ALGORITMA HAAR CASCADE CLASSIFIER UNTUK MENGENALI OBJEK TERSEBAR

Muhamad Rikat Rifaldi[1], Nelly Indriani W., S.Si., M.T.[2]

[1,2] Universitas Komputer Indonesia
Jalan Dipatiukur No. 112-116, Coblong, Bandung, Jawa Barat 40132
E-mail : rikatrifaldi@ymail.com[1], nelly.indriani@email.unikom.ac.id[2]

ABSTRAK

Permainan *hidden object* adalah permainan yang menggunakan kejelian mata dalam mencari objek yang tersembunyi. Objek-objek tersebut dapat tersusun secara acak, horizontal,vertikal dan terlihat sebagian, adapun dalam sebuah permainan *hidden object* terdapat kesulitan dalam mengenali objek dikarenakan posisi objek yang dicari tersembunyi ataupun posisi objek hanya terlihat sebagian saja sehingga objek tersebut sulit ditemukan, sehingga di bangunlah suatu simulator permainan *hidden object* untuk mendapatkan solusi dalam mengenali beberapa objek gambar yang tersebar.

Aplikasi simulator permainan *hidden object* untuk mendeteksi objek tersebar mengunakan pengolahan citra dengan menggunakan metode *haar classifier*. Pada jenis permainan ini objek-objek tersebar terletak pada map yang telah disediakan pada permainan, untuk memainkan jenis game ini user terlebih dahulu memilih map yang didalamnya terdapat objek-objek yang letaknya tersebar, kemudian sistem mencari objek gambar dan menandai semua objek-objek tersebar yang letaknya tersembunyi di dalam map sesuai dengan objek yang akan dideteksi. Hasil uji performansi dari simulator game *hidden object* menunjukkan bahwa simulator ini dapat membantu dalam mendeteksi objek-objek tersebar. Aplikasi pendeteksi objek tersebar pada game *hidden object* menggunakan metode *Haar Cascade Classifier*, dan dapat mendeteksi objek tersebar dengan pengujian performansi berupa objek posisi normal, 90 derajat, 180 derajat, 270 derajat dan terlihat sebagian memiliki nilai 83,5% terdeteksi.

Kata Kunci: *Game, Haar cascade classifier, haar-like feature, hidden object.*

1. PENDAHULUAN

Pada bagian ini akan dipaparkan mengenai latar belakang masalah, maksud, tujuan dan teori yang digunakan.

1.2 Latar Belakang Masalah

Jenis permainan *hidden object* merupakan permainan yang menggunakan kejelian mata untuk mengenali objek-objek benda. Pada jenis permainan ini objek-objek tersebar dari sebuah objek gambar yang terletak pada grid. Untuk memainkan jenis game ini harus mencari dan menemukan objek gambar dan menandai semua objek-objek tersebar yang letaknya tersembunyi di dalam objek gambar pada grid agar pemain dapat melanjutkan permainan. Dalam jenis permainan *hidden object*, daftar objek tersebar yang tersembunyi telah disediakan dan biasanya ada beberapa objek yang sulit bagi pemain untuk dicari menurut presentase yang telah didapatkan dari hasil pembagian kuesioner yang menyatakan bahwa terdapat beberapa objek yang sulit ditemukan dengan dengan nilai 57,4% dan yang menyatakan tidak sulit 42,56%. Penyelesaian dari jenis permainan *hidden object* ini adalah menemukan semua objek yang tersebar yang letak objeknya horizontal, vertikal, diagonal dan bisa juga terlihat sebagian.

Penelitian sebelumnya dengan judul haar cascade classifier dan algoritma adaboost untuk deteksi banyak wajah dalam ruang kelas [1]. Dimana pada penelitian tersebut dilakukan dengan membandingkan citra uji dengan sampel 8 orang dan setiap orangnya memiliki 8 citra wajah dan posisi wajah tegak lurus, rotasi 15^o ke kanan, rotasi 15^o ke kiri, dan mengangkat dagu 15^o ke atas dan menunduk kepala

1.2 Maksud dan Tujuan

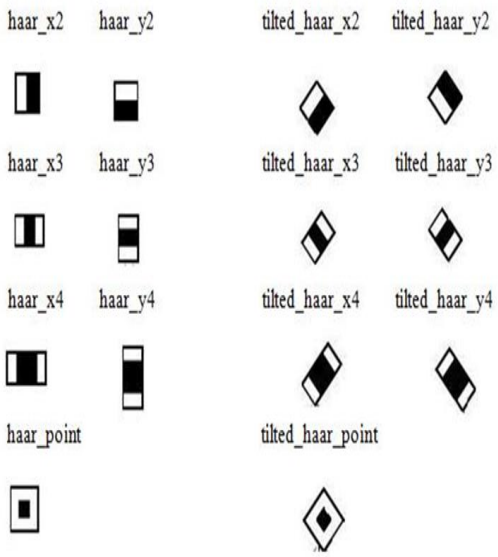
Berdasarkan permasalahan yang diteliti, maka maksud dari penelitian ini adalah mendapatkan solusi dalam mengenali beberapa objek gambar yang tersebar terlepas dari posisi objek dan rotasi objek berada.

Sedangkan tujuan yang diharapkan akan dicapai dalam penelitian ini adalah mengetahui performansi dalam akurasi algoritma *haar cascade classifier* untuk menemukan objek gambar yang diminta dalam objek gambar yang tersebar.

1.3 *Haar Like Fiture*

Algoritma Haar menggunakan metode statistikal dalam melakukan pendeteksian objek. Tiap feature dari *Haar-like feature* didefinisikan pada bentuk dari *feature*, diantaranya koordinat dari feature dan juga ukuran dari *feature* tersebut. *Feature* yang akan di training adalah feature yang didapat dari proses

cropping. Adapun macam-macam variasi feature haar :



Gambar 1. Haar like feature [2]

Tiap feature terbagi atas dua bagian, yaitu bagian berwarna putih dan bagian berwarna hitam. Penggunaan fitur dilakukan karena pemrosesan fitur berlangsung lebih cepat dibandingkan pemrosesan citra perpiksel sehingga cocok untuk diterapkan pada proses pendeteksian objek secara real time. Pada penelitian ini untuk menghitung nilai fitur Haar pada sebuah citra dan pada skala yang berbeda secara cepat menggunakan satu teknik yang disebut integral image.

1.4 Citra Digital

Citra adalah representasi, kemiripan dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video atau bersifat digital yang dapat lansung disimpan dalam suatu media penyimpanan.

Citra digital adalah citra yang dapat diolah menggunakan komputer. Yang disimpan dalam sebuah memori komputer hanyalah angka-angka yang menunjukan besar intensitas pada masing-masing piksel. Karena citra digital berbentuk data numeris, maka citra digital dapat diolah mengunakan komputer [3].

1.4.1 *Grayscaling*

Grayscale adalah jenis gambar yang menangani gradiasi warna hitam dan putih, yang menghasilkan efek warna abu-abu. Pada gambar ini, warna dinyatakan dengan intensitas. Dalam hal ini intensitas berkisar antara 0 sampai 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih. Konversi gambar normal (RGB) ke gambar *grayscale* dilakukan dengan menggunakan rumus. [3].

Grayscale=(0.299*R)+(0.587*G)+(0.114*B)

1.4.2 *Tresholding*

Tresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra.. Proses tresholding ini pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan *tresholding* dengan derajat keabuan dapat digunakan rumus [4]:

$$"g(x,y) = " \begin{cases} 1 \text{ if } f(x,y)>T \\ 0 \text{ if } f(x,y)\leq T \end{cases}$$

Keterangan :

- Putih Jika level keabuannya >T
- Hitam jika level keabuannya <= T

Proses tresholding akan menghasilkan citra biner, yaitu citra yang memiliki dia nilai tingkat keabuan yaitu hitam dan putih. Secara umum proses tresholding citra grayscale untuk menghasilkan citra biner adalah sebagai berikut:

Berikut ini contoh thresholding untuk mengubah citra grayscale menjadi 2 warna yaitu hitam dan putih

1.5 UML (Unified Modelling Language)

UML (Unified Modelling Language) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah untuk dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka yang lain [5]. UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, Object Modelling Technique (OMT) dan Object Oriented Software Engineering (OOSE) [8]. Metode Booch dari Grady Booch sangat terkenal dengan nama metode Design Object Oriented. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iteratif, yaitu identifikasi kelas-kelas dan objek- objek, identifikasi semantik dari hubungan objek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada detail kayanya dengan notasi dan elemen

1.5.1 *Use Case Diagram*

Use Case Diagram adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan

sistem disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat biasa dikatakan use case adalah serangkaian scenario yang digabungkan bersama-sama oleh tujuan umum pengguna [6].

1.5.2 Use Case Scenario

Sebuah diagram yang menunjukkan kasus penggunaan dan aktor dapat menjadi titik awal yang bagus, tetapi tidak memberikan cukup detail untuk desainer sistem untuk benar-benar memahami persis bagaimana kekhawatiran sistem akan dipenuhi. *Use Case Skenario* adalah cara terbaik untuk mengungkapkan informasi penting ini dalam bentuk penggunaan *description* setiap kasus berbasis teks harus disertai dengan satu.[6]

1.5.3 *Class Diagram*

Class Diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang kita kembangkan. Diagram kelas memberikan gambaran/diagram statis tentang sistem/perangkat lunak dan relasi-relasi yang ada di dalamnya.[6]

1.5.4 *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan message (pesan) yang diletakkan diantara objek-objek ini dalam use case.[6]

1.5.5 *Activity Diagram*

Activity Diagram digunakan untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya adalah *activity diagram* mendukung perilaku paralel sedangkan *flowchart* tidak bias[6].

2. ISI PENELITIAN

Berikut adalah analisis dan pengujian yang dilakukan dalam penelitian

2.1 Analisis Masalah

Pengenalan objek tersebar yang berada dalam sebuah permainan hidden object, dalam mengenali objek-objek gambar tersebut berdasarkan dari posisi objek yang lebih dari satu objek dan keakuratan dalam pendeteksian. Oleh karena itu dibuat sebuah aplikasi pendeteksian objek tersebar yang akan diterapkan pada simulasi jenis permainan hidden object. Tujuan utamanya adalah untuk mendapatkan solusi dalam mengenali beberapa objek terlepas dari posisi objek berada. Untuk melakukan pendeteksian objek pada jenis permainan hidden object adapun algoritma yang akan digunakan untuk mengenali objek tersebar tersebut adalah *algoritma haar cascade classifier*. Penerapan *Computer Vision* cukup banyak digunakan dalam sistem keamanan maupun proteksi, salah satunya pengenalan objek tersebar [7]. Sistem pendeteksian objek tersebar ini termasuk kedalam perkembangan teknologi yang menjadi penting dalam beberapa bidang seperti sistem keamanan, sistem kontrol [1]. Metode yang digunakan untuk mendeteksi objek tersebar adalah *Haar Cascade classifier*, yaitu metode yang dapat melihat dari adanya objek tersebar dan metode ini memiliki keakuratan yang tinggi dalam mendeteksi suatu objek.

2.2 Analisis Game

Permainan *hidden object* adalah permainan yang dapat diselesaikan dengan cara menemukan semua objek gambar yang tersembunyi dalam papan grid. Untuk menemukan objek gambar yang tersembunyi dalam papan grid maka letak objek yang mungkin terletak secara vertikal dan horizontal.
Permainan *hidden object* terdiri dari satu proses yaitu pencarian objek yang tersembunyi, proses pencarian objek yaitu melakukan pencarian objek pada papan grid yang telah terisi objek-objek yang tersusun secara acak. Berikut ini adalah *flowchart* dari permainan hidden object:



Gambar 2 *Flowchart Game Hidden Object* Secara Umum

2.3 Analisis *Game Hidden Object*

Analisis pada *game hidden object* dilakukan untuk melihat gambaran umum sistem dengan menggunakan algoritma *haar cascade classifier* untuk menemukan objek gambar yang diminta dalam objek gambar yang tersebar. Dalam

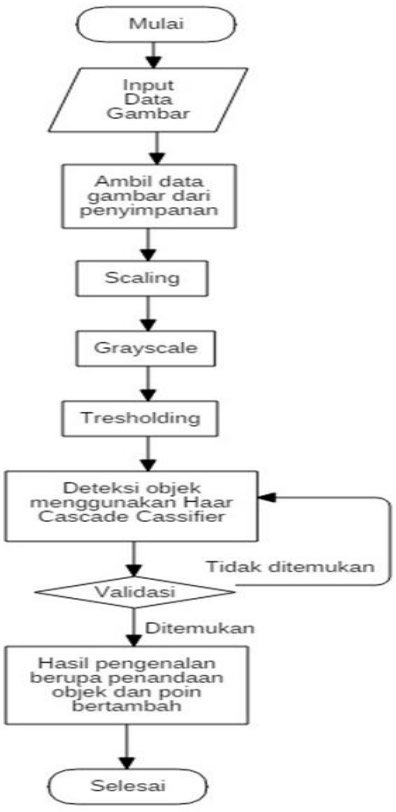
mengimplementasikan algoritma haar cascade classifier akan dibangun sebuah prototype dengan gambaran game hidden object adalah sebagai berikut:

1. Game terdiri dari papan background minimal yaitu 1 background.

2. Objek yang di cari diletakan pada papan background secara random.

3. Pencarian objek hanya dilakukan pada posisi objek rotasi 900 kanan, rotasi 900 kekiri, rotasi 900 bawah, dan posisi tegak lurus.

Dalam permainan *hidden objec*t terdapat beberapa objek-objek gambar yang tersebar secara acak dalam sebuah background yang kemudian akan di deteksi sesuai list gambar yang telah disediakan, ketika objek gambar ditemukan atau terdeteksi maka score akan bertambah 10 point dan ketika objek tidak terdeteksi maka *score* tidak akan bertambah atau tetap.
Dalam membangun sistem pengenalan objek tersebar, dilakukan beberapa tahapan. Tahapan pengenalan objek terdapat pada gambar 3.2:



Gambar 3 *Flowchart Game* Yang Akan Dibangun

Proses pencarian objek dilakukan dengan mencari objek yang akan dicari yang telah di sediakan dalam papan permainan, dalam pencarian objek dilakukan secara berurutan berdasarkan list urutan objek yang akan di deteksi. Proses kecepatan dalam pencarian dilakukan dengan melakukan pencarian secepat mungkin.

2.4 Analisis Data Masukan

Data masukan yang digunakan berupa citra gambar yang di gunakan untuk proses training. Citra yang dimasukan berupa file gambar atau foto dengan format gambar jpg, dimensions 185x150, width 185pixel, height 150 pixel. Setelah citra gambar dimasukan kedalam sistem maka akan dilakukan proses dari tahapan-tahapan yang telah diketahui. Setiap proses pada pengolahan citra menggunakan nilai pixel dari citra untuk perhitungannya. Setiap pixel pada citra terdiri dari 3 unsur warna dasar yaitu Red (R), Green (G) dan Blue (B), yang sering disebut RGB, dan setiap unsur memiliki nilai antara 0-255 yang menjadi kode kedalaman warna masing-masing unsur. Contoh citra warna memiliki nilai R,G,B sebagai berikut:



Gambar 4 Contoh Citra Masukan

2.5 Analisi *Pre Processing*

Pada umumnya, sistem pengenalan objek mengalami beberapa proses sebelum mengenali objek. Setiap prosesnya memiliki peran masing-masing dalam menjalankan proses. Proses-proses yang terlibat dalam pengenalan objek meliputi input gambar, *scaling*, *grayscaling*, dan algoritma *haar cascade classifier* sehingga menghasilkan objek yang di cari.

Dalam analisis pre processing image atau pengenalan pola objek dibagi menjadi 3 tahapan antara lain :

1. *Grayscaling*
2. *Tresholding*

1.Tahapan *Greyscaling*

Dalam tahapan ini dilakukan mengubah citra warna menjadi citra *greyscale*, proses ini dilakukan untuk menyederhanakan model citra. Citra berwarna akan digantikan dengan citra *greyscale* sesusai dengan nilai yang telah didapat dari hasil perhitungan dibawah ini. Contoh untuk menentukan nilai *grayscale* diilustrasikan dengan nilai-nilai R, G, dan B, untuk konversi warna menjadi grayscale digunakan rumus

Grayscale = 0.299*R + 0.587*G + 0.114*B.

Misalkan untuk pixel di koordinat (123,132) perhitungannya sebagai berikut:

Grayscale = (0,299*95) + (0,587*120) + (0,114*28)
= 28,405 + 70,44 + 3,192
= 102,037

Dari perhitugan tersebut didapatkan hasil untuk koordinat (123,123) = 102,037. Perhitungan tersebut dilakukan untuk semua koordinat, maka dari hasil perhitungan grayscale dilakukan rounding (pembulatan) nilai ke paling dekat. Maka hasil pembulatan dari koordinat untuk (123,123) = 102,037 Sehingga didapatkan citra grayscale sebagai berikut:

Tabel 1. Nilai Citra *Grayscale*

	0	1	2	3	4	5	6	7	8	9
0	102	132	169	221	246	252	253	254	254	255
1	86	105	98	114	155	228	248	253	254	255
2	85	104	92	105	108	139	234	248	254	255
3	117	120	109	104	99	123	189	250	254	254
4	103	98	105	103	107	140	236	254	254	254
5	100	104	109	110	131	203	252	252	252	255
6	103	111	113	115	149	247	252	250	254	255
7	113	110	118	135	217	250	253	255	255	254
8	108	113	131	172	244	253	253	255	253	253
9	104	113	139	231	252	251	252	252	254	255

Berikut contoh perubahan citra menjadi citra *grayscale.*



Gambar. 4 Citra RGB Menjadi Citra *Grayscale*

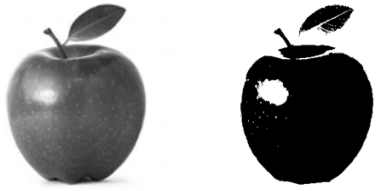
2.Tahapan Tresholding

Selanjutnya adalah tahap thresholding yang digunakan untuk mengubah citra hasil grayscale menjadi citra biner. Dari hasil grayscale pada koordinat (123.123) dengan nilai 102 kemudian dibandingkan dengan nilai T ambang batas, ambang batas yang diambil bernilai 160 karena dengan nilai tersebut kondisi dari citra hitam putih pada threshold terlihat jelas.

Tabel 2 Nilai Citra Grayscale Menjadi CitraBiner

	0	1	2	3	4	5	6	7	8	9
0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1
2	0	0	0	0	0	0	1	1	1	1
3	0	0	0	0	0	0	0	1	1	1
4	0	0	0	0	0	0	0	1	1	1
5	0	0	0	0	0	0	0	1	1	1
6	0	0	0	0	0	0	1	1	1	1
7	0	0	0	0	0	0	1	1	1	1
8	0	0	0	0	0	1	1	1	1	1
9	0	0	0	0	1	1	1	1	1	1

Berikut adalah contoh gambar hasil *Tresholding*:



Gambar 5 Contoh gambar hasil Tresholding

2.6 *Analisis Integral Image*

Integral image merupakan sebuah teknik menghitung fitur-fitur secara cepat dengan mengubah nilai dari setiap pixelnya menjadi sebuah representasi citra yang baru. Adapun cara kerja dari integral image yaitu dengan menjumlahkan nilai-nilai dalam subset pada matrik gambar. Dengan integral image ini, maka perhitungan untuk mendapatkan nilai fitur *Haar* dapat didapatkan dengan waktu yang cepat. Setelah tahap nilai *threshold* dilakukan selanjutnya menghitung nilai *integral image* yang datanya diambil dari nilai threshold kemudian dilakukan proses perhitungan dengan persamaan.

$$s(x,y) = s(x,y)+s(x-1,y)+s(x,y-1)-s(x-1,y-1)$$

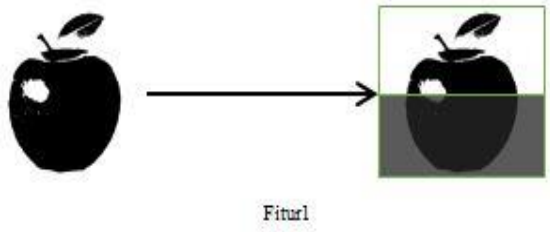
Berikut adalah matrik 150, 185 namun sebagai contoh yang dihitung dengan ukuran 10x10.

Tabel 3. Matrik Biner 150x158

	0	1	2	3	4	5	6	7	8	9
0	467	471	475	479	483	487	491	495	499	503
1	564	568	573	578	583	588	593	598	603	608
2	658	662	667	672	678	684	690	696	702	708
3	750	754	759	764	770	776	782	789	796	803
4	840	844	849	854	860	866	872	879	887	895
5	928	932	937	942	948	954	960	967	976	985
6	1015	1019	1024	1029	1035	1041	1047	1054	1064	1074
7	1100	1104	1109	1114	1120	1126	1132	1140	1151	1162
8	1183	1187	1192	1197	1203	1209	1215	1224	1236	1248
9	1265	1269	1274	1279	1285	1291	1298	1308	1321	1334

2.7 Seleksi Fitur Haar Like Feature

Algoritma tersebut menggunakan metode statistikal dalam melakukan proses pendeteksian objek. Dalam tiap feature dari haar like feature yang didefinisikan pada bentuk dari feature, diantranya koordinat dari dan ukuran feature dari feature tersebut. Feature yang akan ditraining adalah feature yang didapat dari proses feature data masukan. Haar Like fiture digunakan untuk mencari fitur terbaik yang terdapat pada citra yang menunjukan komponen-komponen penting objek. Selanjutnya menentukan jenis haar yang akan diterapkan pada hasil perhitungan sebelumnya. Hal ini dilakukan dengan mengevaluasi setiap fitur terhadap data masukan dengan menggunakan nilai dari fitur tersebut. Fitur yang memiliki batas terbesar antara objek dan bukan bukan objek.



Gambar 6. Menentukan Fitur Haar

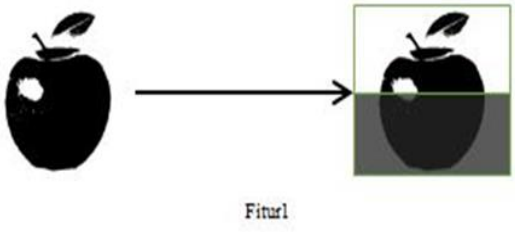
Setelah menentukan fitur *haar* selanjutnya dilakukan perhitungan dari hasil *integral image* yang telah dilakukan, sehingga dapat diketahui nilai *threshold*nya. Berikut perhitungan nilai latih pada fitur 1 terhadap citra uji. Nilai yang didapat fitur hitam A_{hitam} = 91, B_{hitam} = 6905, C_{hitam} = 183, D_{hitam} =9775. Sedangkan nilai fitur putih A_{putih} = 1, B_{putih} = 148, C_{putih} = 90, D_{putih} = 6876.

f(x)=Sumblack rectangle (pixel gray level) – Sumwhite rectangle (pixel gray level)

$=D_{hitam} + A_{hitam} - (B_{hitam} + C_{hitam}))-$
$(D_{putih} + A_{putih} - (B_{putih} + C_{putih}))$
=(9775+91 – (6905+183)) – (6876+1-(148+90))
= 2778–6639
= -3861.

2.8 Seleksi Fitur *Haar Like Feature* Uji

Selanjutnya menentukan jenis haar yang akan diterapkan pada hasil perhitungan sebelumnya. Hal ini dilakukan dengan mengevaluasi setiap fitur terhadap data masukan dengan menggunakan nilai dari fitur tersebut. Fitur yang memiliki batas terbesar antara objek dan bukan objek.



Gambar 7. Menentukan Fitur Haar

Berikut perhitungan nilai latih pada fitur 1 terhadap citra uji. Nilai yang didapat fitur hitam A_{hitam} = 91, B_{hitam} = 6905, C_{hitam} = 183, D_{hitam} =9775. Sedangkan nilai fitur putih A_{putih} = 1, B_{putih} = 148, C_{putih} = 90, D_{putih} = 6876.

f(x)=Sumblack rectangle (pixel gray level) – Sumwhite rectangle (pixel gray level)

$=D_{hitam} + A_{hitam} - (B_{hitam} + C_{hitam}))-$
$(D_{putih} + A_{putih} - (B_{putih} + C_{putih}))$

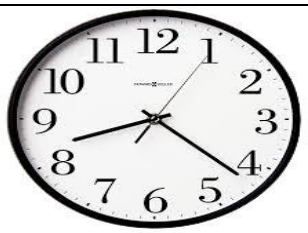
=(9775+91 – (6905+183)) – (6876+1-(148+90))
= 2778–6639
= -3861.

Maka setelah didapatkannya nilai threshold dari citra uji, maka nilai dari hasil citra latih dan citra uji memiliki persamaan hasil itu disebabkan dikarenakan citra uji dan citra latih sama.
Pada citra latih dengan fitur 1 menghasilkan nilai = -3861, sedangkan pada citra uji dengan fitur 1 memiliki nilai =-3861. Maka dari itu dapat disimpulkan hal tersebut sebuah objek yang ingin dideteksi dikarenakan terjadinya persamaan nilai threshold pada citra latih dan citra uji.

2.9 Proses Pengujian

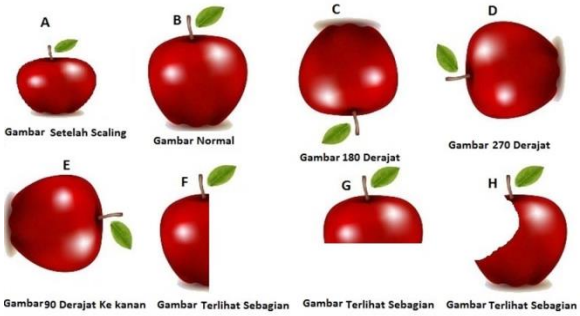
Pada pengujian ini dilakukan untuk menguji apakah objek yang ingin dideteksi sesuai dengan yang diharapkan atau tidak, jika objek tidak ada di list data yang akan dideteksi maka sistem tidak akan melakuan pendeteksian pada objek tersebar.
Berikut adalah tabel data objek gambar :

Tabel 4. Data Objek Gambar

Subjek	Nama Objek	Gambar Objek
1	Jam	
2	Mobil	
3	Apel	
4	Orang	

5	Mata	

Pengujian pada tahap ini dilakukan dengan beberapa variasi bentuk objek horijontal, vertikal dan terlihat sebagian seperti ditunjukkan seperti gambar berikut :



Gambar 8. Bentuk Objek

Berikut adalah tabel pendeteksian akurasi objek dengan objek yang berbeda, Pada pengujian ini dilakukan untuk menguji apakah objek yang ingin dideteksi sesuai dengan yang diharapkan atau tidak, jika objek tidak ada di list data yang akan dideteksi maka sistem tidak akan melakuan pendeteksian pada objek tersebar.

Tabel 5. Akurasi Deteksi Objek

	Subyek	Bentuk objek	Subyek Predicted				
			1	2	3	4	5
Actual	1	A		-	-	-	-
		B		-	-	-	-
		C		-	-	-	-
		D		-	-	-	-
		E	*	-	-	-	-
		F		-	-	-	-
		G		-	-	-	-
		H		-	-	-	-
	2	A	-		-	-	-
		B	-		-	-	-
		C	-		-	-	-
		D	-		-	-	-
		E	-		-	-	-
		F	-		-	-	-
		G	-		-	-	-
		H	-		-	-	-
	3	A					
		B					
		C					
		D					
		E					
		F					
		G					
		H					
	4	A	-	-	-		-
		B	-	-	-		-
		C	-	-	-		-
		D	-	-	-		-
		E	-	-	-		-
		F	-	-	-		-
		G	-	-	-		-
		H	-	-	-		-
	5	A	-	-	-	-	
		B	-	-	-	-	
		C	-	-	-	-	
		D	-	-	-	-	
		E	-	-	-	-	
		F	-	-	-	-	
		G	-	-	-	-	
		H	-	-	-	-	

Keterangan :
Tanda "√" = TP
Tanda "-" = FN
Tanda "*" = FP

TP (true positif) = Ketika objek yang terdeteksi sesuai dengan yang dideteksi
TN (true negatif) = Ketika objek yang tidak ingin dideteksi tidak terdeteksi
FP (false positif) = Ketika objek yang dideteksi tidak terdeteksi
FN (false negatif)= Ketika objek yang terdeteksi tidak sesuai yang dideteksi

Tabel 6. Nilai perhitungan setiap subyek

Subyek 1	**Subyek 2**	**Subyek 3**	**Subyek 4**	**Subyek 5**
Total Pengujian =40	Total Pengujian =40	Total Pengujian =40	Total Pengujian =40	Total Pengujian =40
TP = 7	TP = 8	TP = 8	TP = 8	TP = 8
FN = 0	FN = 0	FN = 32	FN = 0	FN = 0
FP = 1	FP = 0	FP = 0	FP = 0	TP = 0
TN = 32	TN =32	TN = 0	TN =32	FN =32
Total	Total TP= 39	Total FN = 32	Total FP = 1	Total TN = 128
	200			

Hasilnya : $\frac{TP+TN}{TP+FP+TN+FN} \times 100\%$

$$= \frac{(7+8+8+8+8)+(32+32+0+32+32)}{39+32+1+128}$$

$$= \frac{39+128}{200}$$

$$= \frac{167}{200} \times 100\% = 83{,}5\ \%$$

Dari Pengujian diatas dapat dianalisa dan disimpulkan bahwa pendeteksian tidak berpengaruh

pada jumlah perbandingan dengan beberapa objek yang berbeda, hanya saja berpengaruh pada objek yag mirip dengan objek yang akan dideteksi saja dan ketika objek terlihat sebagian. Maka untuk mendapatkan tingkat keakurasian pengenalan objek, perhitungan akan dilakukan pada banyak objek tersebar yang terdeteksi dan terjadi pengenalan. Sehingga didapat presentasi keberhasilan dalam mengenali subjek sebesar (167/200)*100% = 83,5%.

3. PENUTUP

Berdasarkan penelitian mengenai implementasi algoritma Haar cascade classifier untuk mengenali objek tersebar maka dapat dibuat kesimpulan:

1. Algoritma *haar cascade classifier* dapat digunakan untuk mengenali objek tersebar pada permainan hidden object.

2. Aplikasi pendeteksi objek tersebar pada *game hidden object* menggunakan metode Haar Cascade Classifier, dan dapat mendeteksi objek tersebar dengan pengujian performansi berupa objek posisi normal, 90 derajat, 180 derajat, 270 derajat dan terlihat sebagian memiliki nilai 83,5% terdeteksi.

Adapun saran yang dapat diberikan untuk pengembangan selanjutnya antara lain adalah sebagai berikut:

1. Untuk meningkatkan akurasi dalam pendeteksian segala jenis objek diperlukan sampel citra positif dan negatif lebih banyak lagi, khususnya untuk segala jenis tipe dan bentuk dari posisi objek.

2. Penelitian kali ini hanya berupa simulasi program, maka diharapkan dalam penelitian selanjutnya dapat diimplementasikan langsung pada game hidden object yang sebenarnya, sehingga aplikasi tersebut bisa lebih bermanfaat.

4. DAFTAR PUSTAKA

[1] Hadi Santoso dan Agus Harjoko.2013. Haar Cascade Classifier Dan Algoritma Adaboost Untuk Deteksi Banyak Wajah Dalam Ruangan Kelas. Jurnal Teknologi,vol. 6, No. 2, pp. 108-115.

[2] Paul Viola and Michaels J. Jones. (2001). Rapid Jones, Rapid Object Detection using Detection using boosted Cascade of Simple Features. IEEE Features. IEEE CVPR .

[3] Hermawati, F. A.2013. Pengolahan Citra Digital Konsep & Teori.Yogyakarta: Andi.

[4] , R. (n.d.). Menentukan Thresholding. Retrieved from http://rubik.web.id

[5] Gunter, G., & Kenny, R. 2008. Digital booktalk: Digital media for reluctant readers. Contemporary Issues in Technology and Teacher Education.

[6] R. S. Pressman, Rekayasa Perangkat Lunak Pendekatan Praktis, Yogyakarta: Andi, 2012.

[7] Fadilsyah. 2007. Computer Vision dan Pengolahan Citra.Yogyakarta. Penerbit Andi.

[8] Martin, R.C.2002.UML for Java Programmers. United States of American: Alan Apt

[9] Muhadjir,N.1998. Metodelogi Penelitian Kualitatif. Yogyakarta: Rake Sarasin.

[10] Suyanto.2014. Artificial Intellegence Searching, Reasoning, Planning dan Learning, Bandung: Informatika Bandung

[11] Chellappa Rama, Theodoridis Sergio. 2014. Academic Press Library In Signal Processing. USA : Academic Press

[12] P.P. Caffier, U. Erdmann and P. Ullsperger, 2005. The spontaneous eye-blink as sleepiness indicator in patients with obstructive sleep apnoea syndrome - a pilot study. Sleep Medicin.

[13] Saha, S., Basu, S., Nasipuri, M., & Kumar Basu, D. (2009). DEVELOPMENT OF AN AUTOMATED RED LIGHT VIOLATION DETECTION SYSTEM FOR INDIAN VEHICLES. IEEE NATIONAL CONFERENCE COCOSYS, 52-64.