

## LAYAR PROYEKSI HOLOGRAM INTERAKTIF DENGAN KINERJA HAND GESTURE DETECTION DAN TRACKING MODEL REINFORCEMENT LEARNING

### INTERACTIVE HOLOGRAM PROJECTION SCREEN WITH HAND GESTURE DETECTION AND TRACKING PERFORMANCE USING REINFORCEMENT LEARNING METHOD

Muthia Saada<sup>1</sup>, Suryo Adhi Wibowo, Ph.D.<sup>2</sup>, Iwan Iwut Tritoasmoro, S.T., M.T.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik, Universitas Telkom

<sup>1</sup>mutiasaa@student.telkomuniversity.ac.id, <sup>2</sup>suryoadhiwibowo@telkomuniversity.ac.id,

<sup>3</sup>iwaniwut@telkomuniversity.ac.id

---

#### Abstrak

Teknologi *hologram* adalah bentuk topografi yang merekam gambar dalam visual 3 dimensi dan diproyeksikan pada permukaan 2 dimensi. Gambar *hologram* dapat digunakan untuk hiburan, pembelajaran dan lain – lain. *Hologram* sendiri juga bisa menjadi teknologi untuk penyampaian informasi. *Human computer interaction* (HCI) adalah ilmu yang mempelajari bagaimana manusia dan komputer dapat berinteraksi. Pada tugas akhir ini, dirancang layar proyeksi yang menampilkan gambar seolah-olah 3 dimensi yang bisa berinteraksi dengan gerakan tangan (*hand gesture*). Sistem ini bergerak secara *real time* untuk pengenalan dan deteksi gerakan tangan yang ditentukan pengguna yang diambil menggunakan kamera *webcam*. Di sistem ini digunakan SURF untuk mendapatkan fitur *bag of visual words* dari tangan pengguna. fitur itu nanti akan digunakan untuk memprediksi tangan pengguna. Sistem dapat meningkatkan keakuratannya selama waktu pengujian dengan menggunakan metode *reinforcement learning* yaitu disini pengguna memiliki kemampuan untuk memberikan umpan balik positif atau negatif untuk setiap prediksi sistem sebelumnya dan sistem akan memperbaiki sendiri selama waktu pengujian untuk hasil prediksi yang lebih baik. Pada Tugas Akhir ini pengujian dilakukan dalam 3 tahap, 30 *frames*, 40 *frames* dan 50 *frames* serta dilakukan penerapan konfigurasi jarak tangan terhadap kamera. Berdasarkan percobaan yang dilakukan nilai akurasi yang didapat terus meningkat dalam tiap tahapnya. Didapatkan hasil terbaik untuk akurasi prediksi *gesture* tangan saat menggunakan data 50 *frames* sebesar 98.70%. Proses HCI pada *hologram* saat memasukkan gestur unik dari tangan pengguna bisa untuk membuka google, instagram, dan video di youtube

**Kata Kunci:** *Hologram, Hand Gesture Detection, Webcam, 3D, Reinforcement Learning, HCI*

---

#### Abstract

*Hologram is a 3-dimensional projection of an object projected on 2dimensional surface. Holographic image can be applied in various fields such as entertainment, education, etc. Holograms can also be a technology for information delivery. Human computer interaction (HCI) is the study of how humans and computers can interact. In this Final Project, designed a projection screen that displays images as if they were 3 dimensions that can interact with hand gestures to convey information. This system moves in real time for the introduction and detection of user-determined hand movements taken using a webcam camera. In this system SURF is used to get the bag of visual words feature from the user's hand. That feature will later be used to predict the user's hand. The system can improve its accuracy during the test time by using the reinforcement learning method, where the user has the ability to provide positive or negative feedback for each of the previous system's predictions and the system will update itself during the test time for better predictive results. In this final project, the test is carried out in 3 stages, 30 frames, 40 frames and 50 frames and the hand distance configuration is applied to the camera. Based on the experiments conducted, the accuracy value obtained continues to increase in each stage. The best results were obtained for the accuracy of hand gesture prediction when using 50 frames of data at 98.70%. The HCI process on the hologram when entering a unique gesture from the user's hand can open Google, Instagram, and videos on YouTube.*

**Keywords:** *Hologram, Hand Gesture Detection, Webcam, 3D, Reinforcement learning, HCI*

---

## 1. Pendahuluan

Teknologi pada zaman sekarang sudah berkembang pesat. Sudah mulai bermunculan gagasan tentang interaksi manusia – komputer (HCI) menggunakan gerakan tangan. Seperti di film ‘iron man’ yang menampilkan sistem pengenalan gerakan tangan yang begitu membuat terpujau. Pengenalan gerakan tangan ini merupakan sebuah inovasi yang berguna untuk navigasi robot, HCI, *automated homes*, *augmented reality*, dan berbagai macam lainnya. Penggunaan *hologram* untuk teknologi HCI dirasa bisa membuat teknologi tersebut semakin menarik.

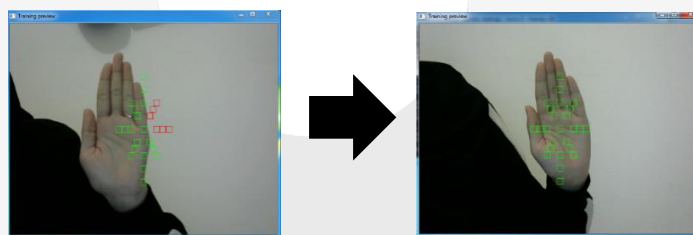
Penggunaan teknologi *hologram* sendiri bisa digunakan untuk menyampaikan informasi [1]. Karena sekarang ini masih sulit untuk menemukan media penyampaian informasi yang menarik. Informasi berbentuk *hologram* dengan visual seolah-olah tiga dimensi di rasa bisa membantu merekayasa informasi agar efisien dan interaktif dalam mengolah informasi yang nantinya akan direalisasi dan dikembangkan.

Untuk bisa berinteraksi antara pengguna dan informasi diperlukan kinerja dari *hand gesture detection*. Ada beberapa penelitian terkait metode apa saja yang bisa digunakan. Pada penelitian *hand gesture detection* dengan menggunakan metode *Hidden Markov Model* (HMM) yang memakai dataset yang cukup besar didapatkan akurasi yang tinggi, tetapi algoritma ini hanya bisa digunakan untuk pengenalan gerak tangan yang membutuhkan gerak minimum saja [2]. Pada penelitian yang lain menggunakan algoritma *adaboost* untuk melakukan deteksi tangan [3]. Deteksi yang didapatkan cukup baik karena dapat mengatasi gangguan yang disebabkan oleh masalah warna kulit, tetapi membutuhkan waktu pelatihan yang sangat lama, membutuhkan dataset yang besar dan relatif lebih sulit untuk memperbarui pengklasifikasian selama waktu pengujian.

Pada Tugas Akhir ini dilakukan implementasi *hand gesture detection* pada *hologram* menggunakan metode *Reinforcement Learning* dengan menggunakan algoritma *Q-learning* untuk memprediksi gerakan tangan yang benar dari *input* tangan pengguna menggunakan *webcam*. Algoritma *Q-learning* merupakan bagian dari *Reinforcement Learning* yang dapat menambah akurasi untuk deteksi gerakan tangan yang dilakukan komputer, algoritmanya mudah, tidak membutuhkan data latih yang besar dan semua proses dalam sistem dilakukan secara *real time*.

## 2. Desain dan Perancangan

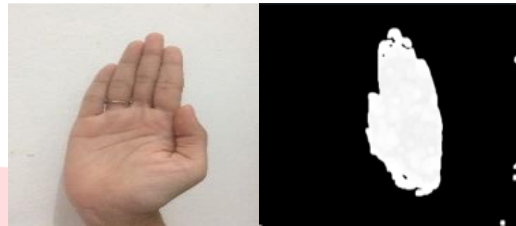
Pada penelitian ini prediksi tangan dilakukan secara *real time*. hal yang pertama dilakukan adalah melakukan segmentasi pada tangan pengguna. Ditampilkan serangkaian 25 jendela tetap di layar *webcam*, lalu masukkan tangan pengguna ke dalam layar *webcam* dan pengguna diminta untuk memasukkan tangannya ke dalam *frame* sehingga benar-benar menutupi semua jendela. Dapat ditunjukkan kepada pengguna jendela sudah menutupi semua dengan berubah warna dari merah menjadi hijau seperti yang terlihat di gambar 2.1. Lalu dilakukan ekstrak profil warna dari wilayah tertentu dari tangan pengguna dalam ruang warna HSV untuk memisahkannya dari latar belakang.



**Gambar 1:** kumpulan *windows* yang ditampilkan untuk mengambil warna tangan

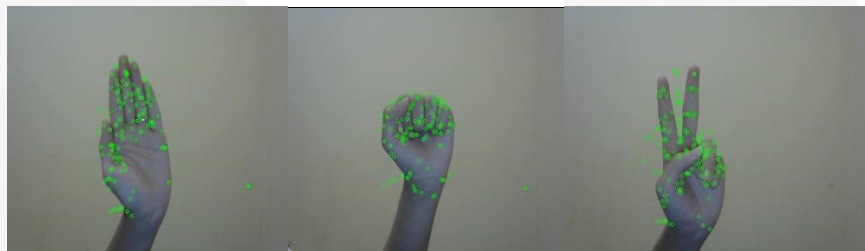
Warna kulit dibedakan dengan yang bukan warna kulit dengan melakukan pengujian terhadap batas atas dan batas bawah rentang *hue*, *saturation*, dan *value* sehingga didapatkan rentang HSV yang sesuai untuk membentuk citra hasil *threshold* untuk tangan[5]. *Hue* (H) adalah ukuran dari jenis warna seperti warna merah, kuning, hijau, dan seterusnya. Representasinya dalam bentuk derajat dengan nilai 0 – 360. *Saturasi* (S) adalah keberwarnaan suatu warna. Semakin berwarna sebuah warna berarti semakin besar nilai saturasinya. Namun apabila suatu warna pucat, itu berarti saturasinya rendah. *Value* (V) adalah nilai kecerahan sebuah warna. Warna cerah memiliki nilai *value* tinggi dan sebaiknya untuk warna yang gelap. Nilai HSV dibentuk dari nilai RGB (*red*, *green*, *blue*) yang telah dikonversi. Jadi untuk *output* dari proses segmentasi ini adalah objek yang sudah

terpisah dengan *background* dalam bentuk gambar biner yang bisa dilihat pada gambar 2.2. Untuk sistem ini digunakan 3 class yaitu tangan berbentuk telapak, *piece* dan kepal.



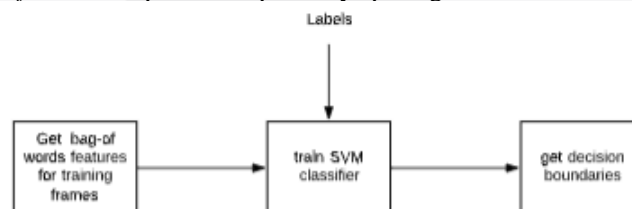
**Gambar 2:** Segmentasi *foreground* tangan.

Setelah segmentasi maka akan dilakukan proses ekstraksi fitur. Dilakukan pemanggilan algoritma SURF untuk mendeteksi *interest point*[6]. Deteksi *interest point* pada tangan pengguna dilakukan untuk mendapatkan deskripsi fitur yang unik. Jadi untuk membedakan gestur tangan yang satu dengan yang lainnya. Bisa dilihat pada gambar 2.3. Sesuai dengan algoritma SURF itu sendiri yang menggunakan perhitungan dengan pendekatan matriks hessian, maka perlu ditentukan *threshold* dari nilai determinan hessian agar suatu titik dapat dikatakan sebagai *interest point*. Semakin besar *threshold* yang digunakan maka semakin sedikit *interest point* yang terdeteksi, sebaliknya semakin kecil nilai *threshold* maka semakin banyak *interest point* yang terdeteksi. Di sistem ini *threshold* yang digunakan bernilai 5. Deteksi *interest point* yang ada pada tangan pengguna lalu di ekstraksi menggunakan SURF. Jadi dimana saat semua fitur sudah ditemukan dalam semua data *training* dilakukan pengklasteran dan membentuk *visual words*. Pada penelitian ini menggunakan *K-Means clustering*. Lalu setelah itu fitur yang sudah diekstrak akan disimpan didalam *bag of words*[7].



**Gambar 3:** *Interest point* pada isyarat telapak (a) *piece* (b) dan kepal (c)

Setelah fitur *bag of words* dari setiap gestur telah diekstraksi, lalu umpankan ke SVM *classifier*[8]. Lalu selama fase pengujian nanti, *classifier* SVM terlatih memprediksi gestur id yang sesuai dengan fitur *bag of words*. Dapat dilihat prosesnya pada gambar 2.5.



**Gambar 4:** *Training Phase*

Setelah itu masuk ke fase pengujian. Di fase pengujian ini menggunakan metode *Reinforcement Learning*. Jadi disini 10 prediksi sebelumnya ditampilkan kepada pengguna di sisi layar *webcam*-nya. Pengguna bisa meninjau dan berikan umpan balik positif / negatif pada frame samping kiri ini menggunakan klik *mouse* (klik kiri untuk umpan balik negatif dan klik kanan untuk umpan balik positif). Sistem menggunakan persamaan dibawah ini untuk memperbarui bobot  $\theta(a)$  berdasarkan umpan balik pengguna[9].

$$\theta_i(a) \leftarrow \theta_i(a) + \alpha(R-Q(s,a))x_i(s) \quad (1)$$

Dalam persamaan di atas,  $\alpha$  adalah *learning rate* yang berkisar antara 0 dan 1. Di sistem ini digunakan *learning rate* 0,01 dikarenakan saat melebihi nilai tersebut hasil akurasi akan menjadi *fluktuatif*.  $x_i(s)$  adalah vektor fitur BOW untuk status *input 's'* R adalah hadiah untuk umpan balik positif dan negatif. Kami menggunakan nilai positif konstan R + untuk umpan balik positif dan nilai negatif konstan R- untuk umpan balik negatif. Bobot yang diperbarui ini kemudian digunakan untuk menemukan nilai-Q untuk *frame* yang masuk berikutnya. Contoh perhitungannya adalah:

- Dilakukan update bobot dengan menggunakan metode *Q-learning* dalam *testing phase*.
- Diketahui nilai  $Q = 10$ ,  $\alpha = 0,5$ ,  $\theta_i(a) = 2$
- Setelah dilakukan tes, sistem melakukan prediksi yang salah terhadap citra tertentu.
- Sistem memberikan *punishment* (R-) sebagai timbal balik atas kesalahan prediksi yang dilakukan, nilai  $R = -3$
- Maka dari itu, berdasarkan rumus untuk meng-update bobot, bobot baru yang akan didapatkan adalah

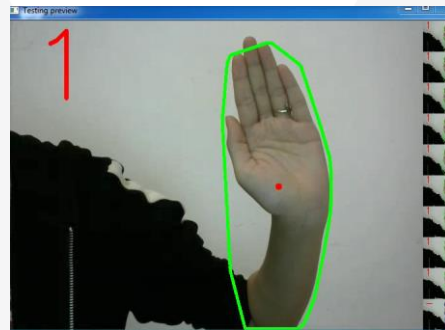
$$\theta_i(a) \leftarrow \theta_i(a) + \alpha(R-Q(s,a))x_i(s)$$

$$\theta_i(a) = 2 + 0,5 (3-10) 0,1$$

$$\theta_i(a) = 2 + (-0,35)$$

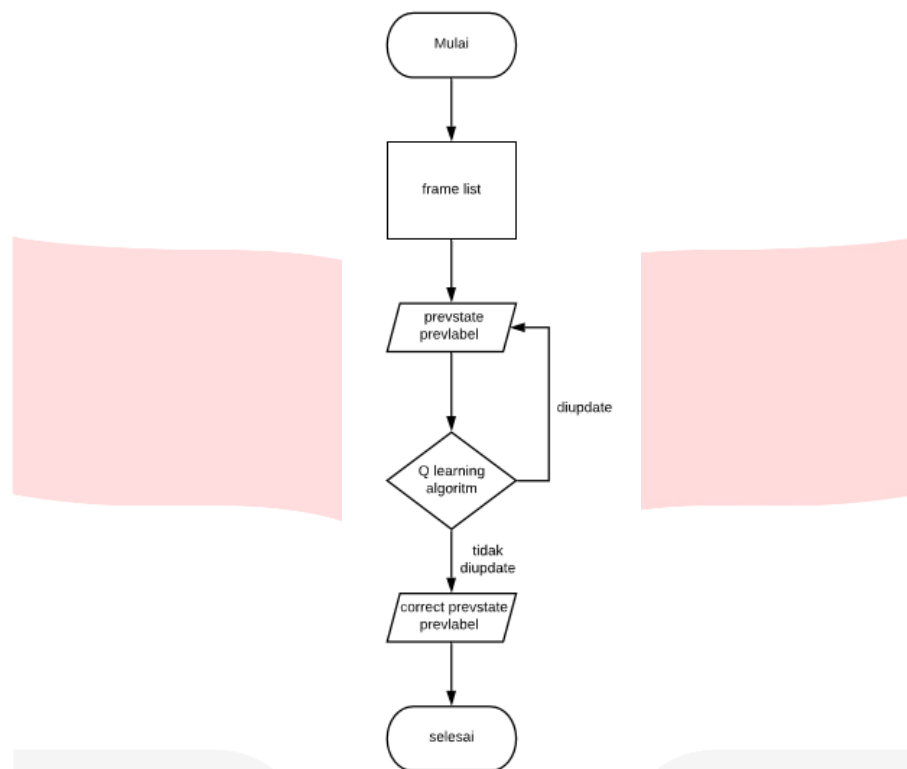
$$\theta_i(a) = 1,65$$

Proses bisa dilihat di gambar 2.6 dan gambar 2.7 untuk diagram alurnya.



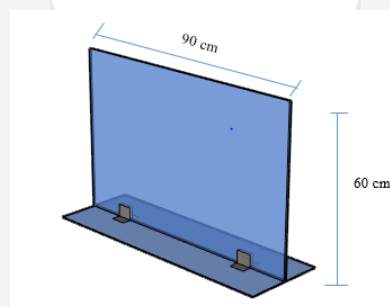
**Gambar 5:** Fase Pengujian

Prediksi label isyarat sistem ada di sudut kiri atas. 10 prediksi sebelumnya ditampilkan sebagai *frame* samping di sebelah kanan untuk umpan balik pengguna.



**Gambar 6:** Diagram alur proses pengujian

Desain perangkat yang digunakan sesuai pada gambar 2.7. Perangkat keras yang digunakan adalah kaca mika bening berbentuk persegi berukuran 90x0.5x60(cm) yang digunakan untuk memantulkan cahaya dari proyektor dengan video objek yang memiliki background hitam atau putih polos sehingga terlihat 3D. Pantulan di cermin bidang (kaca mika) menghasilkan gambar virtual untuk pengamat. Gambar ini nampak oleh penonton untuk diposisikan pada jarak yang sama di belakang permukaan cermin saat objek berada di depan[9]. Untuk menampilkan video, proyektor terhubung dengan personal computer (PC) sebagai pusat pemrosesan sistem.



**Gambar 7:** Desain Perangkat Keras.

### 3. Hasil dan Analisis

#### 3.1. Hasil Kinerja Sistem

Di bagian ini akan ditunjukkan kinerja dari sistem yang telah dibuat. Dilakukan berdasarkan beberapa parameter yang diubah selama proses kinerja sistem. Lalu nanti akan direpresentasikan dalam bentuk tabel dan grafik. Hal ini dilakukan untuk memprediksi isyarat tangan yang benar.

##### 3.2.1. Akurasi

Akurasi merupakan suatu nilai yang mengukur ketepatan sistem dalam mengenali citra atau objek masukan [13]. Akurasi secara matematis dapat dituliskan pada persamaan (4).

$$Akurasi = \frac{\text{Prediksi benar}}{\text{Total jumlah data uji}} \times 100\% \quad (4)$$

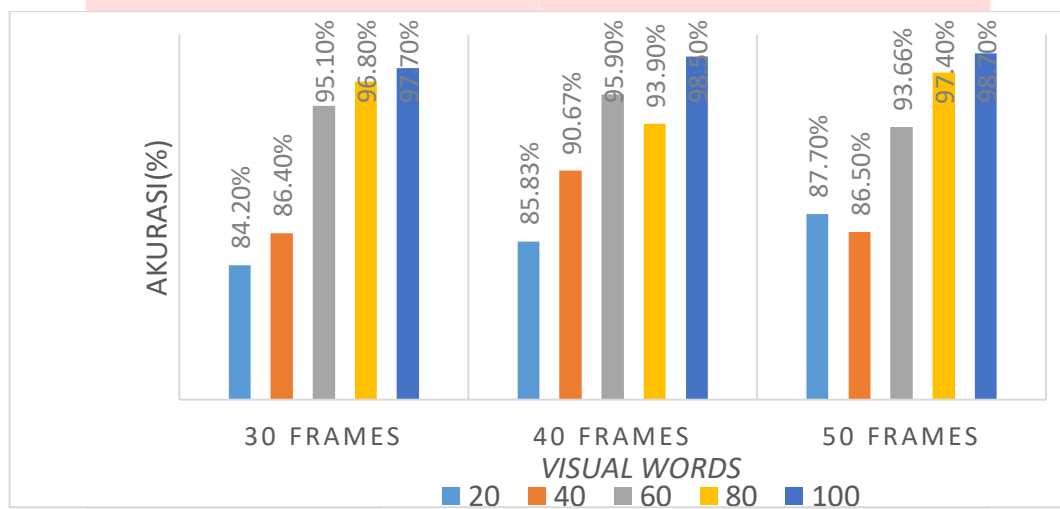
Akurasi merupakan parameter keberhasilan dari suatu sistem dalam memprediksi isyarat tangan yang benar. Jadi akan dilakukan perubahan ukuran *visual words* untuk 3 gesture dengan data latihnya 30 *frames*, 40 *frames*, dan 50 *frames*.

### 3.2.2. Jarak

Selanjutnya pengujian terhadap jarak, Dari jarak ini menentukan apakah objek (tangan pengguna) bisa terdeteksi atau tidak dengan jarak yang berbeda – beda. Semakin dekat jarak objek maka objek tersebut akan terdeteksi namun hal ini berbeda jika jarak makin jauh maka objek tidak akan terdeteksi. Pengujian dilakukan dengan jarak 0 cm sampai 51 cm.

### 3.3. Data Hasil Pengujian Sistem

Pengujian dilakukan dengan 3 gesture tangan dengan melakukan perubahan ukuran *visual words* dengan data latihnya 30 *frames*, 40 *frames*, dan 50 *frames* untuk mengukur akurasinya. Lalu dilakukan pengujian jarak deteksi objek dengan jarak 0 cm sampai lebih dari 51 cm.



**Gambar 8:** Akurasi untuk perubahan ukuran *visual words* pada 30 *frames*, 40 *frames*, dan 50 *frames*.

Pada gambar 3.1 merupakan grafik performansi dari parameter akurasi, dari grafik tersebut dapat dilihat bahwa saat menggunakan 30 *frames* dengan 20 *visual words* akurasi 84.2%, untuk 40 *visual words* akurasi sebesar 86.4%, untuk 60 *visual words* didapatkan akurasi sebesar 95.1%, 80 *visual words* didapatkan akurasi sebesar 96.8%, dan terakhir untuk 100 *visual words* didapatkan akurasi 97.7%. Disini dapat terlihat bahwa dengan meningkatkan ukuran dari *visual words* akan didapatkan akurasi yang akan terus naik hasilnya dikarenakan fitur-fitur unik pada *visual words* yang dipelajari sistem lebih banyak sehingga sistem lebih bisa mengenali objek tangan yang dideteksi oleh kamera webcam.

Untuk perubahan *frames* pada gambar 3.1 berjumlah 30 *frames*, 40 *frames* dan 50 *frames*. Pada *frames* 30 dengan *visual words* 20 nilai akurasinya adalah 84.2% sedangkan pada *frames* 40 dengan *visual words* 20 nilai akurasinya adalah 85.83%. Pada *frames* 30 dengan *visual words* 40 nilai akurasinya adalah 86.4%. Sedangkan pada *frames* 40 dengan *visual words* 40 didapatkan akurasinya adalah 90.67%. Lalu pada *frames* 30 dengan *visual words* 60 adalah 95.1% dan pada *frames* 40 dengan *visual words* 60 didapatkan hasil 95.9%. Pada *frames* 30 dengan *visual words* 80 hasil akurasi adalah 96.8% sedangkan untuk *frames* 40 dengan *visual words* 80 adalah 93.9%. Terakhir untuk *frames* 30 dengan *visual words* 100 didapatkan hasil akurasi 97.7% dan *frames* 40 dengan *visual words* 100 akurasinya adalah 98.5%. Disini dapat terlihat bahwa dengan menambah *frames* lebih banyak akan berpengaruh dengan kenaikan akurasinya. Untuk konfigurasi untuk *frames* 30 dan 40 menghasilkan perbedaan akurasi karena fitur-fitur yang dipelajari oleh sistem lebih banyak sehingga sistem lebih bisa mengenali gambar yang dideteksi kamera.

Begitu juga pada *frames* 40 dengan *visual words* 20 nilai akurasinya adalah 85.83% sedangkan pada *frames* 50 dengan *visual words* 20 nilai akurasinya adalah 87.7%. Pada *frames* 40 dengan *visual words* 40 nilai akurasinya adalah 90.67%. Sedangkan pada *frames* 50 dengan *visual words* 40 didapatkan akurasinya adalah 86.5%. Lalu pada *frames* 40 dengan *visual words* 60 adalah

95.9% dan pada *frames* 50 dengan visual words 60 didapatkan hasil 93.66%. Pada *frames* 40 dengan visual words 80 hasil akurasi adalah 93.9% sedangkan untuk *frames* 50 dengan visual words 80 adalah 97.4%. Terakhir untuk *frames* 40 dengan visual words 100 didapatkan hasil akurasi 98.5% dan *frames* 50 dengan visual words 100 akurasinya adalah 98.7%. Dapat dibuktikan disini dengan menambah *frames* lebih banyak untuk sistem akurasinya akan bertambah cukup signifikan. Walaupun ada penurunan akurasi yang terjadi saat sistem menggunakan 50 *frames* dengan visual words 80 tetapi mengalami kenaikan setelahnya.

**Tabel 1** Pengujian terhadap jarak.

No.	Jarak	Hasil pengujian
1	Sangat dekat ( $\leq 20$ cm)	Gagal
2	Dekat (21-30 cm)	Baik
3	Cukup (30 – 40 cm)	Baik
4	Jauh (40 – 50 cm)	Baik
5	Sangat jauh ( $\geq 51$ cm)	Gagal

Selanjutnya pengujian terhadap jarak. Tangan pengguna tidak terdeteksi di rentang jarak kurang dari 20 cm dikarenakan terlalu dekat dengan kamera *webcam* jadi sistem tidak bisa mendeteksi objek tangan dan juga *frame* hanya terisi oleh bagian tertentu saja dari tangan pengguna. Untuk jarak antara tangan dengan kamera *webcam* berkisar antara 21-30 cm bisa dideteksi dikarenakan jaraknya yang tidak terlalu dekat ataupun tidak terlalu jauh dari kamera jadi fitur-fitur pada tangan bisa terdeteksi. Begitu juga dengan jarak tangan antara kamera *webcam* berkisar antara 30-40 dan 40-50 cm bisa terdeteksi oleh sistem.

Sebaliknya untuk jarak tangan pengguna dengan kamera *webcam* di rentang jarak diatas 50 cm tidak bisa terdeteksi oleh sistem dikarenakan terlalu jauh dari kamera *webcam* dan keterbatasan kamera *webcam* yang hanya bisa mendeteksi dibawah 50 cm saja dan juga dikarenakan komponen fitur tangan terlalu sedikit untuk dapat dideteksi dengan jelas. Lalu untuk keluaran hasil dari prediksi gestur tangan, untuk telapak menggambarkan angka 1, untuk gestur *piece* menggambarkan angka 2 dan kepal untuk menggambarkan angka 3. Dan juga untuk *hologram* agar terlihat menarik bisa dilihat dari proses pengujian saat memasukkan gestur tangan tersebut yang bisa untuk membuka google, instagram, ataupun membuka video di youtube.

#### 4. Kesimpulan

Dari hasil kinerja sistem yang sudah didapatkan dalam memprediksi gerakan tangan, dapat disimpulkan bahwa skenario pertama yaitu untuk akurasi kinerja sistem memiliki hasil paling rendah pada saat ukuran *visual words* 20 dengan 30 *frames* adalah 84.2% dan hasil paling maksimal yaitu sekitar 97.7% untuk ukuran *visual words* 100. Untuk Akurasi sistem saat 40 *frames* dengan hasil paling rendah di *visual words* 20 yaitu 85.83% dan paling tinggi pada *visual words* 100 yaitu 98.5%. Dan terakhir untuk akurasi kinerja sistem saat 50 *frames* hasil paling rendah di ukuran *visual words* 40 yaitu 86.5% dan hasil maksimal di *visual words* 100 yaitu 98.7%. Jadi dengan meningkatkan ukuran *frames* dan *visual words* akan didapatkan akurasi yang akan terus naik hasilnya dikarenakan fitur-fitur yang dipelajari sistem lebih banyak sehingga sistem lebih bisa mengenali objek tangan yang dideteksi oleh kamera *webcam*.

Pengujian terhadap jarak, objek tangan hanya bisa terdeteksi di rentang jarak 20-50 cm. Untuk jarak kurang dari 20cm tidak bisa terdeteksi dikarenakan sistem tidak bisa mendeteksi objek tangan dan juga *frames* hanya terisi oleh bagian tertentu saja dari tangan pengguna. dan untuk jarak lebih dari 50 cm tidak bisa terdeteksi dikarenakan terlalu jauh dari kamera *webcam* dan keterbatasan kamera *webcam* yang hanya bisa mendeteksi dibawah 50 cm saja dan juga dikarenakan komponen fitur tangan terlalu sedikit untuk dapat dideteksi dengan jelas. Proses HCI pada *hologram* saat memasukkan gestur unik dari tangan pengguna bisa untuk membuka google, instagram, dan video di youtube.

**Daftar Pustaka:**

- [1] A. Elmorshidy, "Holographic Projection Technology: The World is Changing," vol. 2, no. 2, 2010.
- [2] J. Guo, "REAL-TIME HAND DETECTION BASED ON MULTI-STAGE HOG-SVM CLASSIFIER Guangdong Provincial Key Laboratory of Robotics and Intelligent System , Shenzhen Institutes of Advanced Technology , Chinese Academy of Sciences The Chinese University of Hong Kong," pp. 4108–4111, 2013.
- [3] Q. Chen, N. D. Georganas, and E. M. Petriu, "Hand gesture recognition using Haar-like features and a stochastic context-free grammar," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 8, pp. 1562–1571, 2008.
- [4] R. S. Sutton, A. G. Barto, and A. B. Book, "Next: Contents Contents Reinforcement Learning: An Introduction."
- [5] J. Wibowo, "Deteksi dan Klasifikasi Citra Berdasarkan Warna Kulit Menggunakan HSV," *None*, vol. 16, no. 2, p. 243286, 2011.
- [6] H. Bay, T. Tuytelaars, and L. Van Gool, "LNCS 3951 - SURF: Speeded Up Robust Features," *Comput. Vision–ECCV 2006*, pp. 404–417, 2006.
- [7] M. M. Farhangi, M. Soryani, and M. Fathy, "Informative *visual words* construction to improve bag of words image representation," *IET Image Process.*, vol. 8, no. 5, pp. 310–318, 2014.
- [8] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Support Vector Machine," 2003
- [9] M. J. R. a. J. D. Wiltshire, "Important Optical Principles and their Occurrence in Nature," IEEE, 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8068906>. [Accessed 25 11 2019].