola Jones . Metode ini mempunyai empat proses utama yaitu, haar - like feature, citra integral , ada - boost , dan cascade classifier . Haar - like feature merupakan kumpulan fitur khusus yang merepresentasikan wajah dan citra integral adalah cara cepat menghitung haar feature . Sedangkan ada - boost adalah pembobotan secara statistik nilai - nilai fitur yang didapat dan di - filter menggunakan cascade classifier . Hasil terakhir berupa deteksi - deteksi yang lolos cascade classifier sebanyak 25 langkah. Dalam Tugas Akhir ini, metode Viola Jones dieksekusi secara paralel menggunakan GPU (Graphic Processing Unit) berbasis CUDA (Compute Unified Device Architecture). CUDA adalah platform komputasi paralel pada GPU yang dibuat oleh perusahaan NVIDIA. Data citra yang digunakan berformat Portable grayscale map (* .pgm). Sebagai perbandingan speedup , dilakukan juga eksekusi secara serial . Dari hasil penelitian, didapat bahwa metode Viola Jones pada GPU mempunyai akurasi yang baik. Didapat speedup 1.38 pada citra ukuran lebih kecil atau sama dengan 640x480, speedup 1.82 pada wajah ukuran diantara resolusi 640x480 sampai 1024x768, dan speedup 1.9 pada wajah ukuran diatas sama dengan 1024x768.

Kata Kunci : Deteksi Wajah, Viola Jones, GPU, CUDA

Abstract

Face Detection is a detection of face which there is a feature that represents a face's shape in general. One of the methods is Viola Jones. This method has four main processes; there are haar - like feature, integral image, ada - boost, and cascade classifier. Haar - like feature are pack of specific features which represent a face and integral image is the fast way to calculate haar feature. While ada - boost is a statistic weightening feature values and filtered by cascade classifier. The result is the detections which pass the 25 stages of cascade classifier. In this undergraduate thesis, Viola Jones method executed with GPU (Graphic s Processing Unit) based on CUDA (Compute Unified De vice Architecture). CUDA is a computation parallel platfor m on GPU made by NVIDIA Company. Image data that used is portable grayscale map (* .pgm). For the speedup comparison, also performed serial execution. From the r esult, we can say that Viola Jones met hod on GPU have good accuracy. Obtained 1.3 8 speedup on image resolution smaller equal to 640x480, 1.8 2 speedup on image resolution around 640x480 until 1024x768, and 1.9 speedup on image resolution larger equal to 1024x768

Keywords : Deteksi Wajah, Viola Jones, GPU, CUDA

2. DASAR TEORI

2.1. Dasar–Dasar Pengolahan Citra

Pengolahan citra berkaitan dengan manipulasi dan analisis dari citra digital. Citra–citra yang dibentuk dapat dimanipulasi sedemikian rupa hingga sesuai dengan keinginan, sebagai contoh proses *editing* foto yang biasanya dilakukan di studio. Bisa juga citra yang ada dianalisis sesuai dengan tujuan yang diharapkan. Dalam tugas akhir ini pengolahan citra digital yang dianalisis adalah bagaimana cara mendeteksi wajah dan dalam prosesnya ada cakupan materi yang terkait seperti persepsi pandangan manusia, bagaimana pengambilan citra, dan sampling dan kuantisasi citra.

a. Persepsi Pandangan Manusia

Persepsi pandangan manusia mempunyai peranan penting dalam pemilihan teknologi. Walaupun citra digital dibangun dari bentuk formulasi matematika dan probabilistik namun dalam prakteknya intuisi dan analisis manusia tetap diperlukan.[3]

b. Pengambilan Citra

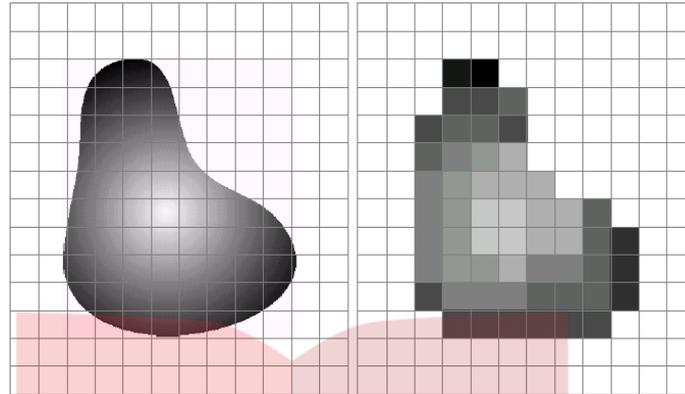
Citra dibentuk dari kombinasi sumber iluminasi (pencahayaan) dan pencerminan atau penyerapan sumber oleh bagian citra yang diambil. Pencahayaan yang baik dapat menghasilkan citra yang bagus dengan begitu Jarak pengambilan gambar juga berpengaruh terhadap citra yang dibentuk.[3]

Pengambilan jarak citra yang baik adalah dengan mempertimbangkan estetika objek dan latar belakangnya, untuk itu diperlukan intuisi dari pengambil gambar. Terlalu jauh objek yang ingin diambil maka semakin kabur citra tersebut, sebaliknya jika objek citra terlalu dekat maka objek maka pengenalan citra tidak dapat dikenali.

c. Sampling Citra dan Kuantisasi

Output dari citra digital berupa tegangan kontinu di mana amplitudo dan spasial dihubungkan ke perwujudan fisik yang ditangkap. Untuk membuat citra digital diperlukan proses pengkonversian dari kontinu ke bentuk digital, yaitu: sampling dan kuantisasi.

Sampling adalah pendigitalisasian nilai kordinat dimana setiap koordinat *continuous image* $f(x, y)$ di koordinat tertentu di ambil untuk merepresentasikan citra. Sedangkan kuantisasi adalah pendigitalisasian nilai amplitudo atau nilai dari tingkat keabuan yang dimana nilai tersebut dibuat ke bentuk diskritnya. Gabungan dari *sampling* dan kuantisasi ini disebut citra digital yang nantinya dapat diproses.[3]



Gambar 2-1(Kiri) Citra kontinue; (Kanan) Hasil *sampling* dan kuantisasi [3]

2.2. Deteksi Wajah

Deteksi wajah adalah suatu teknologi komputer untuk mendeteksi wajah manusia dengan cara menentukan letak dan ukuran wajah manusia di dalam citra digital. Teknologi ini dapat mendeteksi wajah melalui ciri/sifat wajah dan tidak memperdulikan hal hal lainnya, seperti bangunan, pohon dan badan manusia itu sendiri.

Bidang-bidang penelitian yang juga berkaitan dengan pemrosesan wajah (*face processing*) adalah autentikasi wajah (*face authentication*), lokalisasi wajah (*face localization*), penjejakan wajah (*facetracking*), dan pengenalan ekspresi wajah (*facial expression recognition*).[2]

Deteksi wajah merupakan salah satu tahap awal (*preprocessing*) yang sangat penting sebelum dilakukan proses pengenalan wajah (*face recognition*). Deteksi wajah dapat juga diartikan dengan deteksi benda yang spesifik. Dalam kasus ini benda yang dideteksi secara spesifik atau berupa wajah manusia yang sering disebut dengan istilah fitur. Yaitu bagian wajah manusia yang memiliki ciri khusus, seperti mata, hidung, mulut, pipi, dahi dan dagu.

Adapun faktor yang dapat mempengaruhi deteksi wajah, antara lain:

- a. Pose
Bagian wajah yang terlihat pada citra bisa bervariasi (bagian depan terlihat jelas, bagian wajah ada yang tidak terlihat).
- b. Komponen Struktural
Fitur pada wajah seperti kumis, jenggot, kacamata dan beberapa komponen yang bisa membuat wajah berbeda dari satu dengan yang lain. Seperti bentuk wajah, warna kulit, dan ukuran.
- c. Ekspresi Wajah
Ekspresi wajah yang ada pada citra.
- d. Orientasi Citra
Pengambilan gambar pada objek citra.
- e. Kondisi Citra

Kondisi pencahayaan (spektrum), dan karakteristik kamera (sensor, *response*, lensa) berpengaruh terhadap tampilan wajah.

2.3. Metode Deteksi Wajah

Untuk menentukan dan mendeteksi wajah, diperlukan metode-metode yang secara nalar bisa dipahami. Terutama sesuai cara pandang mesin komputer. Komputer melihat sebuah gambar tidak seperti apa yang dilihat oleh manusia. Komputer melihat gambar dalam bentuk angka matriks koordinat (m, n) , dimana m adalah baris dan n adalah kolom). Setiap angka merepresentasikan kode warna dalam suatu *pixel*. Besar matriks yang ditampung sesuai dengan lebar (w atau *width*) dan tinggi (h atau *height*) gambar tersebut.

[0,0]	[1,0]	...	[w,0]
[0,1]			⋮
⋮			⋮
[0,h]	[w,h]

Gambar 2-2 Representasi Citra

Dalam deteksi wajah empat istilah yang sering dipakai antara lain:

1. *True positive* = Wajah terdeteksi benar sebagai wajah.
2. *True negative* = Bukan wajah terdeteksi benar sebagai bukan wajah.
3. *False positive* = Wajah tidak terdeteksi sebagai wajah.
4. *False negative* = Bukan wajah terdeteksi sebagai wajah.

Metode deteksi wajah secara garis besar dibagi menjadi empat kategori berdasarkan pendekatan metode yang digunakan, antara lain: *knowledge-based methods*, *feature invariant approaches*, *template matching methods*, dan *appearance-based methods* [2]. Adapun tabel kategori dan metode pada deteksi wajah yang ada.

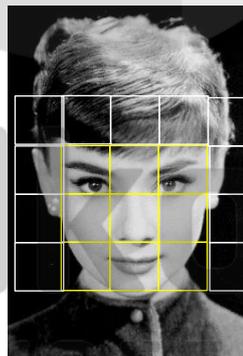
Tabel 2-1 Kategori metode untuk deteksi wajah pada citra

Pendekatan	Karya Representative
<i>Knowledge-based methods</i>	<i>Multiresolution rule-based method</i>
<i>Feature invariant approaches</i> - <i>Facial Feature</i> - <i>Texture</i>	<i>Grouping if edges</i> <i>Space Gray-Level Dependence matrix of face pattern</i>

- Skin Color - Multiple Feature	Mixture of Gaussian Integration of skin color, size, and shape
Template matching methods - Predefined face template - Deformable Templates	Shape template Active Shape Model
Appearance-based methods - Eigenface - Distribution-based - Neural Network - Support Vector Machine - Naïve Bayes Classifier - Hidden Markov Model - Information-Theoretical Approach	Eigenvector decomposition and clustering Gaussian distribution and multilayer perceptron Ensemble of neural networks and arbitration schemes SVM with polymial kernel Joint statistics of local appearance and positions Higher order of startistics with HMM Kullback relative information

2.3.1. Knowledge-based Methods

Pada metode ini, deteksi wajah dikembangkan berdasarkan *rules* atau aturan yang simpel tentang pengetahuan representasi wajah manusia yang dimiliki ilmuwan. Aturan yang digunakan adalah dengan cara menghubungkan fitur-fitur yang ada pada wajah. Sebagai contoh, wajah adalah bagian citra yang memiliki mata yang simetris satu sama lain, memiliki hidung dan mulut. Hubungan antar fitur-fitur ini direpresentasikan *relative distance and positions*. Fitur wajah di inputan citra di ekstrak terlebih dahulu, dan kandidat wajah diidentifikasi berdasarkan peraturan yang telah dibuat. Proses verifikasi dilakukan untuk mengurangi *false detection*[2].



Gambar 2-3 Representasi metode *Knowledge-based* [2]

Pada gambar 2-3 dapat dilihat bahwa ada kotak dengan 2 warna berbeda. Kotak putih mempunyai 4 baris kotak yang merepresentasikan bagian citra dikotak tersebut yang memiliki perbedaan rata-rata *grayscale* yang kecil. Sedangkan di kotak kuning

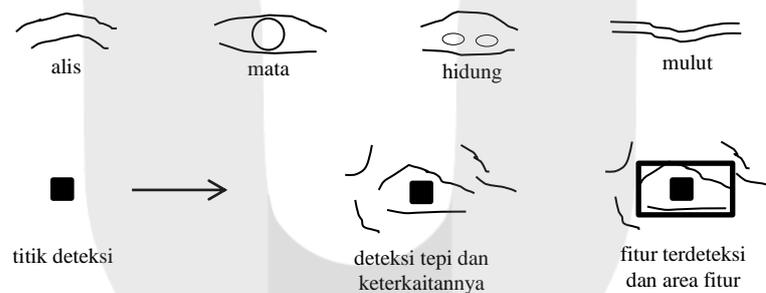
memiliki perbedaan rata-rata *grayscale* yang besar. Dari sini dapat dilihat hubungan tingkat *grayscale* dan lokasi wajah.

Ada satu masalah dalam deteksi menggunakan metode ini, yaitu sulitnya menterjemahkan pengetahuan dengan representasi wajah kedalam sebuah aturan yang terdefinisi dengan baik. Jika aturan yang dibuat detail, memungkinkan terjadinya *true negative*, dimana wajah tidak terdeteksi oleh sistem [2].

2.3.2. Feature Invariant Approaches

Metode ini merupakan lawan dari metode *Knowledge-based* yang bertujuan untuk menemukan struktur fitur wajah meskipun pose, sudut pandang, atau pencahayaan beragam, dan dengan temuan ini dapat mengetahui letak wajah. Ilmuan berasumsi berdasarkan cara manusia dapat mendeteksi wajah dan objek. Dari sana ciri atau fitur yang sama akan didapat dari semua keragaman data yang ada.

Ada banyak metode-metode yang diajukan untuk mengetahui fitur wajah dan menyimpulkan keberadaan wajah pada suatu citra. Fitur wajah seperti alis, mata, hidung, mulut, dan garis pada rambut secara umum diekstrak menggunakan deteksi tepi. Berdasarkan pengestraksian fitur, model statistika dibangun untuk mendeskripsikan hubungan dan untuk memverifikasi keberadaan wajah.



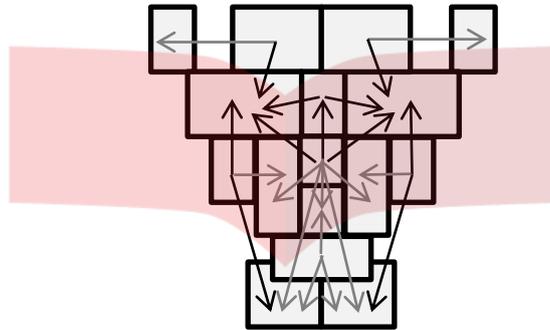
Gambar 2-4 Representasi metode *feature invariant* [2]

Pada gambar 2-4 setiap fitur yang ada pada wajah dimodelkan sebagai pasangan yang berorientasi kepada tepi-tepinya. Metode ini memiliki *point of interest* untuk menentukan titik fitur. Sebagai contoh, titik deteksi berada pada suatu posisi citra. Kemudian dari titik tersebut, dilakukan deteksi tepi ke area sekitar titik deteksi. Deteksi tepi dapat memperjelas fitur yang ada pada citra sekitar. Setelah proses deteksi tepi selesai, hasil deteksi tepi tersebut dibandingkan dengan fitur yang telah disimpan seperti fitur: alis, mata, hidung, dan mulut. Pada contoh ini terdeteksi fitur mata. Dapat dilihat pada gambar, area fitur direpresentasikan sebagai kotak berwarna hitam tebal.

Ada masalah pada metode ini, jika fitur pada citra sangat rusak dikarenakan adanya masalah pada pencahayaan, *noise*, atau terjadi kesalahan pada pengambilan citra. Cakupan fitur pada wajah melemah, dengan itu pula bayangan dapat mempengaruhi deteksi tepi yang menyebabkan persepsi algoritma menjadi tidak efisien [2].

2.3.3. *Template Matching Methods*

Metode *Template matching* merupakan metode dengan beberapa standar pola wajah yang tersimpan untuk mendeskripsikan wajah secara keseluruhan atau bagian fitur wajah tertentu saja. Korelasi antara citra masukan dan pola yang tersimpan adalah perhitungan deteksi yang dilakukan. Pendekatan ini mempunyai kelebihan karena mudah untuk diimplementasikan.



Gambar 2-5 Representasi metode *template matching* [2]

Pada gambar 2-5 dapat dilihat terdapat kotak abu-abu sebanyak 16 kotak. Kotak ini merepresentasikan *pixel* yang ada di lokasi wajah. Terdapat panah yang menghubungkan satu kotak ke kotak lainnya sebanyak 23 buah relasi yang terdefinisi. Relasi yang terdefinisi ini diklasifikasi lagi menjadi 2 yaitu 11 panah (panah hitam) yang menunjukkan hubungan relasi penting, dan 12 panah (panah abu-abu) menunjukkan penegasan hubungan relasi penegasan. Setiap panah pada gambar menunjukkan sebuah relasi, dan ujung panah menunjukkan area kedua (penyebut dari sebuah rasio). Sebuah relasi terpenuhi jika rasio antara dua buah area melebihi *threshold* dan lokasi wajah ditemukan jika nilai hubungan relasi penting dan hubungan relasi penegasan melebihi *threshold*.

2.3.4. *Appearance-based Methods*

Metode ini merupakan lawan dari metode *Template Matching*, model matematis didapat dari kumpulan citra latih yang merepresentasikan variasi tampilan pada wajah. Model ini yang digunakan sebagai deteksi. Secara umum, metode ini menggunakan pendekatan secara analisis statistika dan *machine learning* untuk menemukan fitur yang membedakan antara wajah dan bukan wajah. Salah satunya adalah metode Viola Jones.

Dalam tugas akhir ini dipilih Viola Jones karena metode ini memiliki kecepatan 15x lipat dari metode-metode deteksi wajah lainnya.

“We have presented an approach for face detection which minimizes computation time while achieving high detection accuracy. The approach was used to construct a face detection system which is approximately 15 times faster than any previous approach.” [1]

Metode ini masih dikembangkan dalam masalah deteksi objek secara luas. Metode ini juga sering termasuk dalam *project* yang diadakan oleh Google dan Microsoft karena memiliki kecepatan yang jauh cepat dibandingkan metode lainnya

tanpa mengurangi secara signifikan akurasi deteksi. Oleh karena itu metode ini banyak dipakai untuk kasus deteksi.

Dibawah ini adalah tabel keakuratan deteksi wajah Viola Jones dengan metode yang di-publish peneliti lainnya. Dimana *false detection* maksudnya adalah terdeteksi wajah oleh sistem tapi sebenarnya tidak atau disebut juga dengan *false negative*.

Tabel 2-2 Tabel Akurasi berdasarkan *False Detection* [1]

Detector	False Detection							
	10	31	50	65	78	95	167	422
Viola Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%	94.1%
Viola Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2%	93.7%	
Rowley-Baluja-Kanade	-	-	-	-	-	89.2	90.1	89.9%
Schneiderman-Kanade				94.4%	-	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8)%	-	-	-

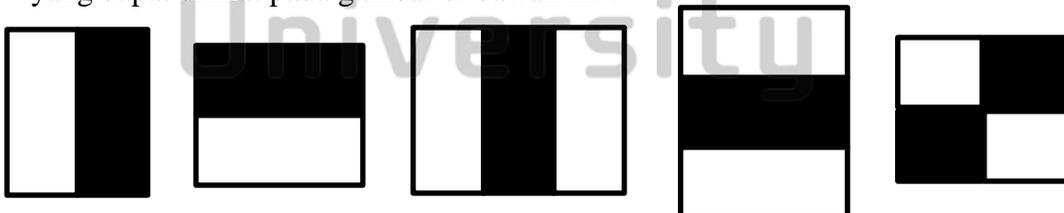
Dari tabel 2-2 menunjukkan bahwa semakin banyak *false detector* yang digunakan maka semakin besar akurasi deteksi wajah.

2.4. Viola Jones

Deteksi wajah dengan Viola Jones menggunakan fitur simple *Haar-like* yang mengevaluasi dengan cepat representasi citra yang baru. Viola Jones meng-generate kumpulan fitur dengan citra integral dan *boosting algoritm* untuk mengurangi kompleksitas waktu. Sebelum dimasukkan ke dalam sistem terlebih dahulu citra dicari nilai keabu-abuannya (*grayscale*). Secara umum metode Viola Jones memiliki empat dasar proses, yaitu:

a. Haar-like feature

Klasifikasi citra dilakukan berdasarkan nilai dari sebuah fitur. Hal ini bertujuan untuk memisahkan citra yang tidak diperlukan, dalam kasus ini, background tidak ikut dihitung. Terdapat 3 jenis fitur berdasarkan jumlah persegi panjang (terang dan gelap) yang terdapat di dalamnya, yaitu: dua,tiga,empat persegi panjang seperti yang dapat dilihat pada gambar di bawah ini :



Gambar 2-6 Fitur *Haar-like* [1]

Pada gambar 2-6 dapat dilihat bahwa terdapat 5 fitur dari kiri ke kanan, pertama dan kedua terdiri dari dua persegi panjang, sedangkan fitur ketiga dan keempat terdiri dari tiga persegi panjang dan fitur kelima empat persegi panjang. Cara menghitung nilai dari fitur ini adalah mengurangkan nilai *pixel* pada area hitam dengan *pixel* pada area putih. Dengan Rumus untuk mendapatkan nilai fitur sesuai dengan jumlah kotak:

B, W : *Black* (Hitam), *White* (Putih)

Dua kotak : $W - B$

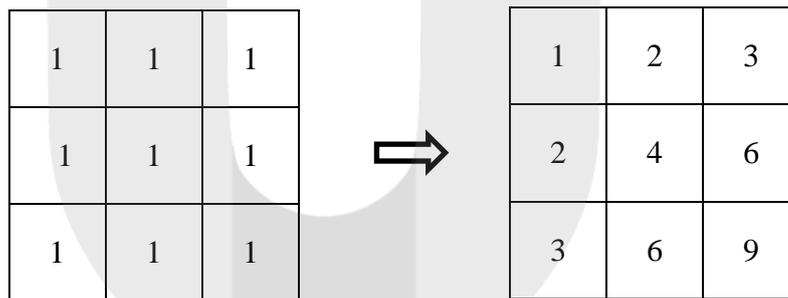
Tiga kotak : $W_1 + W_2 - B$

Empat kotak : $(W_1 + W_2) - (B_1 + B_2)$

Dengan begitu akan dapat ditentukan tingkat *luminance* dari citra yang akan dideteksi dan dapat dibedakan mana bagian citra yang mencirikan wajah manusia. Untuk mempermudah proses penghitungan nilai fitur, metode Viola Jones menggunakan sebuah media berupa citra integral.

b. Citra Integral

Citra integral adalah struktur data dan algoritma yang menjumlahkan nilai-nilai dalam *subset* di matriks citra [1]. Penjumlahannya antar baris dan kolomnya, sebagai ilustrasi:



Gambar 2-7 Ilustrasi perhitungan citra integral

Awalnya matriks :

Baris pertama = 1,2,3. Baris kedua = 4,5,6. Baris ketiga = 7,8,9

Prosesnya:

Baris pertama = 1, (1+1), (1+1+1)

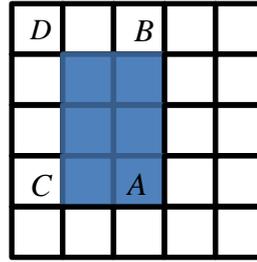
Baris kedua = (1+1), (1+1+1+1), (1+1+1+1+1+1)

Baris ketiga = (1+1+1), (1+1+1+1+1+1), (1+1+1+1+1+1+1+1+1)

Maka diperoleh hasil:

Baris pertama = 1,2,3 Baris kedua = 2,4,6 dan baris ketiga = 3,6,9

Dengan mendapatkan citra integral, penghitungan nilai area dapat dilakukan dengan cara :



$$SUM \text{ [blue square]} : A - B - C + D$$

Gambar 2-8 Ilustrasi perhitungan nilai area

Dari gambar 2-8 untuk menentukan nilai area tidak perlu menjumlah tiap *pixel*. Hanya dengan mengurangi nilai yang sudah di citra integral-kan pada bagian kanan bawah area dengan bagian luar pojok kiri bawah, dan bagian luar pojok kanan atas. Lalu ditambah dengan nilai pada bagian pojok serong kiri area.

Proses pencarian nilai fitur ini dilakukan secara iteratif mulai dari ujung kiri atas hingga ujung kanan bawah dengan pergeseran sebesar Δx dan Δy . Semakin kecil nilai Δx dan Δy , maka semakin akurat deteksi citra tersebut. Nilai Δx dan Δy yang sering digunakan adalah 1.

Permasalahan yang terdapat dalam penghitungan fitur ini adalah Viola Jones memiliki 160.000 jenis fitur yang berbeda. Jumlah ini terlalu besar sehingga tidak mungkin dilakukan penghitungan untuk semua fitur. Hanya fitur-fitur tertentu saja yang dipilih untuk diikutsertakan. Pemilihan fitur-fitur ini dilakukan menggunakan algoritma Ada-Boost.[4]

c. Ada-Boost

Dalam prakteknya tidak satupun fitur yang mampu melakukan pengklasifikasian dengan *error* yang kecil [1]. Algoritma Ada-Boost berfungsi untuk mencari fitur-fitur yang memiliki tingkat pembeda yang tinggi. Hal ini dilakukan dengan mengevaluasi setiap fitur terhadap terbesar antara wajah dan non-wajah dianggap sebagai fitur terbaik [5].

Citra-citra inputan $(x_1, y_1), \dots, (x_n, y_n)$ dimana $y_i = 0, 1$ untuk contoh negative dan positif secara berurutan.

Inisialisasi bobot $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ untuk $y_i = 0, 1$ berurutan, dimana m adalah jumlah citra negatif dan l merupakan jumlah citra positif.

Untuk $t = 1, \dots, T$ dilakukan:

- 1) Normalisasi bobot:

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}} \tag{2.1}$$

w_t merupakan distribusi probabilitas.

- 2) Untuk setiap fitur, j , latih *classifier* h_j yaitu yang dibatasi agar menggunakan fitur tunggal. Tingkat *error* dievaluasi dengan memperhatikan:

$$w_t, \varepsilon_j = \sum_i w_i |h_j(x_i) - y_i| \tag{2.2}$$

- 3) Pilih *min-error classifier* h_t :

$$\varepsilon = \min_{f, p, \theta} \sum_i w_i |(x_i, f, p, \theta) - y_i| \tag{2.3}$$

- 4) Perbarui bobot:

$$w_{t+1, i} = w_{t, i} \beta_t^{1-e_i} \tag{2.4}$$

Dimana $e_i = 0$ Jika x_i diklasifikasi sebagai wajah, sebaliknya $e_i = 1$ jika x_i diklasifikasi bukan wajah maka atur bobot ke bawah:

$$\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t} \tag{2.5}$$

Terakhir menggabungkan secara *linier weak classifier* yg dibentuk, *Final (Strong) Classifier*:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T a_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{Sebaliknya} \end{cases} \tag{2.6}$$

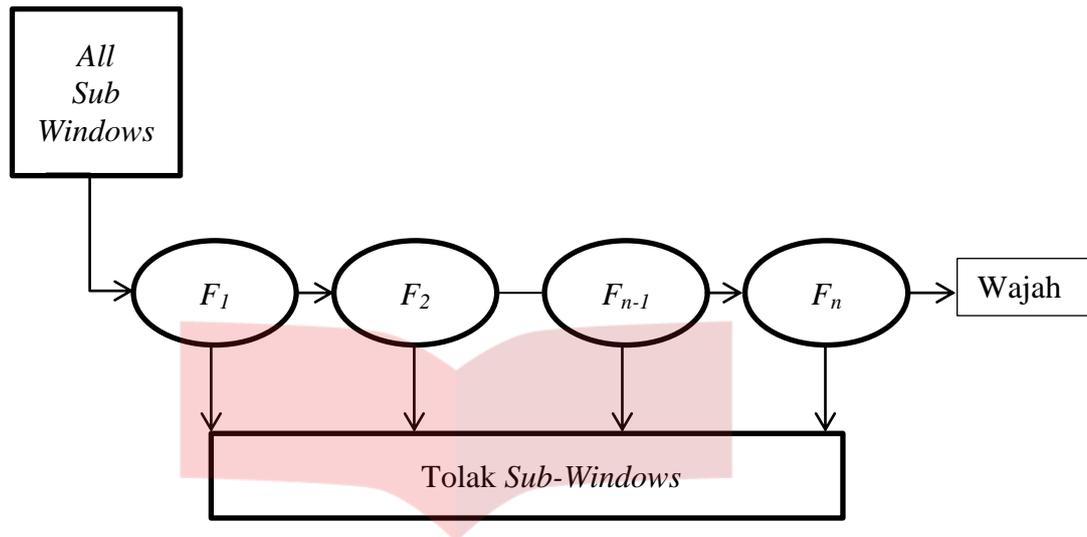
Dimana $\alpha_t = \log \frac{1}{\beta_t}$

d. *Cascade Classifier*

Karakteristik dari metode Viola Jones adalah adanya klasifikasi bertingkat (*cascade classifier*). Klasifikasi pada algoritma ini terdiri dari beberapa tingkatan dan tiap tingkatan mengeluarkan subcitra yang diyakini bukan wajah. Hal ini dilakukan karena lebih mudah untuk menilai subcitra yang bukan wajah daripada menilai apakah subcitra tersebut berisi wajah [1].

Setiap *subwindows* dibandingkan dengan setiap fitur di setiap *stage*. Jika tidak mencapai *target* maka *subwindows* akan bergerak ke *subwindows* berikutnya dan melakukan perhitungan yang sama dengan proses sebelumnya.





Gambar 2-9 Ilustrasi *Cascade Classifier*

Dari gambar 2-9 pada klasifikasi tingkat pertama, tiap subcitra pada subwindows diklasifikasi menggunakan beberapa fitur *haar-like*. Jika subcitra mencapai *threshold* maka proses berlanjut ke stage berikutnya. Tetapi jika tidak mencapai *threshold* maka subwindows ditolak dan proses berlanjut ke subcitra berikutnya.

Pada proses selanjutnya didapat hasil yaitu subwindows yang terdeteksi sebagai wajah dan berlanjut ke subcitra berikutnya. Sampai pada akhirnya didapat kandidat kuat yang terdeteksi sebagai wajah [1].

2.5. GPU Programming

GPU Programming adalah penyelesaian masalah dengan menggunakan sumber atau alat dari GPU. Cara ini memiliki kecepatan yang lebih cepat dibandingkan menggunakan CPU (biasanya). Karena pada GPU programming pengekseskuan perintah dapat dilakukan sekaligus atau dengan kata lain semua eksekusi mempunyai modul-modul sendiri seperti orang yang melakukan pekerjaan secara bersamaan. Salah satu bahasa untuk GPU programming adalah OpenCL yang sudah menjadi hak milik CUDA (NVIDIA).[6]

Perilisan GPU yang memiliki baris data yang bisa diatur menarik perhatian ilmuwan untuk digunakan perhitungan komputasi secara umum. Namun pada pengaplikasiannya penggunaan GPU sebagai media komputasi sangatlah berbelit-belit. Karena standar grafik API (*Application Programming Interface*) seperti OpenGL dan DirectX membatasi kegunaan GPU itu sendiri. Untuk itu para ilmuwan meneliti komputasi secara umum tanpa API dengan cara membuat *rendering* permasalahan secara konvensional agar dapat dilihat dengan sudut pandang GPU. [7]

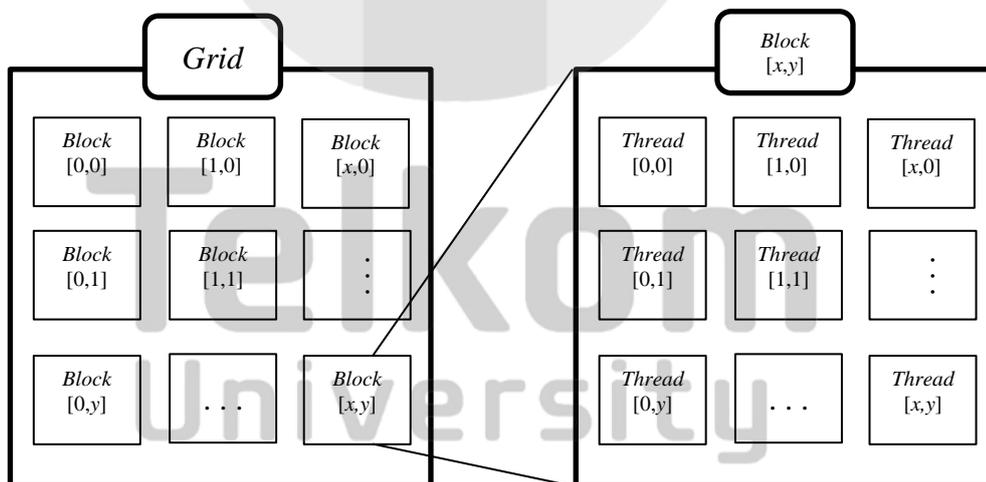
Pada dasarnya, tahun 2000-an GPU didesain untuk menghasilkan warna untuk setiap *pixel* pada layar menggunakan *arithmetic units* atau lebih dikenal dengan

sebutan *pixel shaders*. Secara umum, *pixel shaders* menggunakan posisi (x, y) pada layar serta beberapa informasi tambahan untuk menggabung bermacam-macam inputan didalam perhitungan warna akhir. Informasi tambahan tersebut antara lain warna inputan, koordinat tekstur, atau atribut lain yang berjalan pada *shader*. Tetapi karena *arithmetic* ini merupakan inputan warna dan tekstur sepenuhnya dikendalikan oleh programmer, ilmuwan mengamati bahwa inputan “warna” ini bisa “data” apa saja.

Jadi jika inputan berupa data numeric, programmer dapat mengolah *pixel shader* untuk melakukan perhitungan komputasi dengan data ini. Hasilnya berupa warna *pixel* akhir, meskipun hasilnya bisa berupa apa saja. Hal ini diteliti kembali oleh ilmuwan, dan penggunaan GPU bukan cara bijak menyelesaikan masalah. Intinya, GPU dimanipulasi untuk melakukan pekerjaan *nonrendering* dengan cara membuat pekerjaan tersebut seperti *rendering* pada umumnya. Cara ini sangat cerdas tetapi sangat sukar pula. [7]

CUDA memiliki *unified shader pipeline* yang dapat setiap ALU (*Arithmetic Logic Unit*) di dalam *chip* dijalankan di program yang ditujukan untuk komputasi secara umum. Karena NVIDIA sengaja membuat GPU untuk komputasi secara umum. ALU ini dibuat sesuai dengan persyaratan IEEE (*Institute of Electrical and Electronics Engineers*) untuk *single-precision floating-point arithmetic* dan didesain untuk komputasi secara umum daripada khusus untuk *graphics*. Selanjutnya, *execution unit* pada GPU secara acak mempunyai akses membaca dan menulis ke memori serta mempunyai akses ke *software-managed cache* yang dikenal dengan nama *shared memory*. Semua fitur arsitektur CUDA ini ditambahkan bertujuan untuk membuat GPU dapat mengatasi komputasi sebagai tambahan dari pengolahan gambar pada umumnya.

Secara garis besar, CUDA memiliki *Grid*, *Block*, dan *Thread* untuk memproses data pada GPU. Ilustrasinya sebagai berikut:



Gambar 2-10 Ilustrasi *grid*, *block* dan *thread*

Dari gambar 2-10 dapat dilihat di dalam *grid* terdapat *block*, dan di dalam *block* terdapat *thread*. Banyaknya *block* dan *thread* yang digunakan tergantung kebutuhan. Maksimum banyaknya *thread* dan *block* tergantung *device* yang dipakai. Setiap *thread* pada *block* berjalan secara bersamaan dan independen. Artinya data yang diproses pada *block* lain tidak bisa saling bertukar informasi. Maka tidak semua masalah dapat diparalelkan dan diperlukan observasi terhadap masalah.

2.6. Akurasi

Pada umumnya untuk mendapatkan akurasi dalam deteksi objek diperlukan *rules* dan standar khusus untuk menyatakan deteksi akurat. “Receiver Operating Characteristics (ROC) analysis has been borrowed from Signal Processing to become standard for evaluation and standard setting, comparing True Positive Rate and False Positive Rate.”[10]

Analisis dengan ROC sudah menjadi standar evaluasi dan pengaturan dengan membandingkan *True Positive Rate* dan *False Positive Rate*. Untuk kasus deteksi wajah, tentu diperlukan cara manual (peneliti mengidentifikasi langsung) untuk menentukan *True Positive*, *False Positive*, dan *False Negative*. Adapun Ilustrasi hubungan antar atribut:

Tabel 2-3 Ilustrasi hubungan antar atribut

	P	N	
T	TP	FP	PPV
F	FN	TN	FOR
	TPR	FPR	N

$$\text{True Positive (TP)} = TF \tag{2.7}$$

$$\text{False Positif (FP)} = WD - TP \tag{2.8}$$

$$\text{False Negative (FN)} = WK - TP \tag{2.9}$$

Dimana,

- TF* (*True Positif*) = jumlah wajah yang terdeteksi dengan benar
- WK* (*Wajah Keseluruhan*) = jumlah wajah yang ada dalam citra
- WD* (*Wajah Dideteksi*) = jumlah seluruh wajah yang terdeteksi

$$\text{True Positive Rate (TPR)} = \text{Recall} = \frac{TP}{TP + FN} \tag{2.8}$$

$$\text{False Positive Rate (FPR)} = \text{Fall-out} = \frac{FP}{FP + TN} \tag{2.9}$$

Pada rumus (2.8) *recall* atau *sensivity* merupakan wajah yang terdeteksi dengan benar dibagi wajah yang dideteksi dengan benar ditambah wajah yang tidak

terdeteksi atau dapat diambil diartikan juga wajah yang terdeteksi dengan benar dibagi jumlah target wajah yang ada pada citra.

Pada rumus (2.9) untuk mendapatkan *FPR* diperlukan *True Negative (TN)*. Namun pada kenyataannya dalam deteksi wajah sangat sukar untuk menentukan *TN* tersebut. *TN* dalam deteksi wajah mempunyai makna wajah yang ditolak. Hal ini dapat ditanggulangi dengan mencari nilai *Positive Predicted Value (PPV)* karena *TPR* dan *PPV* tidak memerlukan nilai *TN* pada perhitungannya.[10]

$$\text{Positive Predicted Value (PPV)} = \text{Precision} = \frac{TP}{TP + FP} \tag{2.10}$$

Pada rumus (2.10) *precision* atau nilai *confidence* merupakan ketepatan deteksi yang menyatakan seberapa besar deteksi dengan benar dapat diraih. Adapun cara perhitungannya akurasi menggunakan nilai rata-rata *F_{measure}* dari *precision* dan *recall*. [11] yaitu sebagai berikut:

$$F_{measure} = 2 \times \left(\frac{TPR \times PPV}{TPR + PPV} \right) \tag{2.11}$$

2.7. Performansi

Pada pengolahan citra, performansi salah satu faktor yang penting sebagai tolak ukur cara atau algoritma yang dipilih dalam kasus tertentu. Untuk menghitung kecepatan deteksi dilakukan perhitungan:

Kecepatan deteksi (ms) :

$$\frac{t_{akhir} - t_{awal}}{1000} \tag{2.12}$$

Dimana,

t = waktu (detik)

Untuk mendapatkan efisiensi kinerja GPU untuk pengaplikasian Viola Jones, diperlukan suatu barometer yaitu *speedup*. “*Speedup is the ratio, speedup tells us how much faster a task will run using the computer with the enhancement as opposed to the original computer.*” [9]

Speedup adalah rasio yang dapat merepresentasikan seberapa cepat sebuah pekerjaan dapat diselesaikan dengan komputer yang sudah diperkuat dengan membandingkan komputer awal. *Speedup* didapat dengan membandingkan hasil rata-rata kecepatan deteksi pada CPU dan GPU, Yaitu:

Speedup :

$$\frac{T_{CPU}}{T_{GPU}} \tag{2.13}$$

Dimana,

T_{CPU} = rata-rata waktu CPU

T_{GPU} = rata-rata waktu GPU

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil pengujian dan pembahasan pada bab-bab sebelumnya dapat disimpulkan, bahwa:

- 1) Implementasi GPU pada tugas akhir ini bisa memiliki akurasi yang baik dan sesuai harapan penulis dimana Viola Jones dapat mendeteksi wajah dengan akurat. Namun untuk *speedup* GPU belum sesuai dengan harapan penulis, dimana untuk data citra yang besar harusnya *speedup* yang didapat lebih tinggi dari hasil yang diuji. Dikarenakan semua fungsi dalam metode Viola Jones belum sepenuhnya diubah ke GPU.
- 2) Dari hasil pengujian didapat *speedup* terendah dari data ke-1 sebesar 0.35 dan *speedup* tertinggi dari ke-80 sebesar 3.55.
- 3) GPU bekerja lebih lambat dari CPU untuk resolusi ukuran citra dibawah 65536 *pixel* dan GPU bekerja lebih cepat dari CPU untuk resolusi ukuran citra lebih besar sama dengan 65536 *pixel*.
- 4) Akurasi semua golongan citra relatif akurat sebesar 85.9155%.
- 5) Untuk citra golongan A memiliki akurasi dan *speedup* sebesar 87.7193% dan 1.38.
- 6) Untuk citra golongan B memiliki akurasi dan *speedup* sebesar 87.7879% dan 1.82.
- 7) Untuk citra golongan C memiliki akurasi dan *speedup* sebesar 84.5921%, dan 1.9.

5.2. Saran

Adapun saran untuk pengembangan tugas akhir ini adalah sebagai berikut:

- 1) Mencari *precision* dan *recall* yang seimbang dengan melatih data yang bervariasi.
- 2) Membuat deteksi terhadap objek lain seperti deteksi kendaraan bermotor, badan manusia dan sebagainya.
- 3) Membuat inputan data citra selain *Portable Gray Map* (*.pgm).

DAFTAR PUSTAKA

- [1] Viola Paul, Jones Michael.2004.*Robust Real – Time Face Detection*.25(25).4-16.*International Journal of Computer Vision* 57(2), 137-154
- [2] Yang, M.H., Kriegman, D., Ahuja, N.2002.Detecting Faces in Images: A Survey , *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no1
- [3] Prasetyo,Eko.2011.Pengolahan Citra Digital dan Aplikasinya menggunakan Matlab.ANDI.Yogyakarta
- [4] Adam Harvey.(2012).Adam Harvey Explains Viola Jones Face Detection[online] .Tersedia: <http://makemantics.com/research/viola-jones> [2012, Desember 19]
- [5] Tarhini, Ali.2011. *Efficient Face Detection Algorithm using Viola Jones method*. <http://www.codeproject.com/Articles/85113/Efficient-Face-Detection-Algorithm-using-Viola-Jon>
- [6] Harvey, Jessy Patrick.2009.GPU Acceleration of Object Classification Algorithms Using NVIDIA CUDA: A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering.Rochester Institute of Technology: Rochester, NY, United States.
- [7] Sander, Jason., Edward Kandrot.2011.*CUDA By Example: An Introduction to General-Purpose GPU Programming*.NVIDIA Corporation: Boston, MA, United States.
- [8] Mei Wei, Hwu W.2010.*GPU Computing Gems: Emerald Edition*.Elsevier Inc:Burlington,MA,United States
- [9] Hennessy John L., Petterson David A.2007.*Computer Architecture A Quantative Approach Fourth Edition*. Elsevier Inc: Oxford,United Kingdom
- [10] Powers, David M. W.*Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation*.
- [11] Agarwal, S., Awan, A., Roth, D.: Learning to Detect Objects in Images Via a Sparse, Part-Based Representation. *IEEE TPAMI* 26(11) (2004) 1475–1490