Rancang Bangun Modul Pengenalan Bahasa Isyarat Menggunakan Teknologi Kinect dan Metode Back Propagation Genetic Algorithm Neural Network

Yohanes Aditya Sutanto 1), Wijayanti Nurul Khotimah, S.Kom. 2), M.Sc., dan Dr.Eng. Nanik Suciati, S.Kom., M.Kom 3)  
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)  
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia  
*e-mail*: [yohanes.aditya94@gmail.com](mailto:yohanes.aditya94@gmail.com) 1) , [wijay.cs@gmail.com](mailto:wijay.cs@gmail.com) 2), [naniksuciati@gmail.com](mailto:naniksuciati@gmail.com) 3)

*Abstrak*— Bahasa isyarat adalah hal yang penting dalam komunikasi bagi orang yang menderita gangguan pendengaran. Kecepatan menguasai bahasa dan kemampuan mereka berinteraksi sangat dibutuhkan. Mereka membutuhkan bahan pembelajaran yang tidak hanya berisi tentang komponen aural saja, namun juga secara visual karena lebih nyata. Di sisi lain, teknologi berkembang pesat di segala aspek kehidupan. Berbagai macam terobosan teknologi baru telah diciptakan oleh manusia, salah satunya perangkat Kinect yang diciptakan untuk windows sekitar tahun 2012. Kinect dapat digunakan untuk mendeteksi gerakan isyarat yang diberikan dengan memanfaatkan fitur yang ada. Tugas akhir ini menggunakan fitur skeleton tracking yang ada pada Kinect untuk mendeteksi bahasa isyarat. Hasil pengujian dari tugas akhir ini menunjukkan bahwa metode Back Propagation Genetic Algorithm Neural Network yang digunakan sebagai classifier gerakan isyarat memiliki akurasi yang baik yaitu sekitar 92,5 persen. Hasil tersebut masih dapat ditingkatkan dengan menambahkan data training yang diambil dari sample yang bervariasi.

*Kata Kunci*— Kinect, Bahasa Isyarat, SIBI.

# PENDAHULUAN

B

ahasa isyarat adalah media bagi pada penderita Tunarungu untuk berkomunikasi dengan sekitarnya. Gerakan visual tubuh sangat membantu penderita agar yang ingin disampaikannya lebih mudah dimengerti oleh pasangan komunikasinya. Jika penderita berkomunikasi dengan gerakan bibir, tingkat keakuratan untuk mengartikan gerakan bibir tersebut lebih rendah dibandingkan dengan gerakan tubuh. Hal tersebut menjadi alasan untuk mengembangkan bahasa isyarat di Indonesia [1].

Di Indonesia ada dua dasar bahasa isyarat yang digunakan, salah satunya adalah Sistem Isyarat Bahasa Indonesia (SIBI). SIBI sudah menjadi bahasa isyarat Indonesia yang resmi. Di dalamnya terdapat posisi jari dan gerakan tangan untuk menggantikan kosa kata Bahasa Indonesia. Gerakan isyarat yang ada di dalam SIBI sudah diatur secara sistematis.

Di jaman sekarang, teknologi sudah berkembang sangat pesat. Sudah banyak hardware yang bisa membaca isyarat (*gesture*) yang ada dengan algoritma tertentu. Salah satunya adalah Kinect. Kinect adalah *motion sensing input device* yang dibuat oleh Microsoft untuk *game console* Xbox 360 dan Xbox One serta untuk windows PC. Dengan bentuk seperti webcam, Kinect dapat membuat pengguna mengontrol dan berinteraksi dengan *console* atau komputernya tanpa perlu *controller*, melainkan melalui *natural user interface* dengan menggunakan *gesture* dan perintah suara) [2].

Sebelumnya sudah ada tugas akhir yang dibuat oleh Risal Andika Tridisaputra tentang pengenalan bahasa isyarat Indonesia menggunakan teknologi leap motion dengan metode *Back Propagation Genetic Algorithm Neural Network*, tetapi tugas akhir tersebut hanya sebatas membaca isyarat abjad [3]. Oleh karena itu, dengan kemampuan yang dimiliki Kinect, munculah ide untuk membuat software pengenalan bahasa isyarat menggunakan Kinect. Bahasa isyarat yang digunakan akan berpaku pada Sistem Isyarat Bahasa Indonesia (SIBI).

# TINJAUAN PUSTAKA

## Kinect

Kinect adalah sebuah perangkat keras buatan Microsoft yang menggantikan input game controller pada Xbox dengan *natural user interface* berupa *gesture* dan perintah suara. *RGB camera, 3d depth sensing system, multi-array microphone*, dan *motorized tilt* merupakan komponen dasar dari Kinect [4].

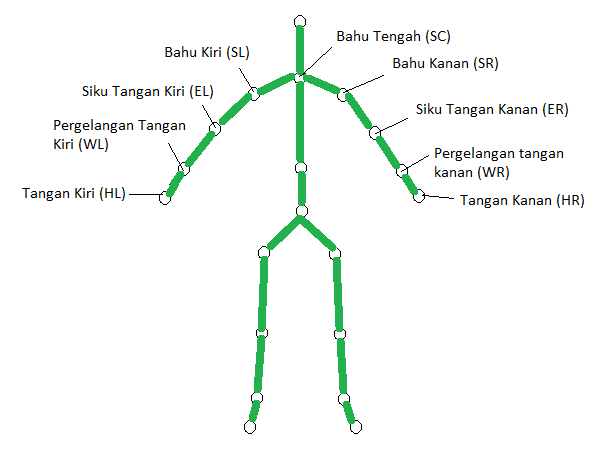
## Kinect SDK

Kinect SDK adalah *library* yang dibuat oleh Microsoft untuk pengembangan aplikasi perangkat lunak yang menggunakan Kinect sebagai alat input utama. Kinect SDK dapat diimplementasikan dengan bahasa pemrograman C#, C++, dan JavaScript. *Library* ini memiliki beberapa fitur diantaranya *skeleton tracking, Thumb Tracking, end of hand tracking, open/close hand gesture* dan lainnya [2]. Pada Kinect SDK yang akan digunakan memiliki 20 *skeleton joints* yang dapat dilihat pada Gambar1.



Gambar 1 *Skeleton Joints* yang Diketahui Kinect

Dari semua skeleton joints yang diketahui, hanya ada 9 yang akan dipakai dalam tugas akhir ini. Hal ini dikarenakan kebanyakan bahasa isyarat memiliki perbedaan satu sama lainnya dengan melihat 9 skeleton joints tersebut. Sembilan skeleton joints tersebut dapat dilihat pada Gambar 2.



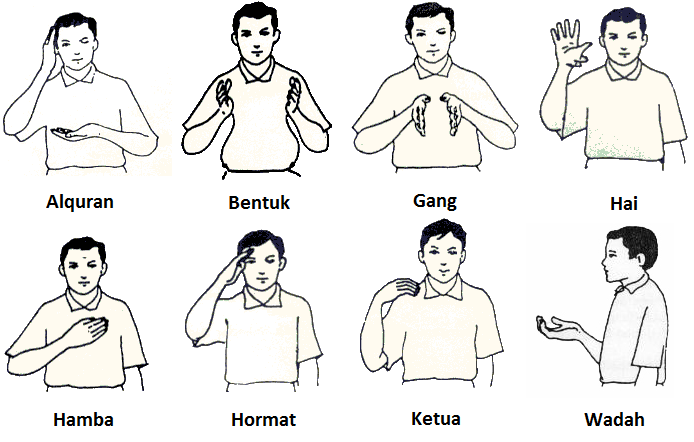
Gambar 2 *Skeleton Joints* yang Digunakan

## Bahasa Isyarat

Bahasa isyarat adalah sarana komunikasi bagi penderita tunarungu. Bahasa Isyarat berkembang dan memiliki karakteristik sendiri di berbagai negara. Di Indonesia, bahasa isyarat yang digunakan berdasarkan pada Sistem Isyarat Bahasa Indonesia (SIBI) [1]. Ada 4 jenis bahasa isyarat dalam SIBI yaitu:

1. Isyarat Pokok: melambangkan sebuah kata atau konsep
2. Isyarat Tambahan: melambangkan awalan, akhiran, dan partikel (Imbuhan)
3. Isyarat Bentukan: Dibentuk dengan menggabungkan isyarat pokok dan isyarat tambahan.
4. Abjad Jari: dibentuk dengan jari-jari untuk mengeja huruf.

Pada tugas akhir ini ada 8 bahasa isyarat pokok yang digunakan. Semua isyarat tersebut bersifat statis (tidak ada gerakan) karena fitur *skeleton tracking* yang dimiliki Kinect v1 sangat terbatas. Selain itu keterbatasan bahasa isyarat statis yang ada membuat ke 8 bahasa isyarat tersebut dipilih. Gambar 3 menunjukkan bahasa isyarat yang dipakai dalam tugas akhir ini.

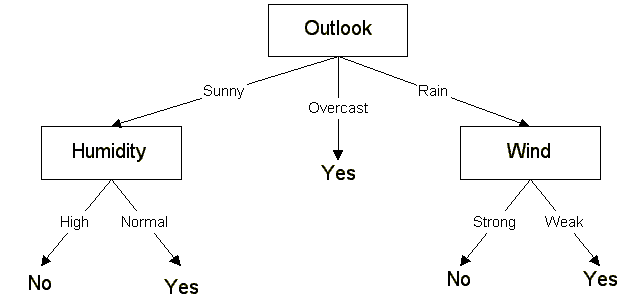


Gambar 3 Bahasa Isyarat yang Dipakai

## Decision Tree

*Decision tree* merupakan salah satu teknik *data mining* yang terkenal. *Decision tree* secara sistematis menganalisis informasi yang terkandung dalam *data source* yang besar untuk mengekstrak *valuable rules* dan *relationships* dan biasanya digunakan untuk *classification.* Dibandingkan dengan teknik *data mining* lainnya, *Decision tree* secara luas diterapkan dalam berbagai area karena cukup kokoh untuk *data scales* dan *distributions.*

Algoritma *decision tree* secara rekursif membagi *dataset* menggunakan pendekatan *depth-first greedy* atau pendekatan *breadth-first*, sampai semua item dimiliki oleh *class* yang diketahui. Struktur dari *decision tree* terbuat dari *root, internal,* dan *leaf nodes.* Kebanyakan *decision tree* *classifiers* melakukan *classification* dalam 2 tahap yaitu *tree-growing* dan *tree-pruning*. *Tree-growing* dilakukan dengan cara *top-down*.Dalam tahap ini *tree* secara rekursif dibagi sampai semua data termasuk dalam *class* yang sama. Dalam tahap *tree-pruning tree* yang sudah dibentuk dipotong. Hal ini dilakukan untuk meningkatkan prediksi dan akurasi dari *decision tree* dengan meminimalisasi *over-fitting* (*nouse* atau banyaknya data dalam *dataset training*) [5]. Contoh *decision tree* dapat dilihat pada Gambar 4.



Gambar 4 Contoh *Decision Tree*

Pada tugas akhir ini dilakukan pembagian *rule* menggunakan metode *decision tree* dengan mengambil level 1 *tree* dari hasil *decision tree* yang didapat, kemudian setiap *leaf* akan menjadi 1 golongan. Hal ini dilakukan untuk menyederhanakan model *classifier* yang akan dibentuk.

## Back Propagation

*Back Propagation* (BP) adalah algoritma *training* yang menggunakan *forward network* atau biasanya disebut *Multi Layer Perceptron* (MLP). Algoritma BP mencari nilai minimum dari *error function* dalam *weight space* dengan menggunakan *gradient descent*. Kombinasi dari berat yang meminimalisir *error function* dianggap sebagai sebuah solusi dari *learning problem.* Pada tugas akhir ini, data yang di *traning* oleh algoritma Back Propagation adalah data *skeleton joints* yang diambil oleh Kinect. Langkah algoritma BP yaitu:

1. Mencari *error* di *node* ke-j pada lapisan *output* menggunakan rumus pada Persamaan 1. *Target* adalah kelas yang diinginkan.

|  |  |
| --- | --- |
|  | (1) |

1. Update bobot pada lapisan *hidden* menggunakan rumus pada Persamaan 2. *Wij* adalah bobot di antara lapisan i dan j (antara lapisan *hidden* dan *output*) dan *α* adalah *learning rate*.

|  |  |
| --- | --- |
|  | (2) |

1. Mencari *error* di *node* ke-j pada lapisan *hidden* menggunakan rumus pada Persamaan 3. *Node* A terhubung ke *node* B dan *node* C. Error dari *node* B dan *node* C dibutuhkan untuk menghasilkan *error* pada *node* A.

|  |  |
| --- | --- |
|  | (3) |

1. Update bobot pada lapisan *input* menggunakan rumus pada Persamaan 2. *Wij* adalah bobot di antara lapisan i dan j (antara lapisan *input* dan *hidden*) dan *α* adalah *learning rate* [6].

## Back Propagation Genetic Algorithm Neural Network

Metode *Back Propagation* *Genetic* *Algorithm* *Neural* *Network* (BPGANN) merupakan metode gabungan dari *Back Propagation* dan *Genetic Algorithm* (GA) dengan menggunakan metode GA untuk mencari bobot inisial dari suatu *neural network* dan mempercepat konvergensi. Hal ini digunakan karena algoritma BP membuat proses konvergensi menjadi lambat [7]. Langkah-langkah BPGANN yang dilakukan dalam tugas akhir ini adalah seperti berikut:

1. Inisialisasi kromosom.
2. Mencari *fitness value* setiap kromosom dengan algoritma *Back Propagation* dan data *training* yang ada.
3. Melakukan proses *selection, cross-over,* dan *mutation.*
4. Memberikan kromosom terbaik untuk tahap *training* selanjutnya.

# ANALISIS DAN PERANCANGAN

## Deskripsi Umum Perangkat Lunak

Tugas akhir yang akan dikembangkan adalah sebuah modul pengenalan bahasa iysarat dengan menggunakan Kinect. Aplikasi ini menggunakan Kinect SDK dan dijalankan dengan perangkat keras Kinect V1.

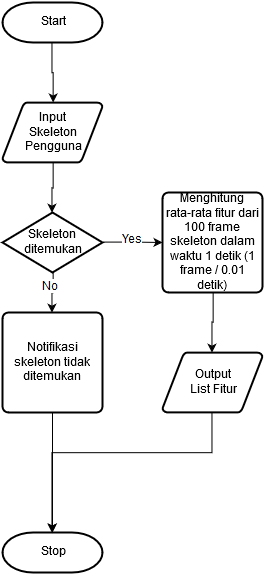
Pengguna utama adalah semua orang yang ingin belajar bahasa isyarat. Pengguna dapat mempelajari isyarat yang sudah ada dalam aplikasi ataupun memberikan isyarat baru sesuai dengan SIBI. Jika isyarat yang diberikan tidak ada dalam aplikasi maka tidak ada output yang diberikan.

## Rancangan Proses Aplikasi

Pada rancangan proses aplikasi akan dijelaskan mengenai proses yang terjadi dalam sistem untuk memenuhi fungsionalitas yang ada pada aplikasi. Proses ini penting agar aplikasi dapat berjalan secara baik dan benar.

1. Rancangan Proses Ekstraksi Fitur

Proses ekstraksi fitur skeleton sangat dibutuhkan bagi pengguna untuk melakukan training maupun testing. Saat pembuatan dataset baru, aplikasi akan menunggu 5 detik untuk mendeteksi skeleton pengguna, kemudian akan ada 2 detik tambahan untuk melakukan gerakan isyarat. Kemudian aplikasi akan mengambil total 100 data dalam waktu 1 detik yang akan dihitung dan dicari rata–ratanya kemudian disimpan dalam file .txt. Rancangan proses pengambilan fitur skeleton dapat dilihat pada Gambar 6.



Gambar 6 Diagram Alir Proses Ekstraksi Fitur

Untuk melakukan ekstraksi *skeleton* pengguna dalam melakukan *training* maupun *testing*, penulis mengambil sebanyak 9 *skeleton joints* yang diketahui oleh Kinect yang dapat dilihat pada Gambar 2. Kemudian *skeleton joints* tersebut diolah untuk dijadikan fitur yang berjumlah 28 buah [8]. Dua puluh depalan fitur yang dimaksud dapat dilihat pada Tabel 1.

Tabel 1.

Fitur yang Digunakan

|  |  |  |
| --- | --- | --- |
| Vector3 (x,y,z) | Angle (float) | Distance (float) |
| ER -> SR | ∠ SC - SR – ER | HR - HL |
| WR -> ER | ∠ SR - ER - WR |
| WR -> HR | ∠ ER - WR - HR |
| EL -> SL | ∠ SC - SL – EL |
| WL -> EL | ∠ SL - EL – WL |
| WL -> HL | ∠ EL - WL – HL |
| HL -> HR |  |

1. Rancangan Proses Normalisasi Fitur

Normalisasi fitur dilakukan setelah mendapatkan semua fitur yang ada di Tabel 1 dengan melakukan proses *feature* *scalling* untuk membuat fitur dalam *range* tertentu sehingga fitur lebih proporsional. Nilai maksimum dan minimum setiap fitur didapatkan dari hasil pengamatan yang dilakukan oleh penulis. Diagram alir proses normalisasi fitur dapat dilihat pada Gambar 7.



Gambar 7 Diagram Alir Proses Normalisasi Fitur

Untuk fitur *vector3* memiliki nilai minimum -2 dan nilai maksimum 2 untuk setiap koordinat. Dengan nilai tersebut persaamaan untuk normalisasi fitur *vector3* dapat dilihat pada Persamaan 4.

|  |  |
| --- | --- |
|  | (4) |

Untuk fitur *angle* memiliki nilai minimum 0 dan nilai maksimum π. Dengan nilai tersebut persamaan untuk normalisasi fitur *angle* dapat dilihat pada Persamaan 5.

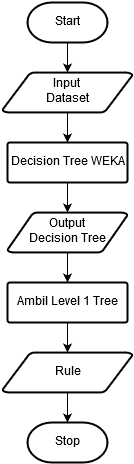
|  |  |
| --- | --- |
|  | (5) |

Untuk fitur *distance* memiliki nilai minimum 0 dan nilai maksimum 3.4641. Nilai maksimum didapatkan dari perhitungan *vector3* *u* dan *vector3* *v* yang memiliki jarak terjauh. Dengan nilai tersebut persamaan untuk normalisasi fitur *distance* dapat dilihat pada Persamaan 6.

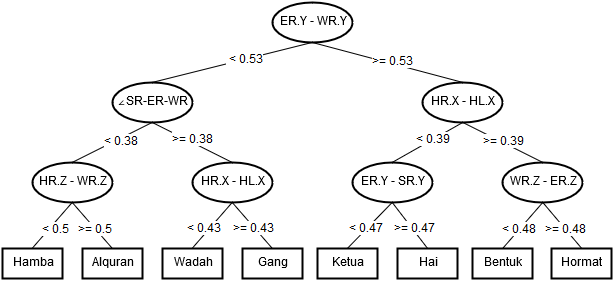
|  |  |
| --- | --- |
|  | (6) |

1. Rancangan Proses Pembuatan *Rule*

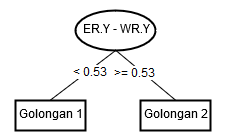
Pembuatan *rule* dilakukan untuk menyederhanakan model BPGANN. Hal ini dilakukan untuk meningkatkan akurasi model yang dibuat karena semakin sedikit target *class* yang diinginkan maka akurasi dari model tersebut akan semakin baik.Diagram alir proses pembuatan *rule* dapat dilihat pada Gambar 8. Tahap pertama pembuatan *rule* yaitu menggunakan metode *Decision Tree* yang ada pada aplikasi *WEKA* dengan data *training* yang sudah ditentukan. Hasil dari metode Decision Tree dapat dilihat pada Gambar 9. Dari hasil tersebut diambil level 1 *tree*. Hasil dari level 1 *tree* dapat dilihat pada Gambar 10. Selanjutnya, dari level 1 *tree* dibuat rule yang dapat dilihat pada Tabel 2.



Gambar 8 Diagram Alir Proses Pembuatan *Rule*



Gambar 9 Hasil Metode *Decision Tree*



Gambar 10 Level 1 *Decision Tree*

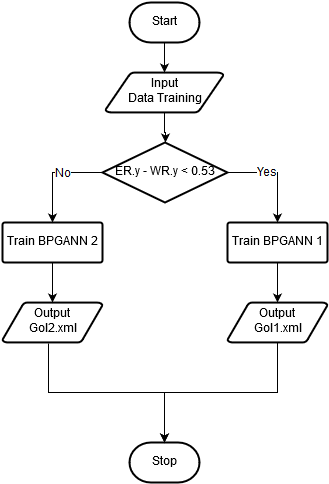
Tabel 2

*Rule* Pembagian Model BPGANN

|  |  |  |  |
| --- | --- | --- | --- |
| *Rule* | *If* | *Then* | *Class* |
| 1 | ER.Y – WR.Y < 0.53f | Gol1.xml | Alquran, Gang, Hamba, Wadah |
| 2 | *else* | Gol2.xml | Bentuk, Hai, Hormat, Ketua |

1. Rancangan Proses *Training*

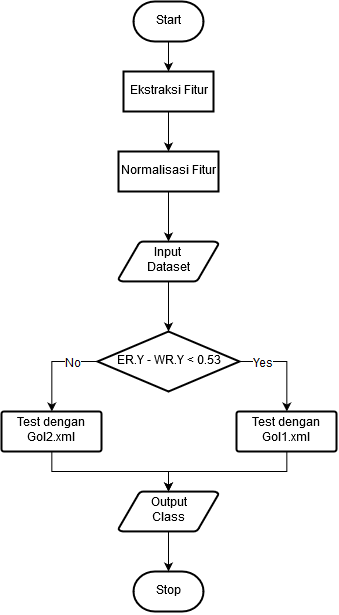
Proses *training* diawali inisialisasi model BPGANN*.* inisialisasi model BPGANNdiawali membagi data *training* yang sudah di normalisasi menjadi 2 kelompok sesuai dengan *rule* pada Tabel 3.7. Kemudian *training* dilanjutkan dengan mencari kromosom terbaik menggunakan *Genetic Algorithm*. Setelah itu dilakukan seleksi terhadap data *training* dengan kromosom yang didapat dan dilakukan proses inisialisasi model. Setelah model diinisialisasi *traning* dilanjutkan dengan input data *training* yang ada, setelah itu dilakukan pemeriksaan *rule*. Untuk data *training* yang memenuhi rule pertama, akan dilatih dengan BPGANN pertama dan menghasilkan output model BPGANN “Gol1.xml”. Untuk data *training* yang tidak memenuhi rule pertama, akan dilatih dengan BPGANN kedua dan menghasilkan output model *BPGANN* “Gol2.xml”. Diagram alir proses *training* dapat dilihat pada Gambar 11.



Gambar 11 Diagram Alir Proses *Training*

1. Rancangan Proses *Testing*

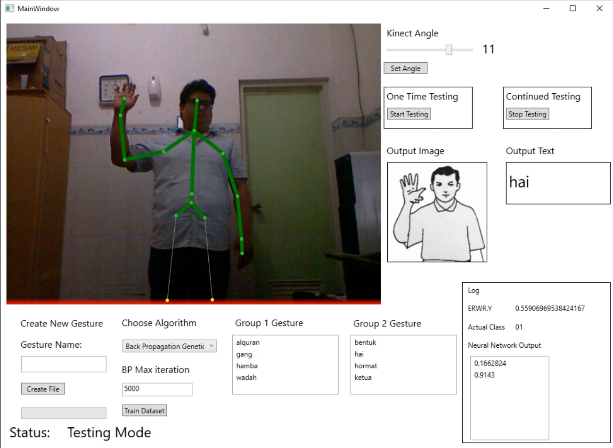
Proses *testing* dilakukan dengan membuat *dataset* baru yang dibentuk dari proses ektraksi fitur, setelah itu dilakukan normalisasi fitur. Setelah *dataset* di normalisasi, *dataset* masuk dalam pemeriksaan *rule*. Jika *dataset* memenuhi *rule* pertama maka *dataset* tersebut akan diuji menggunakan model BPGANN “Gol1.xml”. Jika *dataset* tidak memenuhi *rule* pertama maka *dataset* tersebut akan diuji menggunakan model BPGANN *“*Gol2.xml”. Diagram alir proses *testing* dapat dilihat pada Gambar 12.



Gambar 12 Diagram Alir Proses *Testing*

## Implementasi Aplikasi

Aplikasi ini dibangun dengan bahasa C# dengan menggunakan IDE Microsoft Visual Studio 2015 yang telah dilengkapi dengan Kinect SDK agar aplikasi dapat menggunakan fitur-fitur Kinect. Terdapat beberapa fungsi utama pada aplikasi ini yaitu pembuatan *dataset* bahasa isyarat, *training dataset,* dan *testing dataset.* Implementasi antarmuka aplikasi ini dapat dilihat pada Gambar 13.

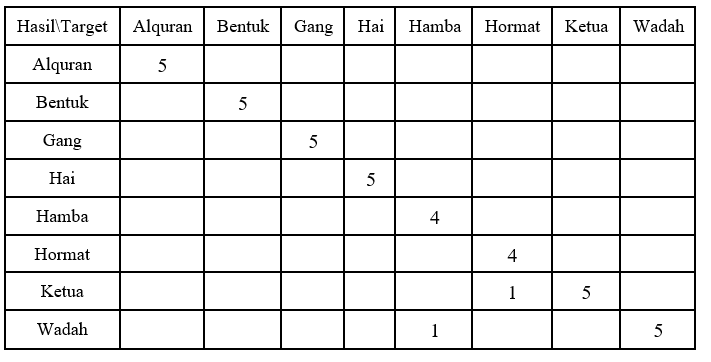


Gambar 13 Implementasi Antarmuka Aplikasi

# PENGUJIAN DAN EVALUASI

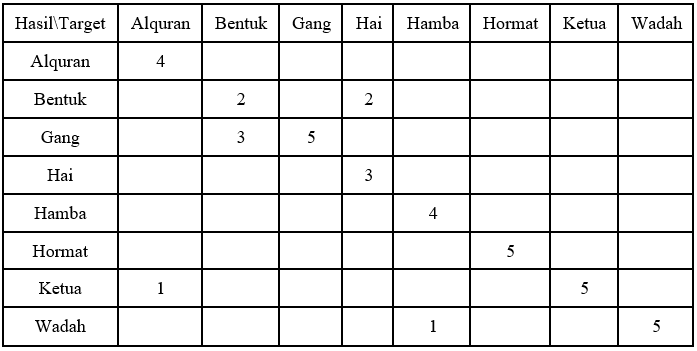
Skenario pengujian yang dilakukan dibagi menjadi 2 skenario A dan skenario B. Kemudian setiap skenario dilakukan sebanyak 2 kali oleh penulis dan orang lain. Pada skenario A, model BPGANN yang digunakan pada saat *testing* adalah hasil *training* dengan 160 data *skeleton* penulis. Sedangkan skenario B, model BPGANN yang digunakan pada saat *testing* adalah hasil *training* dengan 160 data *skeleton* penulis dan 80 data *skeleton* orang lain. Hasil pengujian skenario diatas dapat dilihat pada Tabel 3 sampai Tabel 6.

Tabel 3  
Hasil Pengujian Skenario A1



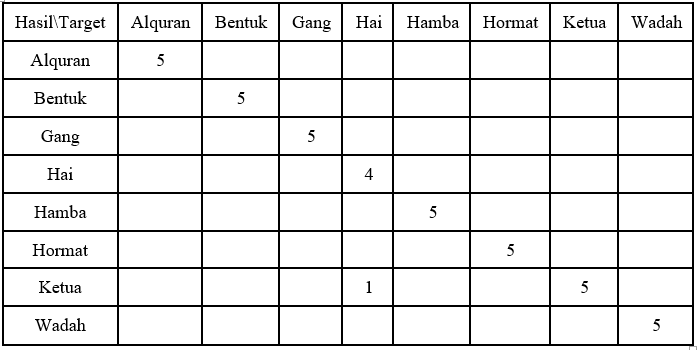
Tabel 4

Hasil Pengujian Skenario A2



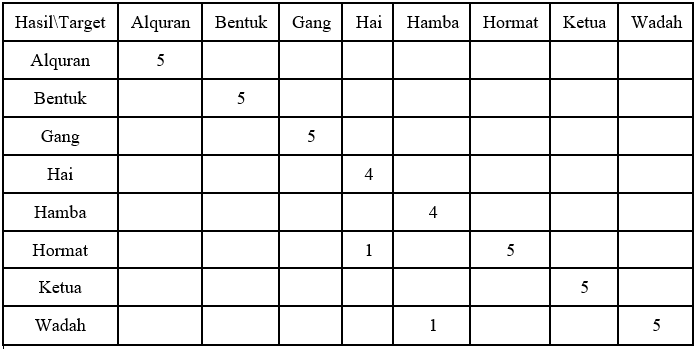
Tabel 5

Hasil Pengujian Skenario B1



Tabel 6

Hasil Pengujian Skenario B2



Hasil pengujian disusun dalam bentuk grafik yang dapat dilihat pada Gambar 14. Dari data yang terdapat pada grafik tersebut, diketahui bahwa aplikasi yang dibuat telah memiliki akurasi rata-rata yang cukup baik yaitu 92.5%.

Gambar 14 Hasil Pengujian Tingkat Akurasi Model BPGANN

# KESIMPULAN/RINGKASAN

Dari proses pengerjaan selama perancangan, implementasi, dan proses pengujian aplikasi yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Fitur yang digunakan untuk mendeteksi bahasa isyarat yang didapat dengan Kinect ada 15 fitur yaitu 7 fitur *vector3*, 6 fitur *angle,* dan 1 fitur *distance.*

2. Jumlah input neuron ditentukan berdasarkan jumlah fitur yang sebelumnya telah diseleksi dengan BPGA.

3. Aplikasi yang dibangun pada tugas akhir ini dapat menerjemahkan bahasa isyarat pokok dengan akurasi rata-rata 92.5%.

4. Perbedaan yang kecil pada gerakan isyarat yang dibentuk dapat mengakibatkan perbedaan output.

5. Semakin bervariasinya data training yang digunakan akurasi model BPGANN semakin meningkat.

UCAPAN TERIMA KASIH

Penulis Y.A.S. mengucapkan terima kasih Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember, Direktorat Pendidikan Tinggi, Departemen Pendidikan dan Kebudayaan Republik Indonesia yang telah mendukung dan memfasilitasi penelitian ini.

DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | E. Rakun, M. Andriarni, I. W. Wiprayoga, K. Danniswara and A. Tjandra, Combining Depth Image and Skeleton Data from Kinect for Recognizing Words in the Sign System for Indonesian Language (SIBI), IEEE, 2013. |
| [2] | “Kinect,” Wikipedia, 12 Desember 2015. [Online]. Available: https://en.wikipedia.org/wiki/Kinect. |
| [3] | R. A. T, "Rancang Bangun Modul Pengenalan Bahasa Isyarat Menggunakan Teknologi Leap Motion dan Metode Back Popagation - Genetic Algorithm Neural Network (BPGANN)," Institut Teknologi Sepuluh Nopember, Surabaya, 2015. |
| [4] | Sílvia Grasiella Moreira Almeida, Frederico Gadelha Guimarães, Jaime Arturo Ramírez, “Feature extraction in Brazilian Sign Language Recognition based on phonological structure and using RGB-D sensors,” *Expert Systems with Applications 41,* p. 7259, 2014. |
| [5] | Venkatadri.M, Lokanatha C. Reddy, “A Comparative Study on Decision Tree Classification Algorithms in Data Mining,” *International Journal of Computer Applications in Engineering, Technology and Sciences (IJ-CA-ETS),* vol. 2, no. 2, p. 24, 2010. |
| [6] | C. MacLeod, “The Synthesis of Artificial Neural Network Using Single String Evolutionary Techniques,” Robert Gordon University, Aberdeen, 1999. |
| [7] | S. Wang, S. Yin and M. Jiang, "Hybrid Neural Network Based On GA," in *Fourth International Conference on Natural Computation*, 2008. |
| [8] | Chao Sun, Tianzhu Zhang, Bing-Kun Bao, Changsheng Xu, “Discriminative Exemplar Coding for Sign Language,” *IEEE TRANSACTIONS ON CYBERNETICS,* vol. 43, p. 1418, 2013. |