PERANCANGAN DAN IMPLEMENTASI APLIKASI PENERJEMAH BAHASA ISYARAT MENJADI SUARA BERBASIS *KINECT* MENGGUNAKAN METODE *DYNAMIC TIME WARPING*

DESIGN AND IMPLEMENTATION OF SIGN LANGUAGE TO SPEECH APPLICATION BASED ON KINECT USING DYNAMIC TIME WARPING METHOD

Imam Hidayat Arifin Hasibuan[1], Asep Mulyana, S.T., M.T.[2], Andrew Brian O., S.T., M.T.[3]

[123]Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom University
[1]misterhasibuan@gmail.com [2]asepm267@gmail.com [3]abosmond@telkomuniversity.ac.id

Abstrak

Suatu isyarat dapat dikenali berdasarkan kombinasi bentuk dan gerak tangan. Dengan mengenali ciri-ciri khusus pada masing-masing isyarat, maka dapat diterjemahkan. Pada tugas akhir ini, dibangun suatu aplikasi pengenalan bahasa isyarat, agar orang-orang normal lebih mudah mengerti maksud isyarat seorang tunarungu.

Masukan isyarat (berdasarkan Kamus Umum Bahasa Isyarat Indonesia) yang diperagakan *user* akan ditangkap oleh sensor kinect secara *realtime,* sehingga menghasilkan skeleton joint setiap framenya. Kemudian frame-frame tersebut akan diproses untuk dikenali menggunakan metode Dynamic Time Warping (DTW) . Keluaran dari sistem ini berupa suara hasil keputusan sistem yang dianggap sebagai terjemahan dari isyarat yang dimasukkan sebelumnya.

Metode DTW dapat diterapkan pada aplikasi tersebut dengan memperoleh akurasi pengenalan hingga 92,78% dan rata-rata waktu komputasi setiap masukan yang dikenali berkisar 6,19760479 ms.

Kata Kunci: Bahasa Isyarat, *Kinect, Dynamic Time Warping*

Abstract

A sign language can be recognized with combination of shape and gesture hand. By recognizing specific characteristics of each sign, so it can be translated easily . This final project builts a sign language recognition application, so the common people will be easier to understand a sign language from the person who has deaf problem.

Sign language input (based Kamus Umum Bahasa Isyarat Indonesia) that practiced by user will be captured by kinect sensor in realtime and it will produce joint skeleton every frame. Then the frames will be processed to be recognized using Dynamic Time Warping method (DTW). The output of the system such as speech, which considered by the system as a translation of sign language that captured previously.

DTW method can be applied in the application with obtaining 92,78% accuracy of recognition and the average of computation time each input which recognited is 6.19760479 ms.

Keyword : Sign Language, Kinect, Dynamic Time Warping

1. Pendahuluan

Bahasa merupakan alat komunikasi yang sangat penting bagi kehidupan manusia, karena pada bahasa terdapat proses pertukaran informasi yang dapat menambah pemahaman manusia. Berdasarkan data dari World Health Organization (WHO) bulan Februari 2014 menyebutkan bahwa Lebih dari 5% dari populasi dunia yaitu sekitar 360 juta orang yang menyandang tunarungu [12]. Walaupun tidak dapat berbicara secara verbal, tetapi para penyandang tunarungu bisa berkomunikasi dengan nonverbal, yaitu menyampaikan maksud dengan bahasa isyarat. Kenyataannya tidak semua orang bisa mengerti berkomunikasi dengan menggunakan bahasa isyarat. Itulah yang menjadi kendala untuk orang normal ketika berkomunikasi dengan para penyandang tunarungu.

Bahasa Isyarat merupakan bahasa yang sering dipakai oleh kaum tunarungu. Bahasa isyarat adalah bahasa yang tidak menggunakan bunyi ucapan manusia atau tulisan dalam sistem perlambangannya [10]. Kaum tunarungu menggunakan bahasa ini mengombinasikan bentuk tangan, orientasi dan gerak tangan, lengan, dan tubuh, serta ekspresi wajah untuk mengungkapkan pikiran mereka. Bahasa Isyarat Indonesia adalah isyarat-isyarat yang pada mulanya diambil dari isyarat-isyarat yang disampaikan oleh tunarungu yang bisa diterima sebagai kosa kata dalam Bahasa Indonesia, dan juga termasuk American Sign Language (ASL) atau Bahasa Isyarat Amerika yang di-Indonesiakan[6].

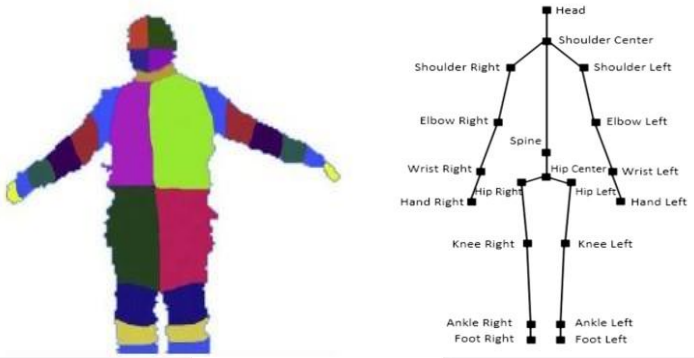
Kinect mempunyai kemampuan dasar seperti mendapatkan *depth image* dan *skeleton* bisa dikembangkan sehingga mampu mengenali suatu pola gerakan (*gesture recognition*). Namun, gesture recognition bukanlah fitur bawaan *Kinect*, sehingga perlu menggabungkan dengan metode pengenalan pola seperti *Dynamic Time Warping*, yang biasa dipakai untuk *speech recognition*[5], *handwriting recognition* [2] dan *gesture recognition* [3].

2. Dasar Teori

2.1. *Kinect* [1][8][13]

Skeletal Tracking memungkinkan *Kinect* untuk mengenali *user* dan mengikuti pergerakannya. Dengan menggunakan kamera inframerah(IR), *Kinect* dapat mengenali sampai dengan enam *user* dalam jangkauan . Dari jumlah tersebut, dua *user* dapat dikenali hingga detail. Pemanfaatan *Skeletal Tracking* pada suatu aplikasi dapat memberikan posisi sendi (*skeleton* joint) dari *user* yang dikenali dan mengikuti pergerakannya dari waktu ke waktu. *Skeletal Tracking* dioptimalkan untuk mengenali *user* yang berdiri ataupun duduk, dan menghadap *Kinect*. Dibawah ini adalah gambar yang *user* yang dikenali.

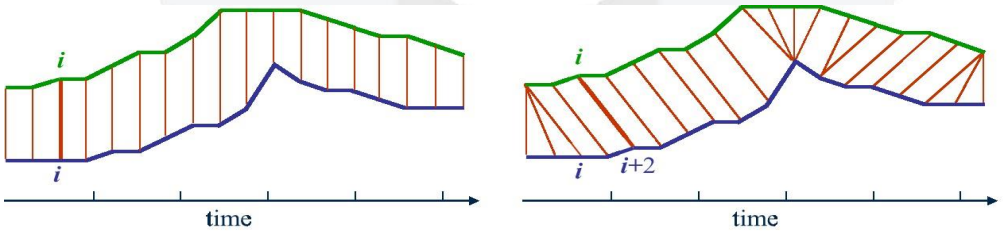
Kinect SDK menyediakan API(*Application Programming Interface*) untuk memudahkan mengakses seluruh titik sendi. Dua puluh titik sendi yang dapat diakses dan diidentifikasi sesuai dengan nama sendinya. Pada gambar 2.1(b) yang merepresentasikan seluruh *skeleton user* yang menghadap *kinect* membentuk dua puluh titik sendi.



Gambar 2. 1 (a) Pemetaan Badan Pemain (kiri) , (b) *Skeleton user* (kanan)

2.2. *Dynamic Time Warping*[7][9][11]

Dynamic Time Warping (DTW) adalah metode untuk menemukan keselarasan optimal antara dua deret waktu dengan beberapa pembatasan tertentu. Secara intuitif deret diselaraskan secara non-*linear* (elastis) satu sama lain. Berikut ini contoh gambaran penyelarasan 2 deret :



Gambar 2. 2 (a) Penyelarasan secara *linear* dan (b) *non-linear*

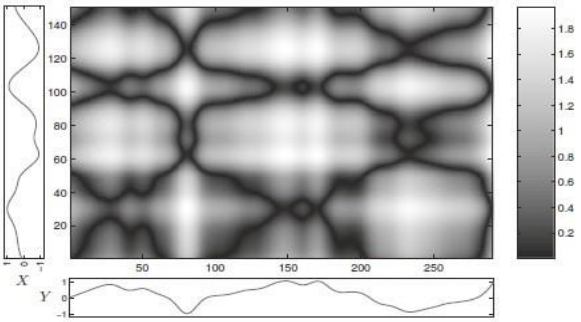
Pada gambar 2.2a dapat dilihat titik ke-*i* pada deret harus diselaraskan dengan titik ke-*i* deret lainnya. Sedangkan pada gambar 2.2b keselarasan dapat dilakukan secara non-*linear* (elastis), memungkinkan penyelarasan berdasarkan pencocokan bentuk walau berbeda fasa dalam axis waktu.

Objek DTW adalah untuk membandingkan dua deret *X* dengan panjang *N* dan *Y* dengan panjang *M*.

$$X = x_1, x_2, \ldots, x_n, \ldots x_N \qquad N \in \mathbb{N}$$
$$Y = y_1, y_2, \ldots, y_m, \ldots y_M \qquad M \in \mathbb{N} \qquad (1)$$

Metode DTW dimulai dengan membuat suatu matriks *C* dengan dimensi *NxM* yang merepresentasikan pasangan jarak antar himpunan *X* dan *Y* . Matriks jarak ini disebut matriks *local cost* menggunakan *Euclidean*

distance atau menggunakan *Manhattan distance.* Pada gambar dibawah ini adalah contoh nilai matriks *local cost.*



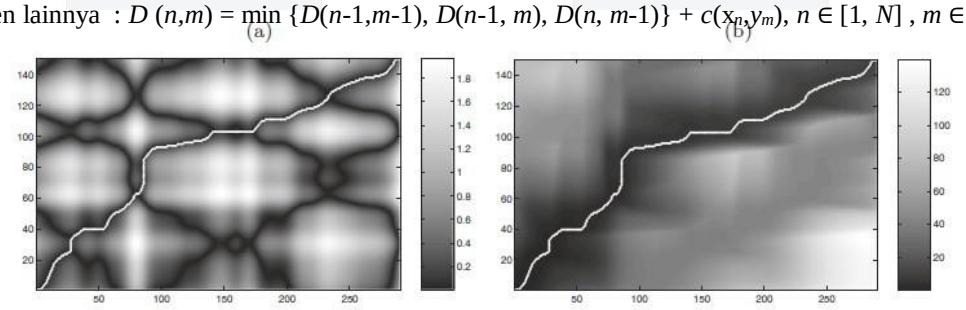
Gambar 2. 3 Matrik *local cost* dari deret *X* (sumbu vertikal) dan *Y* (sumbu horizontal). Daerah *cost* rendah ditunjukkan dengan warna gelap, sedangkan daerah *cost* tinggi ditunjukkan dengan wana terang.

Secara intuitif dapat dilihat keselarasan yang optimal yaitu yang jalur melalui daerah "lembah" *cost* rendah pada matriks C. *Warping path* adalah deret $P = p_1,p_2,...,p_l,..p_L$ yang melalui daerah yang bernilai *cost* rendah pada matriks *C* . Panjang dari *warping path* adalah *L* dan elemen ke-*l* dari *warping path* didefinisikan $p_l = (n_l,m_l) \in [1:N] \times [1:M]$ untuk $k \in [1:L]$. Dengan demikian ada banyak *warping path*, jumlah *warping path* yang didapat akan semakin banyak berbanding lurus dengan jumlah data dari *X* dan *Y* . Namun tujuannya sebenarnya adalah menentukan *optimal warping path* (*p**) , yaitu *warping path* dengan nilai *cost* keseluruhan yang paling rendah dari segala kemungkinan *warping path* yang ada. Untuk mendapatkannya diperlukan untuk mengecek satu persatu nilai *cost* keseluruhan dari *warping path* yang ada, masalah ini bisa jadi tantangan komputasi. Tetapi mengatasi masalah komputasi tersebut DTW menerapkan *Dynamic Programming* yang menggunakan fungsi jarak DTW

$$DTW(X,Y) := c_{p^*}(X,Y) = \min \{ c_p(X,Y), p \text{ adalah } warping\ path \} \quad (2)$$

Kemudian membuat Matriks Akumulasi *cost D* yang setiap elemennya didefinisikan sebagai berikut:
1.Kolom pertama : $D(n,1) = \sum_{k=1}^{n} c(x_k, y_1)$, $n \in [1, N]$. Baris pertama : $D(1,m) = \sum_{k=1}^{m} c(x_1, y_k)$, $m \in [1, M]$.

2.Semua elemen lainnya : $D(n,m) = \min \{D(n\text{-}1,m\text{-}1), D(n\text{-}1, m), D(n, m\text{-}1)\} + c(x_n,y_m)$, $n \in [1, N]$, $m \in [1, M]$.



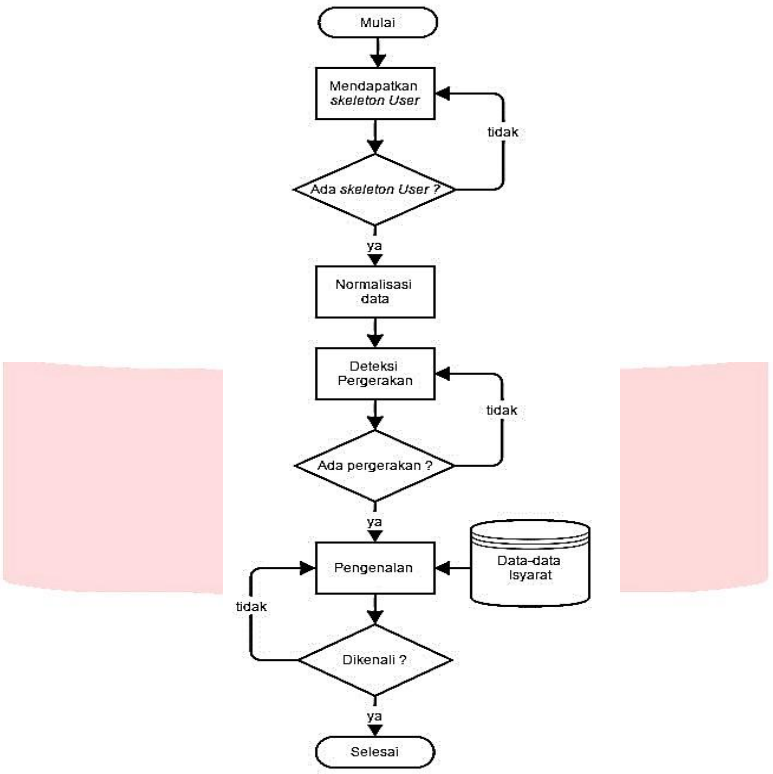
Gambar 2. 4 (a) Matriks *local cost* dan (b) Matriks akumulasi *cost* dengan *Optimal Warping Path* (garis putih)

Pada gambar diatas dapat dilihat pada matrik local cost ada banyak warping path dan hanya satu yang optimal, sedangkan pada matriks akumulasi cost hanya ada satu warping path dan sudah yang optimal.

Setelah matriks Akumulasi *cost D* dibuat , *optimal warping path* bisa didapatkan dengan *backtracking* dari *point* terakhir p_{end} = (M,N) ke *point* awal p_{start} = (1,1). Jika hanya ingin mendapatkan *cost optimal warping path* ($c_{p^*}(X,Y)$) yang telah terakumulasi bisa didapatkan dari *cost* pada elemen *D(N,M).*

3. Perancangan

Pada tugas akhir ini akan dibuat Aplikasi pada PC yang berbasis *Kinect* dengan masukan bahasa isyarat. Isyarat yang diperagakan akan ditangkap oleh sensor *Kinect* untuk diolah, kemudian dikenali. Pengolahan data pada Aplikasi berupa *frame-frame* yang berisi *skeleton user* dengan jumlah deret sebanyak 32 *frame*. Proses pengenalan akan dibuat secara *realtime,* yaitu prosesnya tidak mengalami penyimpanan ke dalam *hardisk* terlebih dahulu melainkan diolah langsung, kemudian dikeluarkan hasilnya. Pada gambar berikut ini diagram alir Aplikasi:



Gambar 3. 1 Diagram alir Aplikasi

Pada gambar 3.1 proses dimulai dari mendapatkan skeleton dari user. Pada proses tersebut juga sebagi penanda apakah user sudah dideteksi atau tidak. Kemudian apabila data skeleton dari user ada maka dilanjutkan untuk dinormalisasi. Hasil dari normalisasi data digunakan untuk mendeteksi pergerakan, dimana user yang awalnya diam dan kemudian bergerak, maka sistem akan mendeteksi sebagai suatu gerakan sehingga data gerakan itu saja yang dikumpulkan untuk dilanjutkan ke proses pengenalan. Pada proses pengenalan akan diukur kemiripan gerakan dengan data-data referensi yang telah ada, apabila ditemukan maka keluaran akan dikeluarkan dalam berntuk teks dan suara.

2.1 Normalisasi Data[4]

User yang melakukan gerakan bahasa isyarat pada dasarnya bebas melakukannya dimana saja selama dalam jangkauan *Kinect*. Karena posisi *user* tidak pasti maka perlu melakukan normalisasi dari data *skeleton user* yang diperoleh untuk mengembalikan posisi *skeleton* ke tengah sumbu tiga dimensi dan menyamakan skala ukuran *skeleton*. Untuk mengembalikan posisi *skeleton user* ke posisi normal , maka dilakukan translasi terhadap titik koordinat sendinya dengan persamaan berikut:

$$= - \left(\frac{ - }{ - } \right)$$

$$= - \left(\quad \right)$$

$$= - \left(\frac{ - }{2} \right) \qquad (3)$$

Setelah melakukan translasi dilanjutkan dengan dilatasi koordinatnya. Dilatasi perlu dilakukan agar ukuran *skeleton* yang berbeda-beda menjadi tidak masalah. Persamaan dilatasi adalah sebagai berikut :

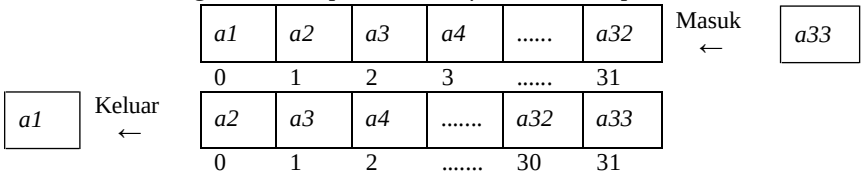
$$= \frac{}{\sqrt{(-)^2 - (-)^2 - (-)^2}}$$

$$= \frac{}{}$$

$$\cos\gamma = \frac{\sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2 - (z_2 - z_1)^2}}{\sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2 - (z_2 - z_1)^2}} \quad (4)$$

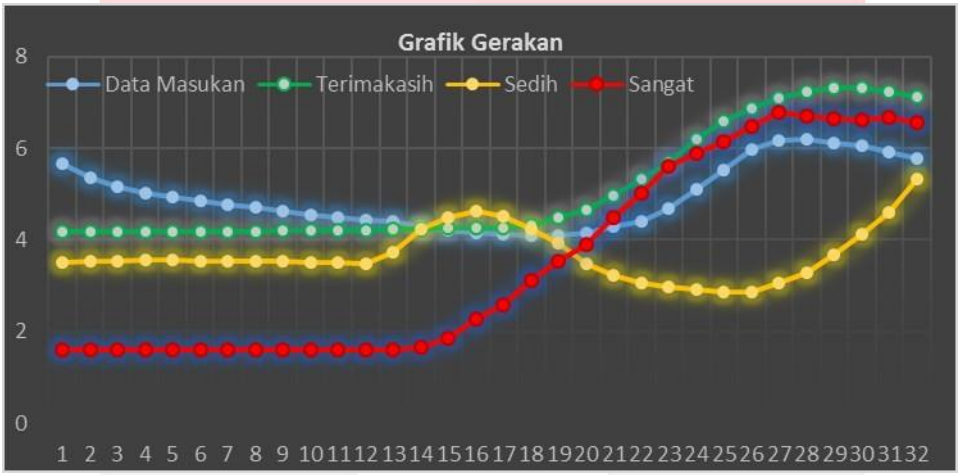
2.2 Pengenalan Pola Gerak dengan *Dynamic Time Warping*

Setelah data titik sendi dinormalisasi, kemudian ditampung dalam sebuah *array* secara berurut dengan panjang *array* 32 *frame*. Data yang ditampung pada *array* tersebut bersifat FIFO (First In First Out) , dimana *frame* pertama masuk akan dibuang ketika ada penambahan *frame* baru di posisi terakhir.



Gambar 3. 2 Deret *frame*

Kemudian dilanjutkan untuk mengukur jarak kemiripan antara data masukan dengan data-data referensi yang ada, bisa dicontohkan pada grafik data masukan dan data-data referensi pada gambar 3.3.



Gambar 3. 3 Grafik gerakan

Kurva warna biru adalah data masukan, sedangkan yang lainnya adalah beberapa data referensi. DTW di sini berfungsi untuk mengukur kemiripan antara pola data masukan dengan data referensi satu persatu. Suatu pola masukan dinyatakan mirip dengan suatu data referensi apabila hasil jarak DTWnya menunjukkan jarak yang paling minimum dibandingkan data referensi lainnya dan juga masih di dalam batas toleransi (threshold). Namun untuk lebih mengoptimalkan komputasi, perlu membatasi jumlah data referensi yang akan dibandingkan, sehingga data realtime akan dibandingkan hanya dengan beberapa data referensi saja. Untuk hal itu setiap frame pertama pad data realtime akan dibanding dengan frame pertama setiap data referensi menggunakan *Euclidean distance* dengan syarat hasil *Euclidean Distance* tersebut masih dibawah nilai *threshold*, sehingga didapatkan beberapa data referensi saja yang akan dibandingakan keseluruhan dengan DTW.

Cara kerja metode *Dynamic Time Warping* yaitu mengukur jarak 2 pola, $A = [a_1, a_2, a_3 \ldots, a_{32}]$ dan $B = [b_1, b_2, b_3 \ldots, b_{32}]$. Dimana setiap $a = [a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}]$ dan $b = [b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}, b_{16}]$

A adalah pola gerak yang langsung diperagakan *user* sedangkan B adalah data referensi untuk pembandingnya. Setiap indeks elemen A dan B merupakan representasi dari nomor *frame*nya dan setiap *frame* mengandung *array* bersisi posisi 6 titik sendi yang telah diekstrak. Sedangkan indeks yang ada pada elemen *frame* a dan b merupakan posisi joint, angka *1* menunjukkan *left hand, 2 whrist left, 3 elbow left, 4 elbow right, 5 whrist right, 6 hand right,* sedangkan x, y, z menunjukkan sumbu axis masing-masing.

Setelah mendapatkan dua data yang akan diukur jarak kesamaannya, selanjutnya menentukan matriks *Accumalated cost* dan menghitung *cost* DTW yang dinormalisasi. Akan tetapi agar menambah performansi perlu dilakukan kustomisasi pada *slope P= 1* dan weighting koefisien horizontal = 1, vertikal =1 dan diagonal = 2 [5].

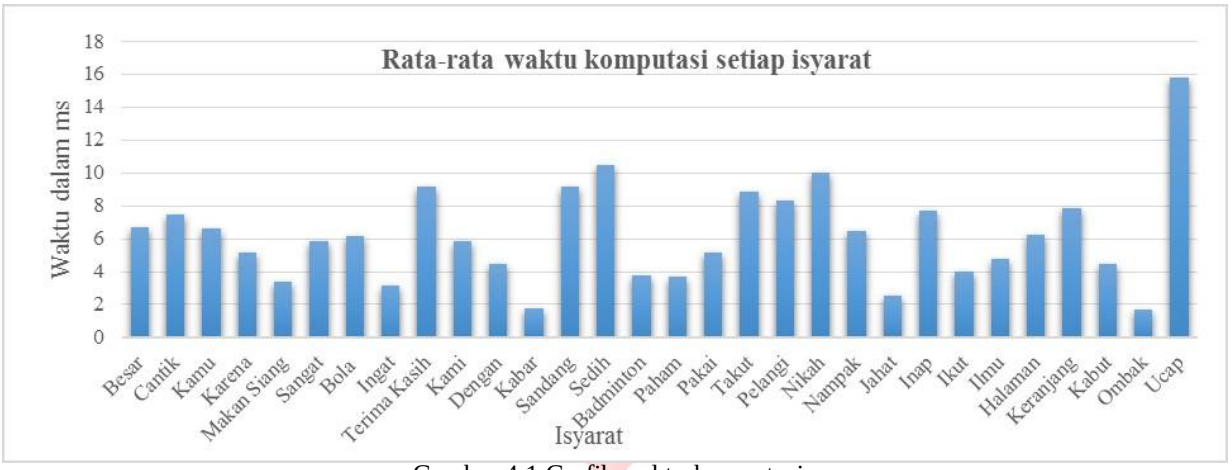
4. Pengujian

Pada pengujian dilakukan oleh 30 orang dengan berbagai ukuran badan, serta bentuk tubuh. Setiap penguji masing-masing memperagakan 6 isyarat. Pengujian ini bertujuan untuk mengetahui bagaimana performansi dari

Aplikasi yaitu waktu komputasi dan akurasi. Skenario untuk pengujiannya yaitu user memperagakan isyarat sebanyak 3 kali, apabila minimal dua kali benar dianggap benar.

4.1 Pengujian Waktu Komputasi Sistem

Berikut ini adalah grafik hasil pengujian waktu komputasi rata-rata setiap isyarat :

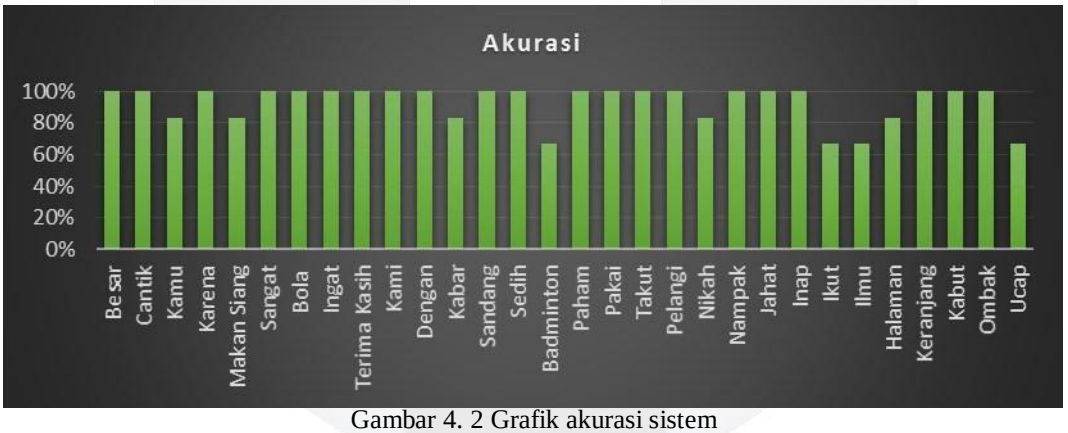


Gambar 4.1 Grafik waktu komputasi

Pada gambar 4.1, didapatkan bahwa kecepatan komputasi terkecil untuk pengenalan yaitu 1 ms dan yang terbesar 32 ms. Waktu komputasi rata-rata untu kata "ucap" memiliki waktu komputasi yang lebih tinggi diantara kata lain. Hal tersebut dikarenakan frame pertama kata "ucap" mirip dengan beberapa frame pertama dari kata lain. Untuk kecepatan komputasi rata-rata keseluruhan isyarat yaitu 6,19760479 ms.

4.2 Pengujian Akurasi Sistem

Berikut ini adalah grafik hasil pengujian akurasi sistem :



Gambar 4. 2 Grafik akurasi sistem

Pada chart dibawah ini, diperlihatkan bagaimana pengaruh jumlah gerakan referensi terhadap akurasi sistem.

Gambar 4. 3 Grafik akurasi sistem berdasarkan jumlah isyarat yang ada

Pada gambar 4.3 dapat dilihat akurasi sistem menurun ketika data referensi semakin banyak. Hal tersebut disebabkan karena semakin beratnya komputasi yang dilakukan dan banyaknya isyarat yang mirip .Untuk akurasi sistem keseluruhan dari 180 kali percobaan dengan masing-masing tiap isyarat 6 kali, maka didapatkan akurasi rata-rata sebesar 92,78%.

5. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa:

1. Berdasarkan pengujian beta menyatakan bahwa aplikasi ini bermanfaat (87%), mudah digunakan (73%) dan mempunyai respons sistem baik (84%).
2. Metode Dynamic Time Warping berhasil diterapkan pada Aplikasi penerjemah bahasa isyarat menjadi suara menggunakan sensor Kinect secara realtime dan memberikan akurasi hingga 92,78% .
3. Waktu rata-rata untuk memproses pengenalan yaitu 6,19760479 ms.
4. Ambang batas yang baik pada proses pengenalan pada Aplikasi ini yaitu 0,932553904, nilai lebih kecil mengakibatkan kesulitan saat pencocokan, sedangkan nilai lebih besar memberikan keluaran yang tidak akurat.

Daftar Pustaka

[1] A. Jana, *Kinect* for Windows SDK Programming Guide, Birmingham: Packt Publishing Ltd., 2012.
[2] A. Efrat, Q. Fan, dan S. Venkatasubramanian, “Curve matching, time warping, and light fields: New algorithms for computing similarity between curves," J. Math. Imaging Vis., vol. 27, no. 3, pp. 203
[3] A. Corradini, "Dynamic time warping for online recognition of a small gesture vocabulary," in RATFG-RTS '01: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01). Washington, DC, USA: IEEE Computer Society, 2001.
[4] A. Santoso, Perancangan Program Aplikasi Pembelajaran Bahasa Isyarat Dengan Metode Dynamic Time Warping, Jakarta: Universitas Bina Nusantara, 2013.
[5] H. Sakoe dan S. Chiba, “Dynamic Programming Algorithm Optimization for Spoken Word Recognition,” *IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING*, 1978.
[6] Imas A. R. Gunawan, Kamus Umum Bahasa Isyarat Indonesia, Jakarta: Lembaga Komunikasi Total Indonesia, 1996.
[7] M. Muller, Information Retrieval for Music and Motion, Berlin: Springer, 2007.
[8] Microsoft, “*Skeletal Tracking*,” [Online]. Available: http://msdn.microsoft.com/en-us/library/hh973074.aspx. [Diakses 21 Oktober 2014].
[9] N. E. Gillian, "*Dynamic Time Warping*," in *Gesture Recognition for Musician Computer Interaction*, Queen’s University Belfast , 2011.
[10] Pusat Bahasa, Kamus Besar Bahasa Indonesia, Jakarta: Balai Pustaka, 1988.
[11] P. Senin, “*Dynamic Time Warping* Algorithm Review,” 2008.
[12] WHO Media centre, “Deafness and hearing loss,” WHO, Februari 2014. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs300/en/.[Diakses 3 Agustus 2014].
[13] W. Jaret dan J. Ashley, Beginning *Kinect* Programming with The Microsoft *Kinect* SDK, Apress.