

Tingkat Akurasi Pengenalan Isyarat Tuna Rungu Menggunakan SVM dengan Fitur Lokal ORIENTED FAST AND ROTATED BRIEF “ORB” dan ADAPTIVE AND GENERIC ACCELERATED SEGMENT TEST “AGAST”

Muhammad Ozzy Calvin Rendy¹, Arief Setyanto, Amir Fatah Sofyan^{3*}

Teknik Informatika UNIVERSITAS AMIKOM YOGYAKARTA

Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta 55283

[¹muhammad.16@students.amikom.ac.id](mailto:muhammad.16@students.amikom.ac.id), [²arief_s@amikom.ac.id](mailto:arief_s@amikom.ac.id), [³amir@amikom.ac.id](mailto:amir@amikom.ac.id)

INTISARI

Perkembangan teknologi berbasis komputer visual yang berkembang sangat pesat dalam memenuhi kebutuhan era teknologi seperti saat ini. Teknologi komputer visual yang dapat membantu dalam berbagai bidang seperti tracking plat nomor kendaraan, pengenalan wajah, tracking wajah, dan lain – lain. Yang dimana teknologi komputer vision membantu dalam berbagai bidang seperti halnya gerakan isyarat tuna rungu yang harus di perhatikan lebih yang dikarenakan tidak semua orang mengetahui gerakan gerakan Bahasa isyarat tuna rungu dan penelitian ini meneliti tentang performa akurasi algoritma fitur lokal ORIENTED FAST AND ROTATED BRIEF “ORB” dan ADAPTIVE AND GENERIC ACCELERATED SEGMENT TEST “AGAST” dalam mentracking gerakan isyarat tuna rungu. Hasil dari penelitian ini akan memaparkan perbandingan akurasi dari algoritma ORB dan AGAST yang bertujuan agar teknologi berbasis komputer visual dapat berkembang lebih jauh lagi dengan adanya penelitian ini.

Kata kunci— ORB, AGAST, gerakan isyarat tuna rungu, komputer vision, fitur lokal..

ABSTRACT

The development of visual computer-based technology that develops very rapidly in fulfilling the needs of today's technological era. Visual computer technology that can help in various fields such as vehicle license plate tracking, face recognition, face tracking, etc. Which is where computer vision technology helps in various areas such as the deaf signal movement which should be noticed more that will not everyone know the movement of the Deaf sign language movements and this research examines the accuracy of the algorithm performance of feature detector ORIENTED FAST AND ROTATED BRIEF "ORB" and ADAPTIVE and GENERIC ACCELERATED SEGMENT TEST "AGAST" in tracking the movement of deaf signals. The results of this study will expose the accuracy comparison of the ORB and AGAST algorithms aimed at visual computer-based technologies to further evolve with the research.

Kata kunci— ORB, AGAST, American Sign Language, computer vision, feature detector .

I. PENDAHULUAN

Feature Descriptors atau biasa disebut Descriptors(C. M. Wang 2015), telah menjadi tantangan yang sulit dan juga menjadi topik pembicaraan yang populer di kalangan peneliti(Mair et al. 2010a; Simo-Serra et al. 2015)Umumnya Descriptors digunakan pada deteksi target (Yansheng Li and Zhang 2018), kontrol otomatis (Madero Orozco et al. 2014), inspeksi produk (Weimer, Scholz-Reiter, and Shpitalni 2016), identifikasi wajah (Ding et al. 2016), pemindai wajah (Parkhi, Vedaldi, and Zisserman 2015), dan masih banyak lain nya. yang jika dikembangkan akan menimbulkan

masalah baru yang butuh untuk di selesaikan, termasuk perubahan sudut pandang (Jain, puneet. Manweiler, Justin. and. Choudhury 2015), skala (Marchand, Uchiyama, and Spindler 2016), dan perputaran rotasi (Fernandes, Wang, and Simons 2015) pada satu ataupun banyak objek, dan semua itu membutuhkan algoritma yang efektif untuk menyelesaikan masalahnya masing-masing (C. M. Wang 2015).

Pada pengimplementasiannya, algoritma pengolah gambar yang rumit umumnya digunakan untuk mendeteksi fitur dalam gambar. Fitur yang diolah oleh algoritma

berupa sesuatu yang unik di sekitar gambar tersebut seperti, titik, sudut, dan persimpangan garis, tergantung dari algoritma apa yang digunakan. Seperti halnya mengolah titik pada gambar yang bisa dilakukan oleh beberapa algoritma seperti Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Feature (SURF), Binary Robust Invariant Scalable Keypoints (BRISK) dan lain-lain (Johansson 2015), algoritma pengolahan sudut, Smallest Univalued Segment Assimilating Nucleus (SUSAN), Features from Accelerated Segment Test (FAST), Features from Accelerated Segment Test - Enhanced Repeatability (FAST-ER), dan lain-lain (Anjum et al. 2016), dan algoritma persimpangan garis, Maximally Stable Extremal Regions (MSER), Intensity Extrema-Based Region (IBR), Edge-Based Region (EBR) dan lain-lain (Cayon, Djelouah, and Drettakis 2015). Semua fitur unik dari Descriptors di kalkulasi agar dapat di klasifikasi dan di identifikasi (Billinghurst, Clark, and Lee 2015).

Seperti halnya tulisan yang telah dibuat oleh Yang Yu, Yingchun Guo, Ruili Wang, Susha Yin, dan Ming Yu dimana mereka membandingkan algoritma oriented FAST and rotated BRIEF (ORB) dan Fast retina keypoint (FREAK) dalam kinerja tracking AR (Y. Yu et al. 2017). L. Yu, S. K. Ong, dan A. Y. C. Nee, menggunakan algoritma ORB dan menambahkan algoritma ESM tracking a near-field 3D tracking system (L. Yu, Ong, and Nee 2016). Chuin-Mu Wang, membandingkan algoritma ORB, Binary Robust Independent Elementary Features (BRIEF), binary robust invariant scalable keypoints (BRISK) dan FREAK untuk merujuk pembaca untuk menggunakan algoritma yang sesuai dengan situasinya (C. M. Wang 2015). Mariusz Oszust, mengoptimasi algoritma FAST dan memberinya nama Optimised Binary Robust FAST Features (OBRAF) yakni algoritma yang teroptimasi dapat mempresisikan kepresisian dari key point saat tracking dan membandingkan algoritma Speed Up Robust Feature (SURF), Scale Invariant Feature Transform (SIFT), ORB (Oszust 2016). Peng Chen, Zhang Peng, Dalong Li, dan Lijuan Yang, menganalisis algoritma ORB untuk pada objek marker based dan marker-less based (Chen et al. 2016).

Dari beberapa tulisan - tulisan di atas penulis mengambil algoritma ORIENTED FAST AND ROTATED BRIEF (ORB) dan ADAPTIVE AND GENERIC ACCELERATED SEGMENT TEST (AGAST), Tingkat Akurasi Pengenalan

Isyarat Tuna Rungu Menggunakan SVM Dengan fitur lokal ORIENTED FAST AND ROTATED BRIEF “ORB” dan ADAPTIVE AND GENERIC ACCELERATED SEGMENT TEST “AGAST”. Guna mendapatkan hasil perbandingan akurasi saat mentracking suatu objek agar pembaca tulisan ini dapat mengimplementasikan Descriptors lebih akurat ataupun mengembangkan teknologi berbasis komputer visual ke tingkat yang lebih tinggi lagi.

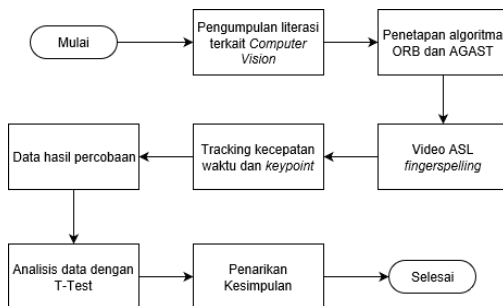
II. METODOLOGI PENELITIAN

Penelitian ini dimulai dari pengumpulan literasi - literasi terkait tentang Computer Vision, Penelitian dilanjutkan dengan menentukan algoritma yang akan dibandingkan dan penulis menetapkan untuk menggunakan algoritma ORB dan AGAST yang dimana kedua algoritma tersebut memiliki dasar algoritma yang sama yakni FAST, dilanjutkan dengan pengumpulan bahan mencari dan menentukan dasar video untuk pentrackingan kedua algoritma yakni bahasa isyarat tuna rungu agar dapat berkontribusi dalam membantu penderita tuna rungu dengan tulisan ini dikemudian hari.

Dilanjutkan dengan prototyping algoritma dengan menggunakan bahasa Python dan library OpenCV, setelah prototype tersedia penulis melakukan pengumpulan data percobaan dengan mentrackingan video yang di repetisi sebanyak 10 kali di setiap algoritma nya agar dapat menghasilkan data yang valid, dengan adanya data - data yang didapat berupa jumlah keypoint, lokasi keypoint pada setiap gambar dan kecepatan tracking dari setiap algoritma, data yang telah di peroleh oleh penulis kemudian divalidasi agar dapat di lanjutkan ke tahap pengolahan data, dari data jumlah keypoint dan kecepatan tracking penulis mencari standar deviasi dari masing – masing data agar dapat dilihat perbedaannya, dilanjutkan dengan mengkomparasi data tersebut dengan metode T-Test agar dapat ditarik kesimpulan, sedangkan untuk akurasi dari algoritma ORB dan AGAST, setelah penulis mendapatkan data lokasi dari keypoint, dilanjutkan dengan memvalidasi data yang telah di peroleh, setelah data di validasi penulis menggunakan alat bantu Support Vector Machine (SVM) dengan menjadikan 30% dari total data 840 sebagai data testing, yang di ulang sebanyak 10 kali dan di lanjutkan dengan mengkomparasi data tersebut menggunakan metode T-Test agar dapat di tarik kesimpulan.

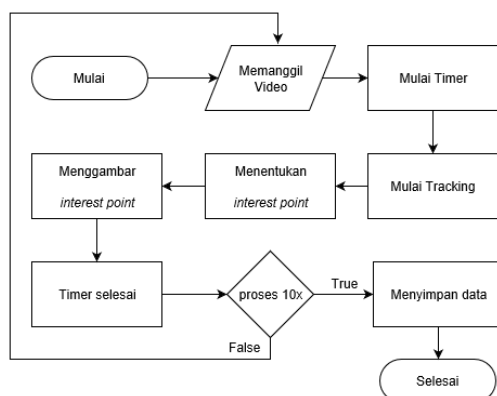
Dalam menentukan perbandingan Keypoint dan kecepatan, perbandingan dapat dilihat setelah melakukan standar deviasi dan setelah itu di masukan dedalam tabel dan di jadikan chart lalu di olah menggunakan metoda T-Test.

Kesimpulan dapat ditarik dari data yang telah dianalisis agar penelitian ini mendapatkan jawaban dari rumusan masalah yang ada pada tulisan ini.



Gambar 1. Diagram alir penelitian.

Diagram alir proses Tracking kecepatan waktu dan keypoint dapat dilihat pada Gambar 3.2 Pada pengolahan data terdapat proses repetisi sebanyak 10 kali di setiap algoritmanya untuk mendapatkan data hasil experiment valid yang dapat di olah.



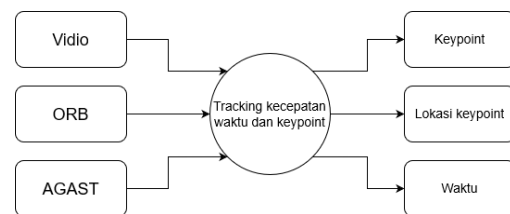
Gambar 2. Diagram alir Tracking kecepatan waktu dan keypoint.

A. Jenis, Sifat, dan Pendekatan Kuantitatif

Penelitian ini menggunakan metode experimental untuk mendapatkan data keypoint, lokasi keypoint dan kecepatan tracking dari algoritma ORB dan AGAST. Data hasil dari experimental tersebut sebelum di olah akan di cari rata-rata nya, menghitung standar deviasi, menghitung varian dan di validasi. lalu di jadikan tabel dan chart agar dapat di baca, dan kemudian dilanjutkan dengan mengkomparasi data tersebut menggunakan metode t-test guna mendapatkan hasil. Kegiatan penelitian membutuhkan hasil dari komparasi dua algoritma yakni ORB dan AGAST yang memiliki dasar algoritma yang sama yaitu

FAST guna mendapatkan hasil komparasi berupa tingkat Keypoint dan tingkat kecepatan. Penulis menggunakan bahasa isyarat tunarungu dalam bentuk video ASL fingerspelling sebagai dasar tracking dari kedua algoritma.

Dalam tulisan ini, ada 5 bagian utama yang akan dilakukan penulis yaitu tahap bahan video yang dimana penulis mencari video berupa ASL fingerspelling, kemudian pembuatan prototype yang menggunakan bahasa pemrograman Python dan library OpenCV, lalu melakukan proses pentrackingan video yang di repetisi sebanyak 10 kali guna mendapatkan hasil keypoint, lokasi keypoint dan kecepatan yang valid, setelah mendapatkan data hasil experiment penelitian ini dilanjutkan dengan memvalidasi data hasil, dilanjutkan mengkomparasi data yang telah di validasi menggunakan metode T-Test guna mendapatkan hasil komparasi dari kedua algoritma.



Gambar 3. Proses input output tracking algoritma

B. Metode Pengolahan Data

Dengan membuat prototype untuk menjalankan algoritma AGAST dan ORB, penelitian ini dilanjutkan dengan mentracking video American Sign Language (ASL) fingerspelling A-Z untuk mendapatkan data hasil experiment berupa keypoint, lokasi keypoint dan kecepatan dari setiap algoritma tersebut, guna mendapatkan data yang dapat divalidasi, penelitian ini melakukan repetisi video sebanyak 10 kali di setiap algoritma nya.

Setelah mendapatkan data hasil experiment, lalu data tersebut di rata – rata, di cari standar deviasi dan varian nya dan kemudian di validasi dan di tampilkan dalam chart dan tabel. Setelah memvalidasi data hasil experiment penelitian ini dilanjutkan dengan mengkomparasi keypoint dan kecepatan waktu dari algoritma ORB dan AGAST agar dapat dilihat perbedaannya dan dapat di tarik kesimpulan.

C. Metode Analisis Data

Dengan didapatkannya data hasil experiment yang didapat dengan mentracking video American Sign Language (ASL) fingerspelling A-Z dari setiap algoritma yang digunakan, memiliki chart dan tabel dari data experiment kecepatan tracking, keypoint dan lokasi keypoint yang valid. penelitian ini dilanjutkan

dengan mengkomparasi statistik dan tabel menggunakan T-Test di setiap alfabet ASL fingerspelling, agar penelitian ini dapat dilanjutkan ke tahap kesimpulan.

III. HASIL DAN PEMBAHASAN

A. Memanggil Video

Pemanggilan video dilakukan agar dapat melakukan proses pentrackingan, sebelum algoritma melakukan tracking terlebih dahulu video dipecah per frame nya agar algoritma ORB dan AGAST dapat melakukan proses tracking.

Video ASL fingerspelling berjudul "ASLV.avi" yang memiliki durasi 00:28:00 dengan total 840 frame (30 frame per second) di dalam nya, setelah memecah video "ASLV.avi" menjadi frame per gambar dan mengetahui seberapa banyak frame yang dibutuhkan untuk membentuk satu alfabet ASL fingerspelling.

TABEL I.

TOTAL FRAME ALFABET A-Z

Alfabet	Frame No	Total Frame	Alfabet	Frame No	Total Frame
A	1-32	32	N	407-436	29
B	33-67	34	O	437-467	30
C	68-88	20	P	468-519	51
D	89-114	25	Q	520-559	39
E	115-146	31	R	560-604	44
F	147-179	32	S	605-636	31
G	180-211	31	T	637-668	31
H	212-246	34	U	669-705	36
I	247-283	36	V	706-727	21
J	284-311	27	W	728-752	24
K	312-341	29	X	753-774	21
L	342-366	24	Y	775-793	18
M	367-406	39	Z	794-840	46
Total Frame A-Z				840	

B. Memulai Timer

Timer dimulai setelah pemecahan video ke gambar per frame dan sebelum pemanggilan algoritma, agar dapat menghitung waktu yang terjadi dalam proses pentrackingan suatu algoritma.

C. Mulai Tracking

Memanggil algoritma ORB dan AGAST untuk memulai pentrackingan video yang telah dipecah menjadi gambar per frame.

D. Menentukan dan Menggambar Keypoint

Menentukan Keypoint dari gambar per frame dalam point ini kedua algoritma melakukan kerja nya mencari, menentukan dan menggambar Keypoint. Algoritma AGAST dan ORB, kedua algoritma tersebut sebelum memulai mentracking suatu frame gambar terlebih dahulu membuat gambar tersebut menjadi grayscale dan setelah nya menentukan keypoint pada fitur di setiap algoritmanya, yang dimana algoritma AGAST menggunakan pohon keputusan AGAST Dengan rumus $S_n \rightarrow x$ dan bantuan pohon keputusan AGAST mencari pixel yang di sekitar nya memiliki 2 warna pixel yang berbeda, not brighter (b) atau not darker (d) (lihat Rumus 2.1). Setelah lokasi keypoint dari AGAST ditentukan dan setelah nya lokasi dari keypoint tersebut di extract yang hasil nya menjadi data lokasi keypoint.

Sedangkan algoritma ORB rumus Centroid $c = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}})$ dan menandai pertengahan sudut sebagai Keypoint nya, sebagai contoh ketika algoritma ORB telah menemukan sudut yang presisi dan telah menemukan sumbu x dan sumbu y dari sudut tersbut kemudian algoritma ORB menggunakan rumus centroid untuk menghasilkan pertengahan sudut dan menjadikan nya sebagai keypoint. Setelah lokasi keypoint ditentukan lalu lokasi dari keypoint tersebut di ambil dan dikumpulkan guna mendapatkan data lokasi keypoint.

E. Timer Selesai

Proses ini menyatakan selesainya pentrackingan satu frame dengan ditandai dengan muncul nya kecepatan tracking satu frame, Dengan menggunakan waktu yang ada pada windows menyimpan waktu saat memulai dan menyimpan waktu saat proses berakhir dan menghitung nya agar mendapatkan waktu kecepatan tracking, seperti contoh di bawah.

waktu selesai - waktu mulai = waktu kecepatan tracking

00:01:30.02 - 00:01:29.04 = 00:00:00.08

F. Langkah Penerapan SVM

Setelah mendapatkan data lokasi keypoint dari algoritma ORB dan AGAST, di lanjutkan dengan pembuatan dataframe dari data yang telah di dapat,

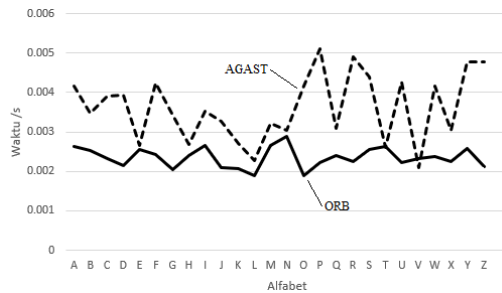
	feature				label
0	583.	402.	583.	407.	A
1	583.	402.	584.	405.	A
2	578.	400.	582.	411.	A
3	578.	400.	585.	400.	A
4	578.	400.	573.	402.	A
...	...	...	...	...	...
836	496.	432.	502.	432.	Z
837	494.	432.	500.	432.	Z
838	495.	432.	506.	434.	Z
839	494.	432.	500.	432.	Z
840	495.	432.	506.	434.	Z

Gambar 4. Contoh Dataframe

dan di lanjutkan dengan menerapkan SVM guna mendapatkan akurasi dari setiap algoritma.

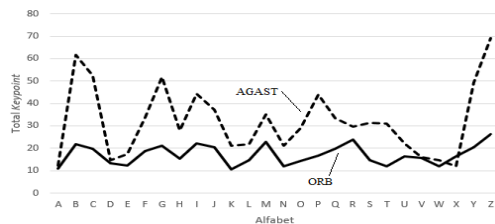
G. Grafik Data Standar Deviasi Algoritma

Setelah semua data kecepatan tracking dan keypoint diperoleh setelah itu data – data tersebut di cari standard deviasi nya agar dapat dimasukan ke dalam grafik dan ditelaah.



Gambar 5. Grafik perbandingan waktu algoritma ORB dan AGAST.

Dari gambar di atas dapat di lihat bahwa algoritma AGAST memiliki waktu tracking lebih tinggi dibandingkan dengan algoritma ORB namun algoritma ORB terlihat lebih stabil dalam waktu tracking yang dapat dilihat dengan tingkat fluktuasi yang terjadi pada gambar di atas.

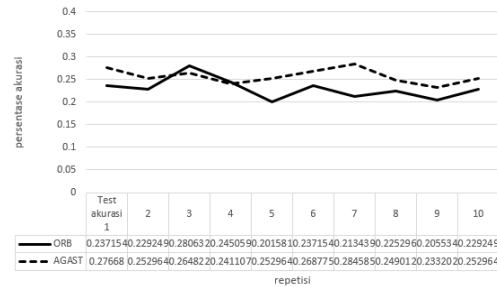


Gambar 6. Grafik perbandingan keypoint algoritma ORB dan AGAST

Dari gambar perbandingan keypoint algoritma ORB dan AGAST di atas walaupun memiliki perbedaan yang signifikan namun dapat di lihat bahwa kedua algoritma tersebut memiliki tingkat fluktuasi yang relative sama.

H. Data Akurasi Menggunakan SVM

Setelah mendapatkan data lokasi keypoint dilanjutkan dengan membuat dataframe menggunakan data yang tersedia agar data yang tersedia tersebut dapat di baca oleh SVM, dengan menggunakan 30% data frame yang tersedia menjadi data test dari jumlah total 840 dataframe, dan menggunakan multiclass SVM untuk menentukan akurasi nya. Dengan presentase akurasi 1 sebagai 100% dan repetisi dilakukan sebanyak 10 kali, dapat di lihat pada Gambar di bawah.



Gambar 6. Grafik perbandingan akurasi algoritma ORB dan AGAST

I. Perhitungan Akurasi Dengan T-Test

Pada Bagian analisis data penelitian akan dilanjutkan dengan mengkomparasi data experiment yang telah di validasi pada sub bab sebelumnya menggunakan metode T-Test guna mendapatkan hasil, metode T-Test yang digunakan adalah T-Test for two dependent means yang dimana kedua algoritma tersebut yakni ORB dan AGAST di perlakukan sepenuhnya sama, dimulai dengan mengkomparasi Keypoint kedua algoritma menggunakan T-Test, analisis data menggunakan perhitungan T-test dapat di lihat pada tabel dibawah.

TABEL II.

PERHITUNGAN AKURASI DENGAN T-TEST

	Difference (AGAST- ORB)	Deviation (Difference – M)	Square Deviation
Test 1	0.04	0.01	0
Test 2	0.02	0	0
Test 3	-0.02	-0.04	0
Test 4	0	-0.03	0
Test 5	0.05	0.02	0
Test 6	0.03	0	0
Test 7	0.07	0.04	0
Test 8	0.02	0	0
Test 9	0.03	0	0
Test 10	0.02	0	0
M: 0.03		S: 0.01	

Dari data pada tabel atas dilanjutkan dengan menghitung Difference Score agar dapat menghitung nilai dari T nya.

Difference Scores Calculations

Mean: 0.03

$\mu = 0$

$S^2 = SS/df = 0.01/(10-1) = 0$

$S^2M = S^2/N = 0/10 = 0$

$SM = \sqrt{S^2M} = \sqrt{0} = 0.01$

T-value Calculation

$t = (M - \mu)/SM = (0.03 - 0)/0.01 = 3.47$

Dari perhitungan T-Test di atas diketahui t score akurasi kedua algoritma adalah 3.47 maka dapat di lihat perbedaan dengan menambahkan signifikan level 5% yang dimana Ttable dari signifikan level 5% nya adalah 2,059539, dengan mengabaikan bilangan positif dan negatif maka t score akurasi = $3.47 > T_{table} =$

2,059539, dengan adanya hipotesis $H_0 =$ Terdapat perbedaan antara algoritma ORB dan AGAST, yang dimana $\mu_0 > \mu_1$ dan selain dari H_0 maka akan dianggap memiliki kesamaan. Maka keypoint dan kecepatan (μ_0) lebih besar dari Table (μ_1) menunjukkan bahwa perbedaan ini dianggap sangat signifikan $< 5\%$ secara statistik, dengan kata lain dapat disebut bahwa Keypoint, kecepatan, dan akurasi pada algoritma ORB dan AGAST sangat berbeda dimana algoritma ORB dan AGAST hanya memiliki kesempatan $< 5\%$ untuk menjumpai kesamaan secara kebetulan.

IV. KESIMPULAN

Dari experiment yang telah di lakukan penelitian ini menemukan kesimpulan dimana algoritma tracking ORB dan AGAST memiliki perbedaan dari segi keypoint dan kecepatan dalam mentracking video Bahasa isyarat tuna rungu berjudul “ASL fingerspelling”, dan dengan experiment di atas maka penulis dapat menarik kesimpulan bahwa Dari experiment yang telah di lakukan dan di bandingkan menunjukkan bahwa algoritma ORB dan AGAST memiliki perbedaan yang sangat signifikan secara statistik yang di mana algoritma ORB dan AGAST memiliki kesempatan 5% untuk menjumpai kesamaan secara statistik, Tingkat akurasi dari algoritma ORB dan AGAST memiliki perbedaan yang signifikan berdasarkan ilmu statistik, namun dapat di lihat pada Gambar 6. dari 10kali percobaan ada saat dimana algoritma ORB lebih unggul dalam akurasi (lihat test ke2) dan sedangkan dengan 9 percobaan nya algoritma AGAST memiliki akurasi yang lebih unggul, yang dapat dikatakan bahwa akurasi algoritma AGAST lebih baik dalam mentracking gerakan bahasa isyarat tuna rungu.

UCAPAN TERIMA KASIH

Terimakasih kepada semua yang telah berkontribusi membantu dalam menyelesaikan tulisan ini.

REFERENSI

- [1] Anjum, F, N Kanwal, Z Altaf, and A Shaukat. 2016. "Comparing Multiple Corner Detection Algorithms Using Non- Parametric Statistical Tests." 68(1): 97–102.
- [2] Billinghamurst, Mark, Adrian Clark, and Gun Lee. 2015. "A Survey of Augmented Reality." A Survey of Augmented Reality 6(4): 74–239.
- [3] Cayon, Rodrigo Ortiz, Abdelaziz Djelouah, and George Drettakis. 2015. "A Bayesian Approach for Selective Image-Based Rendering Using Superpixels." Proceedings - Robust Infrared Small Target Detection Using Local Steering Kernel Reconstruction." Pattern Recognition 77: 113–25. <https://doi.org/10.1016/j.patcog.2017.12.012>.
- [14] Lindeberg, Tony. 1993. "Detecting Salient Blob-Like Image Structures and Their Scales with a Scale-Space Primal Sketch: A Method for Focus-of-Attention." International Journal of Computer Vision 11(3): 1–36. papers2://publication/uuid/BEA14951-266B-44AF-9312-D61319A7FC0A.

- [15] ———. 1998. "Feature Detection with Automatic Scale Selection." *International Journal of Computer Vision* 30(2): 79–116.
- [16] Madero Orozco, Hiram et al. 2014. "Automated System for Lung Nodules Classification Based on Wavelet Feature Descriptor and Support Vector Machine." *BioMedical Engineering Online* 14(1): 1–20.
- [17] Mair, Elmar et al. 2010a. "Adaptive and Generic Corner Detection Based on the Accelerated Segment Test." In , 183–96. http://link.springer.com/10.1007/978-3-642-15552-9_14 (April 5, 2019).
- [18] ———. 2010b. "Computer Vision -- ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II." Adaptive and Generic Corner Detection Based on the Accelerated Segment Test: 183–96.
- [19] Marchand, Eric, Hideaki Uchiyama, and Fabien Spindler. 2016. "Pose Estimation for Augmented Reality: A Hands-On Survey." *IEEE Transactions on Visualization and Computer Graphics* 22(12): 2633–51.
- [20] Oszust, Mariusz. 2016. "An Optimisation Approach to the Design of a Fast, Compact and Distinctive Binary Descriptor." *Signal, Image and Video Processing* 10(8): 1401–8.
- [21] Parkhi, Omkar M., Andrea Vedaldi, and Andrew Zisserman. 2015. "Deep Face Recognition." *Proceedings of the British Machine Vision Conference 2015 (Section 3)*: 41.1-41.12. <http://www.bmva.org/bmvc/2015/papers/paper041/index.html>.
- [22] Quesada, Luis, Gustavo López, and Luis Guerrero. 2017. "Automatic Recognition of the American Sign Language Fingerspelling Alphabet to Assist People Living with Speech or Hearing Impairments." *Journal of Ambient Intelligence and Humanized Computing* 8(4): 625–35.
- [23] Rosin, Paul L. 1999. "Measuring Corner Properties." *Computer Vision and Image Understanding* 73(2): 291–307.
- [24] Rublee, Ethan, V Rabaud, and K Konolige. 2011. "ORB: An Efficient Alternative to SIFT or SURF About Local Feature and Matching Motivation OFAST – Oriented FAST BRIEF (Calonder et Al . 2010)." *Intl. Conf. Computer Vision*: 1–5.
- [25] Simo-Serra, Edgar et al. 2015. "Discriminative Learning of Deep Convolutional Feature Point Descriptors." *Proceedings of the IEEE International Conference on Computer Vision 2015 Inter*: 118–26.
- [26] Valli, C., Ceil Lucas. 2000. *Linguistics of American Sign Language: An Introduction*, Library of Congress Cataloging-in-Publication Data, An Imprint of Gallaudet University Press, Washington, DC, 3rd edition,
- [27]
- [28] Wang, Chuin Mu. 2015. "Real Time Non-Rigid Surface Detection Based on Binary Robust Independent Elementary Features." *Journal of Applied Research and Technology* 13(2): 297–304. <http://dx.doi.org/10.1016/j.jart.2015.06.005>.
- [29] Wang, Han, and Michael Brady. 1995. "Real-Time Corner Detection Algorithm for Motion Estimation." *Image and Vision Computing* 13(9): 695–703.
- [30] Weimer, Daniel, Bernd Scholz-Reiter, and Moshe Shpitalni. 2016. "Design of Deep Convolutional Neural Network Architectures for Automated Feature Extraction in Industrial Inspection." *CIRP Annals - Manufacturing Technology* 65(1): 417–20. <http://dx.doi.org/10.1016/j.cirp.2016.04.072>.
- [31] Yilmaz, Alper, Omar Javed, and Mubarak Shah. 2006. "Object Tracking." *Object Tracking: A Survey* 38(1-45): 119–46.
- [32] Yu, L., S. K. Ong, and A. Y.C. Nee. 2016. "A Tracking Solution for Mobile Augmented Reality Based on Sensor-Aided Marker-Less Tracking and Panoramic Mapping." *Multimedia Tools and Applications* 75(6): 3199–3220.
- [33] Yu, Yang et al. 2017. "Registration Based on ORB and FREAK Features for Augmented Reality Systems." *Transactions of Tianjin University* 23(2): 192–200.