

# Analisis dan Implementasi Algoritma Klasifikasi *Random Forest* Dalam Pengenalan Bahasa Isyarat Indonesia (BISINDO)

Muhammad Rafii Amiarrahman<sup>1</sup>, Tri Handhika<sup>2</sup>

<sup>1</sup>Teknik Informatika, Universitas Gunadarma

<sup>2</sup> Pusat Studi Komputasi Matematika, Universitas Gunadarma

E-mail: <sup>1</sup>[\\*amiarrahman@gmail.com](mailto:*amiarrahman@gmail.com), <sup>2</sup>[trihandika@staff.gunadarma.ac.id](mailto:trihandika@staff.gunadarma.ac.id)

**Abstrak** – Bahasa Isyarat Indonesia (BISINDO) adalah salah satu bahasa isyarat yang digunakan sebagai sistem komunikasi praktis dan efektif bagi penderita tuli di Indonesia. Namun demikian, tidak semua orang dengan kemampuan mendengar dapat memahami BISINDO. Untuk menjembatani komunikasi penderita tuli dengan orang-orang yang memiliki kemampuan mendengar diperlukan suatu teknologi yang dapat menerjemahkan gerakan isyarat pada BISINDO menjadi Bahasa Indonesia yang dipahami oleh orang dengan kemampuan mendengar. Pada makalah ini dilakukan analisis dan implementasi algoritma klasifikasi *Random Forest* untuk mengenali BISINDO dengan bantuan rekaman gerakan isyarat yang diperoleh dari sensor *Kinect*. Data rekaman tersebut kemudian diolah dengan menggunakan *skeleton tracking* untuk menghasilkan titik koordinat yang diperlukan. Adapun algoritma klasifikasi *Random Forest* menghasilkan rata-rata akurasi sebesar 96% dalam mengenali beberapa kata dalam BISINDO.

**Kata Kunci** — BISINDO, *Gesture Recognition*, *Kinect*, *Random Forest*, *Skeletal Tracking*.

## 1. PENDAHULUAN

Dengan adanya penemuan sensor kedalaman, pengenalan isyarat (*gesture recognition*) mendapatkan perhatian yang signifikan dalam bidang *computer vision* serta interaksi manusia dan komputer. *Gesture recognition* adalah salah satu cabang ilmu komputer dan teknologi bahasa dalam mengenali gerak tubuh manusia yang melibatkan jari tangan, tangan, lengan, kepala, wajah, atau badan dengan maksud untuk menyampaikan informasi ataupun berinteraksi dengan lingkungan [1].

Pengenalan isyarat ini dapat digunakan untuk menjembatani komunikasi antara kaum tuli dengan orang-orang yang memiliki kemampuan mendengar dengan memanfaatkan bahasa nonverbal. Di Indonesia terdapat dua bahasa isyarat yang digunakan, yaitu Bahasa Isyarat Indonesia (BISINDO) dan Sistem Bahasa Isyarat Indonesia (SIBI). BISINDO adalah sistem komunikasi yang praktis dan efektif untuk kaum tuli Indonesia karena dikembangkan oleh mereka sendiri. Sedangkan, SIBI adalah sistem hasil rekayasa dan ciptaan orang normal untuk berkomunikasi dengan kaum tuli sehingga dalam menerjemahkan satu kata perlu dilengkapi dengan awalan dan akhiran. Hal ini dirasa cukup sulit bagi kaum tuli yang tidak pernah mengenal tata bahasa Indonesia.

Jumlah penerjemah dan besarnya biaya menyebabkan tidak semua kaum tuli dapat dilayani serta didampingi oleh para penerjemah. Beberapa orang dengan kemampuan mendengar pun hanya mampu berbahasa isyarat sebatas agar dapat berkomunikasi dengan keluarga serta kerabat mereka yang tuli.

Pada makalah ini dilakukan analisis algoritma klasifikasi *Random Forest* dalam mengenali BISINDO yang diperoleh melalui *skeleton tracking* pada sensor *kinect*. Terdapat 5 (lima) kata umum yang telah ditentukan untuk diuji akurasi dari algoritma tersebut, diantaranya: “Hari”, “Pohon”, “Ibu”, “Aku”, dan “Kamu”. Setiap kata diperagakan sebanyak 5 kali secara berulang dengan posisi duduk dan direkam menggunakan sensor *kinect*. Peragaan ke-1 hingga ke-4 digunakan untuk pelatihan data sedangkan peragaan ke-5 digunakan dalam pengujian data.

## 2. METODE PENELITIAN

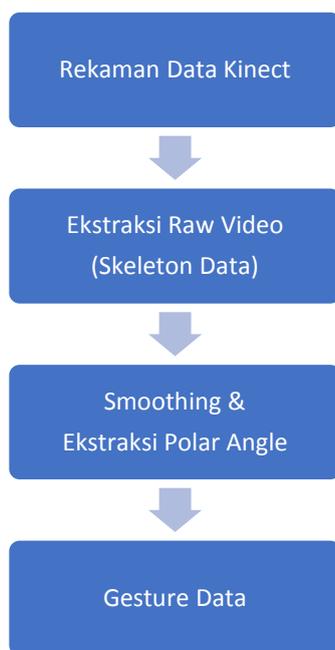
Data yang digunakan dalam penelitian ini diperoleh dengan memanfaatkan perangkat sensor *kinect* [2], [3], [4], [5] serta perangkat lunak Weka dimana sensor *kinect* digunakan sebagai media masukan untuk menangkap citra atau gambar serta kedalaman. Selanjutnya, data tersebut diolah hingga menghasilkan koordinat *skeleton* dan sudut diantara dua vektor yang terdapat pada tubuh menggunakan *skeleton tracking*. Data yang diperoleh selanjutnya diuji menggunakan sebuah paket *tools machine learning* praktis Weka untuk mendapatkan nilai akurasi dari algoritma klasifikasi *Random Forest* [6], [7].

### 2.1. Diagram Alur Sistem

Diagram alur sistem terbagi menjadi dua bagian, Gambar 1 merupakan diagram alur pelatihan data dan Gambar 2 merupakan diagram alur algoritma klasifikasi *Random Forest*. Alur dari pelatihan data yang dibangun dimulai dari

pengumpulan data gerakan bahasa isyarat menggunakan sensor *kinect* dengan melalui beberapa tahap proses ekstraksi *raw video* [8], [9], pelembutan (*smoothing*) dan ekstraksi sudut antara dua vektor hingga mendapatkan data gerakan BISINDO yang dapat diolah [10].

Data yang didapat berupa angka-angka yang berisikan 10 (sepuluh) titik koordinat *skeleton* dengan posisi duduk terhadap sumbu x, y, z. Penggunaan pelacakan *skeleton* pada *user* dengan posisi duduk lebih tepat jika dibandingkan dengan posisi berdiri, karena keseluruhan *gesture* yang diperagakan hanya tubuh bagian atas saja yang bergerak. Perlu diketahui bahwa setiap data memiliki perbedaan jumlah *record*. Hal ini disebabkan oleh keberhasilan pengumpulan data yang direkam ketika *skeleton tracking* dalam merekam *skeleton* pada *user* dimana masing-masing data gerakan (*frame*) memiliki waktu yang berbeda pada setiap data rekamannya, yaitu sekitar 30 *f/s*.



Gambar 1. Diagram Alur Pelatihan Data

Setelah data diperoleh dari proses pelatihan, langkah selanjutnya adalah menganalisis algoritma klasifikasi *Random Forest*. Algoritma ini merupakan sebuah *classifier* dengan *ensemble method* yang terdiri atas beberapa pohon keputusan (*decision tree*) hingga membentuk suatu “hutan” (“*forest*”) secara acak (*random*) [6]. Algoritma klasifikasi *Random Forest* digunakan pada penelitian ini karena memiliki akurasi yang tinggi pada pengenalan bahasa isyarat SIBI [10]. Gambar 2 merupakan diagram alur dari algoritma klasifikasi *Random Forest*.



Gambar 2. Diagram Alur Random Forest

Berikut ini dijelaskan alur algoritma klasifikasi *Random Forest* dari Gambar 2 [11]:

- 1) Pertama-tama lakukan penentuan jumlah *n* pohon (*decision tree*) yang ingin dibentuk.
- 2) Lakukan teknik *bagging* pada data gerakan BISINDO melalui pemilihan sampel berulang kali, dengan pengembalian. Adapun sampel yang diambil secara acak adalah sebesar 60% dari data yang tersedia.
- 3) Pada masing-masing pohon dipilih atribut terbaik menggunakan *gini index* untuk kemudian di-*split* pada setiap *node* menjadi 2 simpul anak.
- 4) Jika *node* belum homogen, lakukan pemilihan atribut pada data yang belum homogen secara berulang dengan menggunakan *gini index* hingga menjadi satu pohon keputusan.
- 5) Lakukan langkah ke-2 sampai ke-5 secara berulang hingga menghasilkan *n* pohon yang diinginkan.
- 6) Lakukan *voting* dengan menghitung probabilitas hasil tiap *tree* dalam *forest* kemudian merata-ratakannya dan menjadikan hasil prediksi dengan nilai probabilitas tertinggi.

Setelah algoritma klasifikasi *Random Forest* dijalankan, selanjutnya hitung keakurasian dari algoritma tersebut dalam mengenali data gerakan BISINDO.

### 2.2. Pengumpulan Data

Pengumpulan data diperagakan langsung oleh kaum tuli untuk 5 (lima) kata umum yang diuji, diantaranya: “Hari”, “Pohon”, “Ibu”, “Aku”, dan “Kamu”, masing-masing sebanyak 5 (lima) kali. Pengambilan data ini memanfaatkan aplikasi yang telah disediakan oleh *kinect developer toolkit*. Adapun *tools* yang digunakan adalah *kinect studio* untuk merekam, menyimpan data rekaman, serta menampilkan kembali hasil rekaman yang telah diperoleh. *Tools* kedua adalah *kinect explorer* yang digunakan untuk menampilkan data dari sensor *kinect* secara visual. Gambar 3(a) – 3(e) merupakan proses pengumpulan data gerakan BISINDO.



Gambar 3(a). Proses Pengumpulan Data “Hari”



Gambar 3(b). Proses Pengumpulan Data “Pohon”



Gambar 3(c). Proses Pengumpulan Data “Ibu”



Gambar 3(d). Proses Pengumpulan Data “Aku”



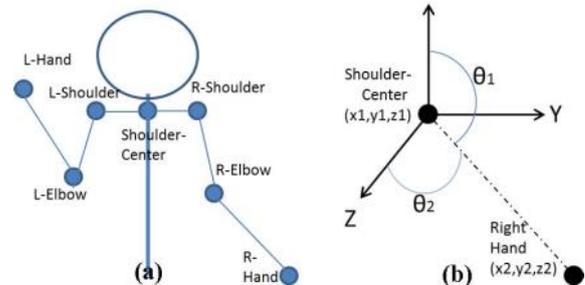
Gambar 3(e). Proses Pengumpulan Data “Kamu”

### 2.3. Pengolahan Data

Gerakan *skeleton* yang dilacak diperoleh dari titik sendi dan sudut antara dua vektor yang dihasilkan oleh sensor *kinect*, seperti yang terlihat pada Gambar 4(a). Adapun sudut antara dua vektor dihasilkan dari titik-titik sendi yang terpusat pada titik sendi antar bahu, seperti terlihat pada Gambar

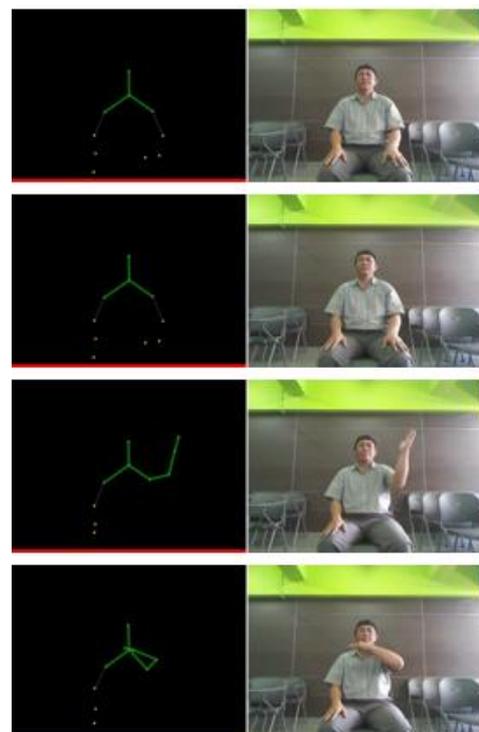
4(b), dengan menggunakan fungsi *dot product* pada persamaan (1) dimana  $\theta$  merupakan sudut yang dicari, sedangkan  $\vec{a}$  dan  $\vec{b}$  masing-masing adalah vektor yang membentuk sudut  $\theta$  [10].

$$\theta = \text{Arccos}(\vec{a} \cdot \vec{b}) \dots \dots \dots (1)$$



Gambar 4. Titik Sendi Pada *Kinect Skeleton Data* [10]

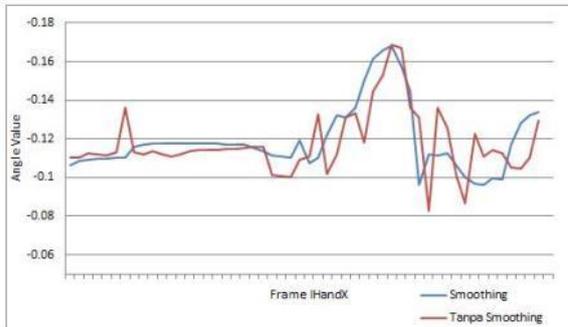
Gambar 5 menunjukkan contoh rekaman gerakan *skeleton* untuk kata “Hari” yang diperoleh dari sensor *kinect*. Terlihat bahwa sensor *Kinect* dapat mendeteksi titik sendi seperti telah dijelaskan sebelumnya sebagai acuan dalam pengenalan gerakan BISINDO.



Gambar 5. Gerakan *skeleton* dari Data “Hari”

Pengolahan data kemudian dilanjutkan dengan proses pelembutan (*smoothing*), seperti ditunjukkan

dengan grafik pada Gambar 6. Grafik berwarna merah merupakan *skeleton joint* yang belum di-*smoothing*, sementara grafik biru menunjukkan hasil *smoothing* dari *skeleton joint* tersebut.



Gambar 6. Hasil *Smoothing Skeleton Data*

Data pada proses *smoothing skeleton* didapat dari pelatihan data. Dapat dilihat pada Gambar 7 merupakan kumpulan data *gesture* yang didapat dari proses pelatihan data. Data ini yang digunakan untuk membangun pohon keputusan pada algoritma klasifikasi *Random Forest*.

No.	1: AngleRight	2: AngleRightE	3: AngleLeft	4: AngleLeftE	5: Class Kata
22	135.0	38.0	129.0	32.0	Transisi
23	135.0	38.0	129.0	32.0	Transisi
24	135.0	38.0	129.0	32.0	Transisi
25	135.0	38.0	129.0	32.0	Transisi
26	135.0	38.0	129.0	32.0	Transisi
27	135.0	38.0	129.0	32.0	Transisi
28	135.0	38.0	129.0	32.0	Transisi
29	135.0	38.0	129.0	32.0	Transisi
30	135.0	38.0	129.0	32.0	Transisi
31	135.0	38.0	129.0	32.0	Transisi
32	135.0	38.0	129.0	32.0	Transisi
33	135.0	38.0	129.0	32.0	Transisi
34	135.0	38.0	129.0	32.0	Hari
35	133.0	38.0	129.0	32.0	Hari
36	132.0	39.0	129.0	32.0	Hari
37	152.0	49.0	129.0	32.0	Hari
38	146.0	51.0	130.0	32.0	Hari
39	132.0	50.0	130.0	33.0	Hari
40	117.0	49.0	130.0	33.0	Hari
41	109.0	53.0	130.0	33.0	Hari

Gambar 7. *Dataset Gesture Hari*

Berdasarkan Gambar 7 dapat disimpulkan bahwa data *gesture* bahasa isyarat Indonesia (BISINDO) menghasilkan 35 atribut, diantaranya:

- 4 atribut untuk sudut antara dua vektor (sudut bahu kiri, bahu kanan, sudut siku kiri, siku kanan).
- 1 atribut berisikan kelas kata.

- 30 atribut untuk koordinat *skeleton joint* dengan posisi duduk (kepala, bahu, siku, lengan, dan pergelangan tangan) terhadap sumbu (x, y, z).

Berdasarkan keterangan ini, 10 koordinat titik *skeleton* didapat dari *skeleton tracking* dengan posisi duduk terhadap sumbu x, y, z. Sehingga menghasilkan 30 atribut. Penggunaan posisi duduk lebih tepat jika dibandingkan dengan penggunaan posisi berdiri yang memiliki 20 koordinat titik *skeleton*. Karena keseluruhan *gesture* yang diperagakan, tubuh bagian atas saja yang bergerak, maka dari itu pelacakan dengan posisi duduk lebih baik.

Setiap peraga tentunya memiliki tinggi dan bentuk badan yang berbeda. Ini mempengaruhi nilai koordinat titik *skeleton*, maka dari itu penggunaan sudut antar dua vektor dapat menyelesaikan permasalahan ini dengan menggunakan nilai sudut yang dibentuk dari pergerakan *skeleton*. Nilai sudut ini tidak terpengaruh terhadap tinggi dan bentuk badan peraga yang berbeda. Sementara 1 atribut (kelas kata) berguna untuk membedakan (*frame*) dari kata satu dengan yang lain.

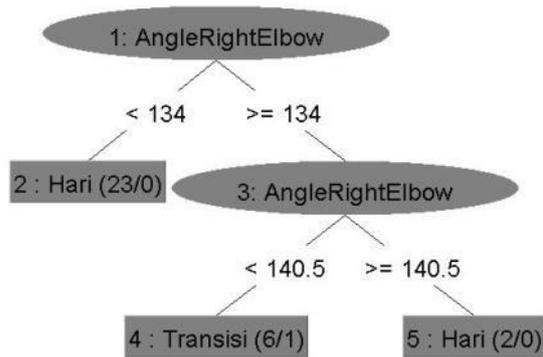
#### 2.4. *Random Forest*

Data *gesture* yang didapat pada proses pelatihan data digunakan untuk membangun pohon keputusan pada algoritma klasifikasi *Random Forest*. Data dikelompokkan berdasarkan nilai data yang homogen untuk setiap *record* satu dengan yang lain.

Pembangunan pohon keputusan memanfaatkan pencarian *gini index* terkecil. Nilai *gini index* terkecil akan menjadi *root node* dan di-*split* menjadi 2 simpul anak. Rumus pencarian *gini index* dapat dilihat pada persamaan (2), dimana  $P_i$  adalah rasio jumlah data yang diberi label kelas  $i$  dalam  $D$ . Jika data  $D$  dibagi kedalam dua *subset*, *gini index* dari data yang terbagi dari kelas  $m$ .

$$Gini Index (D) = 1 - \sum_{i=1}^m P_i^2 \dots\dots\dots(2)$$

Setelah nilai *gini index* terkecil didapat dan *node* di-*split* menjadi 2 simpul anak hingga *node* di setiap pohon keputusan menjadi homogen. Dapat dilihat pada Gambar 8 merupakan salah satu pohon keputusan yang ada pada algoritma klasifikasi *Random Forest*.



Gambar 8. Pohon Keputusan Pada *Random Forest*

Pohon keputusan yang ada pada Gambar 8 adalah salah satu pohon keputusan yang digunakan untuk proses klasifikasi. Nilai atribut disetiap *frame* dicocokkan dengan nilai yang ada pada pohon keputusan, untuk setiap *frame* akan menghasilkan satu prediksi kata disetiap pohon keputusan. Karena *Random Forest* memiliki beberapa pohon keputusan maka prediksi dari semua pohon akan di-*voting* / dihitung probabilitas tertinggi untuk mendapatkan hasil klasifikasi.

### 3. HASIL DAN PEMBAHASAN

Uji coba dilakukan untuk melihat seberapa akurat algoritma klasifikasi *Random Forest* dalam mengenali data gerakan BISINDO. Proses pengujian dilakukan dari gabungan data percobaan pertama hingga keempat terhadap percobaan kelima. Sebelumnya perlu diketahui bahwa setiap data memiliki perbedaan jumlah *record*. Hal ini disebabkan oleh keberhasilan pengumpulan data yang direkam ketika *skeleton tracking* dalam merekam *skeleton* pada *user* dimana masing-masing data gerakan (*frame*) memiliki waktu yang berbeda pada setiap data rekamannya, yaitu sekitar 30 *f/s*.

Pada makalah ini, jumlah pohon keputusan yang dibangun dalam algoritma klasifikasi *Random Forest* adalah sebanyak 40 pohon. Sedangkan, penentuan jumlah *record* yang dipilih secara acak untuk membangun satu pohon keputusan adalah sebesar 60% dari data asli. Beberapa skenario pengujian pun dilakukan, untuk melihat hasil akurasi dari data yang diujikan. Dapat dilihat pada Tabel 1 dan Tabel 2 merupakan proses pengujian nilai akurasi pada data *gesture*.

Tabel 1 berisi ringkasan nilai akurasi algoritma klasifikasi *Random Forest* yang diperoleh dengan memanfaatkan pelembuatan (*smoothing*) *skeleton* tanpa menggunakan sudut antara dua vektor.

Tabel 1. Nilai Akurasi Algoritma Klasifikasi *Random Forest* dengan Memanfaatkan Pelembuatan (*Smoothing*) *Skeleton* tanpa Menggunakan Sudut Antara Dua Vektor

No	Kata	Akurasi (%)
1	Hari	53.125
2	Pohon	72.093
3	Ibu	77.2727
4	Aku	95
5	Kamu	54.5455

Berdasarkan Tabel 1, diketahui bahwa nilai akurasi tertinggi dari algoritma klasifikasi *Random Forest* dengan memanfaatkan pelembuatan (*smoothing*) *skeleton* tanpa menggunakan sudut antara dua vektor terjadi ketika pengenalan kata "Ibu", yaitu sebesar 77%. Sedangkan, nilai akurasi terendah terjadi pada pengenalan kata "Hari", yaitu sebesar 53%. Dengan demikian nilai rata-rata untuk pengujian pada Tabel 1 adalah 70%.

Sementara itu dilakukan uji coba yang kedua dengan memanfaatkan *smoothing skeleton* dan sudut antara dua vektor. Dapat dilihat pada Tabel 2 merupakan ringkasan nilai akurasi algoritma klasifikasi *Random Forest* yang diperoleh dengan memanfaatkan pelembuatan (*smoothing*) *skeleton* dan sudut antara dua vektor.

Tabel 2. Nilai Akurasi Algoritma Klasifikasi *Random Forest* dengan Memanfaatkan Pelembuatan (*Smoothing*) *Skeleton* dan Sudut Antara Dua Vektor

No	Kata	Akurasi (%)
1	Hari	95.3125
2	Pohon	95.3488
3	Ibu	97.7273
4	Aku	95
5	Kamu	97.7273

Berdasarkan Tabel 2, diketahui bahwa nilai akurasi tertinggi dari algoritma klasifikasi *Random Forest* dengan memanfaatkan pelembuatan (*smoothing*) *skeleton* dan sudut antara dua vektor terjadi ketika pengenalan kata "Ibu dan Kamu", yaitu sebesar 97%. Sedangkan, nilai akurasi terendah terjadi pada pengenalan kata "Aku", yaitu sebesar 95%. Dengan demikian nilai rata-rata untuk pengujian pada Tabel 2 adalah 96%. Oleh karena itu penggunaan *smoothing skeleton* dan sudut antara dua vektor sangat diperlukan untuk meningkatkan nilai akurasi pada algoritma klasifikasi *Random Forest*.

Adapun beberapa faktor yang dapat mempengaruhi akurasi dari algoritma klasifikasi *Random Forest* adalah sebagai berikut [11]:

1. Data yang terseleksi pada proses *bagging*.
2. Jumlah *record* yang digunakan pada proses pembangunan pohon keputusan dalam pelatihan data.
3. Pemilihan *node* disetiap pembangunan pohon keputusan.
4. Penentuan jumlah pohon keputusan yang dibangun.

#### 4. SIMPULAN

Berdasarkan analisis yang dilakukan dari hasil pengujian dapat ditarik kesimpulan sebagai berikut:

1. Dengan memanfaatkan *smoothing skeleton* dan sudut antara dua vektor diperoleh bahwa nilai akurasi algoritma klasifikasi *Random Forest* cukup bervariasi, berkisar antara 53.125% hingga 97.7273% untuk 5 (lima) macam kata umum, yakni "Hari", "Pohon", "Ibu", "Aku", dan "Kamu" dengan rata-rata tertinggi adalah 96%.
2. Pada proses pengumpulan data menggunakan sensor *kinect* perlu diperhatikan posisi dan jarak yang tepat untuk merekam gerakan. Jika gerakan yang dilakukan terlalu cepat proses *skeleton tracking* tidak mendapatkan hasil yang maksimal. Selain itu, perlu diberi jeda beberapa detik sebelum melakukan pergerakan gerakan. Hal ini berguna untuk proses *skeleton tracking* yang memerlukan waktu beberapa *millisecond* dalam melakukan pelacakan.

#### 5. SARAN

Pada makalah ini, implementasi algoritma klasifikasi *Random Forest* masih dilakukan per-*frame*. Padahal, suatu kata dalam BISINDO direpresentasikan dalam kumpulan *frame* yang bergantung pada waktu. Oleh sebab itu, untuk penelitian selanjutnya dapat diterapkan konsep *clustering* untuk mengumpulkan *frame-frame* tersebut terlebih dahulu sebelum dilakukan proses klasifikasi. Selain itu, penambahan informasi terkait pendeteksian jari tangan juga diharapkan dapat meningkatkan akurasi dari algoritma klasifikasi *Random Forest*.

#### DAFTAR PUSTAKA

- [1] Kumar, H. 2014. *Gesture Recognition Using Hidden Markov Models Augmented with Active Difference Signatures*. Tesis. New York: Rochester Institute of Technology.
- [2] Catuhe, D. 2012. *Programming with Kinect for Windows Software Development Kit*. Washington: Microsoft Press.
- [3] Jana, A. 2012. *Kinect for Windows SDK Programming Guide*. Birmingham: Packt Publishing Ltd.
- [4] Jean, J. St. 2013. *Kinect Hacks*. California: O'Reilly Media, Inc.
- [5] Webb, J., & Ashley, J. 2012. *Beginning Kinect Programming with the Microsoft Kinect SDK*. New York: Paul Manning.
- [6] Adiwijaya, Prima. A. K, dan Amalia, F. 2011. *Analisis dan Implementasi Random Forest dan Classification and Regression Tree (CART) untuk Klasifikasi pada Misuse Intrusion Detection System*. Bandung: Universitas Telkom.
- [7] Loh, W. Y. 2008. *Classification and Regression Tree Method*. Madison: University of Wisconsin. Eds. 315-323.
- [8] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. 2003. Detecting Moving Objects, Ghosts and Shadows in Video Streams, *IEEE Trans. on Patt. Anal. And Machine Intell.*, vol. 25, no. 10, Oct. 2003, pp. 1337-1342.
- [9] Yasin, H., Bjorn, K., & Andreas, W. 2014. *Motion Tracking, Retrieval and 3D Reconstruction from Video*. Bonn: University of Bonn.
- [10] Rakun, E, Adriani, M, Wipayoga, I.W, dkk. 2013. Combining Depth Image and Skeleton Data from Kinect for Recognizing Words in the Sign System for Indonesian Language (SIBI [Sistem Isyarat Bahasa Indonesia]). *ICACISIS*. Jakarta: Universitas Indonesia.
- [11] Breiman, L. 2001. *Random Forest*. California: University of California, Berkeley.