

Pengenalan Wajah Menggunakan *Raspberry Pi* Dengan Menggunakan Metode *Real-Time Content Based Sample Image Matching System*
FACE RECOGNITION USING RASPBERRY PI AND REAL-TIME CONTENT BASED SAMPLE IMAGE MATCHING SYSTEM METHOD

Angga Gita Purnama, Ir. Agus Virgono, M.T.², Andrew Brian Osmond S.T., M.T.³

^{1,2,3}Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom
 Jl. Telekomunikasi No 1 Terusan Buah Batu Bandung 40257 Indonesia
¹anggapurnama16@gmail.com, ²avirgono@telkomuniversity.ac.id,
³abosmond@telkomuniversity.ac.id

ABSTRAK

Face recognition adalah sebuah teknologi yang digunakan dalam berbagai aplikasi dengan mengidentifikasi wajah manusia menggunakan gambar digital. Pada penelitian ini *Face recognition* dijalankan menggunakan raspberry pi. Raspberry pi akan digunakan sebagai *operating system* untuk menjalankan programnya. Pada penelitian kali ini akan menggunakan *camera modul* dari raspberry. Untuk Bahasa pemrograman menggunakan Bahasa python. Metode yang digunakan adalah *realtime content based sample image matching system* yang efisien untuk pengolahan gambar. Metode ini digunakan karena metode ini menggunakan Orb yang lebih cepat untuk mendapatkan *keypoint* dibandingkan dengan SIFT dan SURF. Pada metode *realtime content based sample image matching system* terdiri dari dua algoritma algoritma orb dan *Histogram Intersection*. Algoritma Orb digunakan untuk melakukan pencocokan secara cepat dan dicocokkan lagi menggunakan *Histogram Intersection*. Pada penelitian ini akan metode *realtime content based sample image matching system* diuji coba mencocokkan wajah dengan berbagai kondisi, seperti pengaruh resolusi, wajah, cahaya, dan jarak. Pada penelitian ini dapat mencocokkan hingga 98% pada resolusi 640x480 dengan batas *keypoint* diatas 20 dengan *threshold* 0.7.

Kata kunci : *Raspberry pi, face recognition, content based sample image matching system, Orb*

ABSTRACT

Face recognition is a technology that is used in various applications to identify human faces using digital images . Face recognition is executed using the raspberry pi . Raspberry pi will be used as the operating system to run the program . In this research will use a camera module of a raspberry . To use the language python programming language . The method used is realtime content based sample image matching system that is efficient for image processing. This method is used because this method uses Orb faster to get *keypoint* compared with SIFT and SURF. In the method of realtime content based sample image matching system consists of two algorithms algorithms Orb and Histogram Intersection. Orb algorithms used to perform fast matching and matched again using Histogram Intersection. In this research, the method of realtime content based sample image matching system tested matching faces with various conditions, such as the effect of the resolution, different faces, light, and distance. In this study can be matched up to 98% at a resolution of 640x480 with a limit above the 20 threshold *keypoint* 0.7.

Keyword : *Raspberry pi, face recognition, content based sample image matching system, Orb*

1. Pendahuluan

Teknologi komputer telah menjadi bagian dari kehidupan sehari-hari sejak lama. Pengaplikasian teknologi komputer yang semakin banyak di tahun-tahun mendatang akan menjadikan peran komputer semakin penting dan bisa jadi krusial. Menurut Pada penelitian kali ini menggunakan mini komputer yaitu raspberry pi. Ash Stone (2012), Raspberry Pi adalah sebuah mini komputer yang ukurannya sama dengan credit card yang dapat digunakan untuk banyak hal seperti yang komputer bisa lakukan, seperti spreadsheets, word processing, permainan, dan juga pemrograman. Raspberry Pi juga bisa digunakan untuk menangkap gambar dengan camera module dan memproses untuk mengenali wajah manusia.

Tujuan pada penelitian ini mencocokkan objek wajah menggunakan metode real time content based sample image matching system untuk digunakan dalam beberapa proses sebagai bukti identitas autentik seseorang berdasarkan ciri-ciri yang sesuai dengan citra wajah secara komputerisasi. Untuk lebih memfokuskan permasalahan yang akan diteliti, maka permasalahannya dibatasi. Pertama, pencocokan wajah digunakan pada dua image. Kedua, menggunakan objek citra wajah. Ketiga, uji coba yang digunakan adalah image wajah yang sama dan image wajah yang berbeda. Keempat, pengujian dilakukan dengan latar belakang polos. Terakhir, hasil matching dengan berbagai kondisi.

Pada sistem pengenalan wajah ini menggunakan metode yang digunakan adalah metode Real-Time Content Based Sample Image Matching System yang diharapkan dapat mempercepat prosesnya dengan hasil yang memuaskan selain itu diharapkan dapat berfungsi dengan optimal pada raspberry pi.

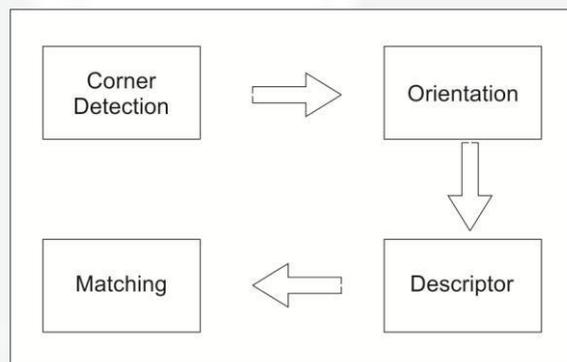
2. Landasan Teori

2.1 Realtime Content Based Sample Image Matching System

Real-Time Content Based Sample Image Matching^[1] adalah metode untuk menentukan apakah gambar tersebut ada didalam database atau tidak. Pada tulisan ini hanya akan digunakan untuk pencocokan wajah, meskipun cara pengambilan lain bisa dilakukan, biasanya akan terkendala oleh masalah komputasi yang tinggi. Oleh karena itu digunakan metode ini karena menggunakan algoritma Orb untuk mempercepat prosesnya.

2.2 Fast Image Matching Components

Algoritma ORB^[2] adalah algoritma yang paling efisien untuk pengolahan gambar . Ini merupakan metode yang efektif untuk deteksi gumpalan . Algoritma ORB menggunakan FAST (Features from Accelerated Segment Test) dan BRIEF (Binary Robust Independent Elementary Features) algoritma .



Keunggulan dari algoritma ORB dalam fitur mendeteksi titik dan membuat descriptors secara cepat. Namun, titik lemah adalah bahwa mengkonsumsi banyak waktu selama pencocokan video real-time. Tujuan dari penelitian ini adalah untuk meningkatkan kecepatan pencocokan citra algoritma ORB dan juga untuk meningkatkan algoritma pencocokan. Algoritma ini juga akan diuji dalam melakukan pengenalan wajah. Selain itu, mengingat BRIEF yang tidak tampil baik dengan rotasi, ORB merupakan sebuah versi BRIEF dengan menggunakan orientasi keypoint.

ORB menggunakan pengukuran sederhana dengan orientasi sudut - pusat massa intensitas [2]. Pertama, patch pada orb didefinisikan sebagai:

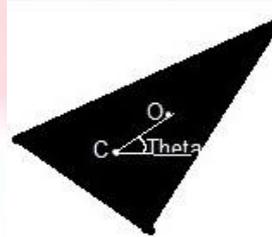
$$\Sigma$$

Dengan ini dapat menemukan pusat massa, "pusat massa" dari patch sebagai:

$$\left(\frac{\sum x}{\sum 1} \right)$$

Kita dapat membuat vektor dari pusat sudut O, ke -oc massa. Orientasi patch kemudian diberikan oleh:

Berikut adalah ilustrasi untuk membantu menjelaskan metode:



Setelah menghitung orientasi patch, kita dapat memutar untuk rotasi kanonik dan kemudian menghitung deskriptor, sehingga mendapatkan beberapa rotasi invariant. Berikut flow chart dari algoritma orb.



2.3 Fast matching decision

Fast matching decision^[1] digunakan setelah proyeksi untuk memperoleh representasi vektor dari gambar yang memutuskan apakah query terdaftar di database sesuai dengan kesamaan antara H1 dan H2 dengan menggunakan *Histogram intersection similarity* untuk menghitung kesamaan antara gambar.

Histogram adalah representasi grafis dari distribusi intensitas dari suatu gambar yang mengkuantifikasi jumlah piksel untuk setiap nilai intensitas. Histogram sendiri biasanya terbagi dengan beberapa *bins* untuk memberikan batasan setiap nilai intensitasnya. Berikut adalah contoh matrix yang berisi informasi dari gambar (intensitas dalam kisaran 0-255).

234	143	209	176	190	229	177	230	192	0	239	142	139	04	0	03	26	0	89	03
27	88	231	75	141	107	148	210	13	238	141	05	08	242	110	208	244	0	33	88
94	42	17	215	239	234	37	41	86	189	55	233	232	37	122	209	76	110	100	100
9	180	192	71	101	100	08	123	07	232	10	147	174	1	143	211	176	188	192	68
179	20	238	192	190	132	44	240	22	134	83	133	110	234	176	238	188	234	51	204
232	25	0	103	174	130	01	10	110	189	0	173	197	183	152	53	52	87	68	118
235	35	151	185	129	81	238	170	105	94	30	21	07	101	08	37	100	149	52	154
135	242	54	0	104	100	189	47	138	254	225	154	31	181	131	15	128	25	232	205
223	114	79	129	147	0	201	68	89	107	58	44	233	84	38	1	62	0	231	218
55	188	237	186	80	101	131	241	60	133	124	151	112	29	150	4	240	78	117	145
132	152	229	78	90	217	219	105	119	77	39	49	2	9	214	181	205	116	136	23
182	94	178	199	20	140	37	223	232	112	92	45	172	15	91	119	190	119	208	61
224	110	154	172	75	59	69	180	187	185	41	44	0	170	157	121	131	31	29	112
236	83	36	7	83	69	173	183	86	237	87	227	19	218	248	237	75	182	201	146
98	185	224	207	140	22	31	118	234	34	182	118	23	47	68	242	189	152	116	248
140	37	101	230	248	145	122	04	27	56	239	1	225	143	91	159	98	90	40	190
231	1	178	139	121	93	97	174	249	182	77	115	223	188	182	32	65	232	83	196
179	180	223	230	87	102	149	78	176	19	17	4	184	176	163	102	93	81	132	206
173	137	185	242	181	181	214	49	74	238	197	97	88	102	15	217	148	0	102	188
85	0	17	222	18	210	70	21	26	241	184	218	83	83	208	102	153	212	118	47

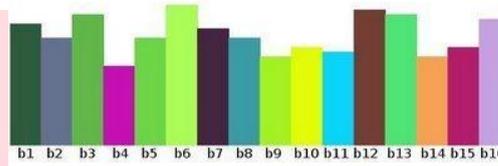
Gambar 2.3.1 Contoh representasi pixel

Untuk nilai informasi untuk kasus ini adalah 256 nilai, maka segmen jangkauan di subparts (*bins*) seperti:

$$[0, 255] = [0, 15] \cup [16, 31] \cup \dots \cup [240, 255]$$

$$\text{range} = \text{bin}_1 \cup \text{bin}_2 \cup \dots \cup \text{bin}_{n=15}$$

Dari penjelasan diatas maka dapat menghitung jumlah piksel pada setiap $\text{bin}_{\{i\}}$. Dari contoh diatas maka data yang didapat adalah (sumbu x mewakili *bins* dan sumbu y jumlah piksel).



Gambar 2.3.2 Contoh pengelompokan data pada histogram

Diagram diatas merupakan contoh sederhana bagaimana sebuah histogram bekerja dan mengapa hal ini berguna. Histogram dapat menghitung tidak hanya intensitas warna, tapi gambar apapun yang ingin kita mengukur.

Setelah mendapatkan nilai histogram maka tahap selanjutnya akan membandingkan dua nilai histogram. Pada penelitian ini akan menggunakan *histogram intersection* untuk membandingkan nilai histogramnya.

Histogram intersection similarity (Hist), yang didefinisikan sebagai berikut

$$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I))$$

Pada Histogram Intersection Similarity semakin besar nilainya maka kedua gambar tersebut semakin cocok.

3. Perancangan Sistem

3.1 Gambaran Umum Sistem

Pada tugas akhir ini memanfaatkan raspberry pi dengan menggunakan modul kamera ekstensi. Kamera yang digunakan adalah kamera modul raspberry pi. Pada sistem ini masukan yang di dapat melalui kamera yang akan menangkap gambar wajah. Algoritma akan di tanamkan di raspberry pi. Untuk pencocokan gambar yang digunakan metode *Real-Time Content Based Sample Image Matching System*.



kamera adalah sebagai masukan sistem tersebut. Kamera digunakan untuk menangkap gambar. Setelah gambar didapat maka proses selanjutnya adalah gambar tersebut diolah dan dicocokkan menggunakan *Real-Time Content Based Sample Image Matching System* pada *raspberry pi*. Setelah dilakukan pencocokan maka raspberry pi akan memberikan hasil keluaran yang akan di tampilkan pada monitor.

3.2 Diagram Alir Perancangan Umum Sistem

Pada diagram alir tersebut citra akan didapat dari kamera lalu akan di simpan langsung melalui *raspberry pi*

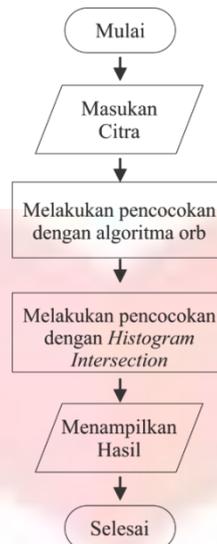


Diagram alir proses mencocokkan citra ke database

Dari gambar tersebut masukan citra bisa dari kamera langsung maupun citra yang sudah ada. Setelah sampel citra didapat maka tahap selanjutnya mencocokkan citra tersebut menggunakan algoritma orb dengan citra pada database. Setelah di temukan yang cocok maka proses selanjutnya pencocokan dengan menggunakan *Histogram Intersection*. Setelah pencocokan selesai akan ditampilkan citra yang paling cocok dengan database dan juga hasil dari proses kedua algoritma tersebut.

3.3 Deskripsi Sistem Pengenalan Wajah

Tugas utama yang dilakukan oleh sistem pada penelitian ini melakukan proses pencocokan citra wajah dengan menggunakan *Real-Time Content Based Sample Image Matching System*.

3.4 Analisa Kebutuhan Sistem

Sistem yang dibangun dalam penelitian ini diimplementasikan untuk mengetahui apakah metode *Real-Time Content Based Sample Image Matching System* dapat mencocokkan citra masukan dengan citra pada database.

3.5 Akurasi Sistem

Akurasi merupakan ukuran ketepatan sistem dalam mengenali masukan yang diberikan sehingga menghasilkan keluaran yang benar. Rumus akurasi yang dapat dituliskan sebagai berikut.

$$\text{Akurasi} = \frac{\text{Jumlah Benar}}{\text{Jumlah Total}} \times 100 \%$$

4. Pengujian dan Analisis

4.1 Spesifikasi Sistem

4.1.1 Perangkat Keras, Raspberry Pi 2 Model B (Raspberry P. , 2016)

1. Broadcom BCM2837 Arm7 Quad Core Processor 900MHz
2. 1GB RAM
3. 16GB micro sd
4. CSI camera port for connecting Camera
5. Raspberry Camera Module 5 Megapixels

4.1.2 Perangkat Lunak

1. Operating System, Raspbian Jessie
2. Python 2.7
3. Library OpenCV 3.1

4.2 Spesifikasi Percobaan Pengujian

4.2.1 Pengujian Resolusi

Percobaan pengujian yang pertama dilakukan dengan menggunakan beberapa resolusi citra yaitu 1024x768, 800x600, 640x480, dan 320x240 untuk mengetahui pengaruh resolusi terhadap kecepatan proses dan juga akurasi. Pada pengujian ini menggunakan wajah pada satu orang yang sama dengan batas maksimal 500 *keypoint* untuk setiap gambar.

Dengan melakukan percobaan tersebut maka didapat hasil sebagai berikut, pada resolusi 1024x768 mendapat tingkat kecocokan dengan rata-rata 34%, pada resolusi 800x600 mendapatkan tingkat kecocokan dengan rata-rata 75%, pada resolusi 640x480 mendapatkan tingkat kecocokan dengan rata-rata 98% dan pada resolusi 320x240 mendapatkan tingkat kecocokan rata-rata 72%. Dengan data tersebut maka resolusi yang mendapat akurasi paling baik pada resolusi 640x480, karena pada resolusi tersebut mendapat jumlah kecocokan yang paling besar dengan rata-rata 98% dengan batas keypoint lebih dari 20. Untuk lama prosesnya pada resolusi 1024x768 mendapatkan rata-rata waktu 26.77 detik, pada resolusi 800x600 mendapatkan rata-rata waktu 21.23 detik, pada resolusi 640x480 mendapatkan rata-rata waktu 17.57 detik, dan pada resolusi 320x240 mendapatkan waktu rata-rata 12.37.

4.3 Pengujian Beda Muka

Pada pengujian ini dilakukan dengan cara memasukan sample muka orang yang berbeda-beda. Tujuan percobaan ini yaitu untuk mengetahui apakah algoritma ini dapat membedakan wajah. Pada pengujian ini digunakan citra dengan resolusi 640x480 dengan hasil sebagai berikut :

Image sample	Image Template	Keypoint Match	Keypoint Image >20	Persentase kecocokan
X	A	86	Matching	100%
	B	72	Matching	
	C	74	Matching	
	D	41	Matching	
	E	66	Matching	
Total Matching			5/5	
Y	A	0	No Matching	0%
	B	5	No Matching	
	C	4	No Matching	
	D	2	No Matching	
	E	2	No Matching	
Total Matching			0/5	
Z	A	3	No Matching	0%
	B	3	No Matching	
	C	2	No Matching	
	D	2	No Matching	
	E	3	No Matching	
Total Matching			0/5	

Tabel 4.3 beda muka

*Keterangan

X = sample wajah orang yang sama dengan template

Y,Z = sample wajah orang yang berbeda dengan template

Hasil dari pengujian ini sample "X" adalah orang yang sama dengan orang pada template, hasilnya dari 5 kali hasil pencocokan semuanya diatas 20 keypoint. Sedangkan untuk "Y" dan "Z" adalah orang yang berbeda, hasilnya "Y" dan "X" hasil pencocokan tidak ada yang melebihi 20 keypoint.

4.4 Pengujian Tingkat Kecerahan

Pada pengujian ini terdapat 5 sample wajah yang sama dengan berbeda-beda tingkat kecerahan dan akan dicocokkan dengan 5 wajah dengan pencahayaan kondisi normal, dengan hasil pada tabel sebagai berikut :

Image sample	Image Template	Keypoint Match	Keypoint Image >20	Persentase Kecocokan
"A" Dengan Kecerahan 100%	A	395	Matching	100%
	B	337	Matching	
	C	46	Matching	
	D	68	Matching	
	E	36	Matching	
Total Matching			5/5	
"A" Dengan	A	60	Matching	60%

Kecerahan 80%	B	57	Matching	
	C	16	No Matching	
	D	21	Matching	
	E	13	No Matching	
Total Matching			3/5	
"A" Dengan Kecerahan 60%	A	4	No Matching	0%
	B	3	No Matching	
	C	5	No Matching	
	D	2	No Matching	
	E	1	No Matching	
Total Matching			0/5	
"A" Dengan Kecerahan 40%	A	0	No Matching	0%
	B	0	No Matching	
	C	0	No Matching	
	D	0	No Matching	
	E	0	No Matching	
Total Matching			0/5	
"A" Dengan Kecerahan 20%	A	0	No Matching	0%
	B	0	No Matching	
	C	0	No Matching	
	D	0	No Matching	
	E	0	No Matching	
Total Matching			0/5	

Tabel 4.4 Tabel kecerahan

4.5 Pengujian Beda Jarak

Pada pengujian ini akan dilakukan pencocokan dengan objek dengan jarak sample image yang berbeda-beda dan akan dicocokkan dengan template dengan jarak wajah 90cm. Pada pengujian ini sample yang digunakan mulai dari jarak 90cm, 120cm,150cm,180cm dan dicocokkan dengan batas keypoint lebih dari 20. Berikut adalah hasil pengujian :

Image Sample A dapat mencocokkan lebih dari 20 keypoint pada kelima template. Sedangkan hasil pencocokan dengan sampel jarak 120cm,150cm, dan 180cm hasilnya tidak ada keypoint yang melebihi 20 keypoint.

4.6 Pengujian histogram intersection

Pada pengujian ini sebuah sample yang dicocokkan dengan tiga template yang memiliki kondisi Keypoint Match yang hampir sama. Maka yang menjadi patokan pencocokan menggunakan *histogram intersection*. Pada *histogram intersection* nilai yang paling besarlah yang paling cocok antara *sample* dan *template*. Berikut adalah tabel pengujiannya :

Image sample	Image Template	Keypoint Match	Histogram Intersection
A	a 	5	2,53
	b 	4	2,34
	c 	4	1,53

Tabel 4.6 *Histogram Intersection*

Dari tabel diatas nilai *Histogram intersection* yang paling besar adalah sample "A" yang dicocokkan dengan template "a" dengan nilai 2,53. Maka dari hasil tersebut algoritma Histogram intersection dapat membantu mencocokkan wajah.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan pengenalan wajah dengan metode *Realtime Content Based Sample Image Matching System* yang menggunakan raspberry pi adalah sebagai berikut:

1. Hasil dari pengujian menunjukkan bahwa, metode *realtime content based sample image matching system* jika dijalankan pada raspberry pi dapat mencapai akurasi yang baik mencapai 98% dengan batas keypoint diatas 20 pada resolusi 640x480.
2. Untuk proses yang paling cepat menggunakan resolusi 320x240 dengan rata-rata waktu 12.37 detik.
3. Pada metode *realtime content based sample image matching system* ini juga berhasil membedakan wajah, karena saat memasukan wajah yang berbeda keypoint yang didapat dibawah 20 keypoint.
4. Hasil dari pengujian tingkat kecerahan bahwa metode ini masih bisa menemukan kecocokan dengan tingkat kecerahan 80% dengan tingkat akurasi 60% dengan batas keypoint diatas 20. Sedangkan untuk kecerahan dibawah 80%.
5. Dari hasil pengujian jarak metode *realtime content based sample image matching system* dapat mencocokkan dengan baik apabila jarak sample dan jarak pada template sama, apabila jarak wajah berbeda maka metode ini tidak dapat mencocokkan dengan baik keypoint yang didapat tidak melebihi 20 keypoint.
6. Algoritma Histogram Intersection dapat digunakan untuk membedakan wajah apabila orb mendapatkan *keypoint* yang hampir sama. Karena algoritma Histogram Intersection mendapatkan nilai yang paling besar saat dicocokkan dengan wajah orang yang sama.

5.2 Saran

Untuk mendapatkan pengembangan penelitian lebih lanjut penulis memberikan saran, antara lain :

1. Untuk meningkatkan tingkat akurasi maka template harus lebih banyak dan dengan berbagai kondisi.
2. Untuk mempercepat proses pencocokan maka bisa digunakan Embedded Board dengan spesifikasi yang lebih tinggi

DAFTAR PUSTAKA

- [1] L.Yong, T.Jinhui, L. Yangxi, dan X.Chao , "FIM: A Real-Time Content Based Sample Image Matching System," in Seventh International Conference on Image and Graphics, 2013
- [2] E. Rublee, V. Rabaud, K. Konolige, dan G. Bradski, "Orb: an efficient alternative to sift or surf," in *IEEE International Conference on Computer Vision*, 2011
- [3] K.Hari, T.A.B Wirayuda, dan R.N Dayawanti (2010) ANALISIS DAN IMPLEMENTASI FACE DETECTION MENGGUNAKAN METODE VIOLA JONES
- [4] V. Ramanath (2015). Implementation of Improved Face Recognition Technique for Car Ignition Access Control Using" in Raspberry Pi Microcontroller
- [5] M. Rahayu, A. Budi, E. Haritnam (2014). PENGONTROLAN ALAT ELEKTRONIKA MELALUI MEDIA WI-FI BERBASIS RASPBERRY PI.
- [6] Y.D. Kim, J.T Park, I.Y Moon and C.H. Oh (2014). Performance Analysis of ORB Image Matching Based on Android
- [7] Sepritahara (2012). SISTEM PENGENALAN WAJAH (FACE RECOGNITION) MENGGUNAKAN METODE HIDDEN MARKOV MODEL (HMM)
- [8] N.P. Pradnyawati, H. Aryadita, E.M. Adams (2013). Pengembangan Sistem Presensi Mahasiswa Berbasis Face Recognition Dengan Menggunakan Algoritma Eigenface.
- [9] Koeshardianto Meidya (2014) Pencocokan Obyek Wajah Menggunakan Metode SIFT (SCALE INVARIANT FEATURE TRANSFORM)