



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metode penilitian yang digunakan dalam implementasi algoritma Principal Component Analysis dan K-Nearest Neighbor untuk pengenalan bahasa isyarat tangan terdiri dari beberapa tahap, yaitu studi literatur, analisis kebutuhan sistem, perancangan sistem, pengumpulan data dan pembuatan sistem.

a. Studi Literatur

Pada tahap ini akan dilakukan studi/pembelajaran terhadap berbagai teori yang mendukung penilitian ini melalui berbagai literatur. Teori yang dipelajari seperti citra digital, pengenalan citra, Principal Component Algorithm, K-Nearest Neighbor, bahasa isyarat, dan *confusion matrix*.

b. Analisis Kebutuhan Sistem

Pada tahap ini akan dilakukan perancangan & analisis kebutuhan sistem seperti menentukan spesikasi *software* maupun *hardware* yang digunakan, datadata yang dibutuhkan pada penelitian dan lain-lain.

c. Perancangan Sistem

Proses tahap ini, dilakukan perancangan terhadap sistem, seperti mendesain *user interface*, dan membuat *Flowchart* dari aplikasi.

UNIVERSITAS MULTIMEDIA NUSANTARA

17

d. Pengumpulan Data

Penelitian ini akan mengumpulkan data latih dan data uji berupa citra foto dari gambar tangan yang merepresentasikan huruf abjad dalam bahasa isyarat SIBI. Jumlah data yang akan dikumpulkan adalah sebanyak 45 buah citra untuk setiap huruf pada data latih dan 30 buah citra untuk setiap huruf dan kondisi pada data uji yang diambil menggunakan *webcam*.



(i) Akuisisi citra

> Akuisisi citra merupakan tahapan awal untuk mendapatkan citra tangan. Pengambilan citra pada penelitian ini menggunakan webcam pada perangkat komputer/laptop. Dan proses segmentasi untuk mengambil bagian tangan sebagai tanda bahasa isyarat akan digunakan library Haar Cascade Classifier dari OpenCV.

(ii) Preprocessing

> Preprocessing adalah proses pengolahan citra latih maupun uji sebelum memasuki tahapan ekstraksi fitur. Pengolahan yang akan dilakukan adalah *gravscale* (mengubah citra menjadi citra keabuan) dan resize (mengubah ukuran citra sesuai resolusi citra yang diinginkan) serta dilakukan histogram equalization (penyamarataan histogram).

(iii) Ekstraksi Fitur dengan PCA

> Tahap ini akan dilakukan pengambilan fitur/ciri berupa fitur grayscale pada citra yang telah diinput dengan memanfaatkan metode PCA yang akan digunakan sebagai parameter dalam proses *training* maupun *testing* nantinya.

Klasifikasi dengan K-NN (iv)

Dalam tahap ini akan dilakukan klasifikasi dengan K-Nearest Neighbor berdasarkan parameter fitur/ciri yang dihasilkan pada tahap sebelumnya. Identifikasi (v)

Dalam tahap ini akan ditampilkan output dari hasil klasifikasi.

f. Pengujian & Evaluasi

Pada tahap ini sistem akan dilakukan uji coba pada aplikasi yang dibuat disertai hasil evaluasi. Proses pengujian akan dilakukan dengan metode *black box* untuk menguji implementaasi dari algoritma PCA dan K-NN. Kemudian, hasil dari pengujian akan dianalisis dengan mengukur tingkat akurasi, *precision* dan *recall* menggunakan metode *confusion matrix* sebagai hasil evaluasi dari sistem.

3.2 Analisa Sistem

Dalam sistem ini, akan dibuat pengenalan bahasa isyarat tangan abjad statis yang mengacu pada SIBI. Abjad J dan Z tidak akan dikenali karena kedua abjad tersebut merupakan bahasa isyarat yang bergerak. Sistem ini akan dibuat dalam bentuk aplikasi *desktop*. Sistem ini juga akan menggunakan *storage* lokal untuk menyimpan data fitur citra latih dalam bentuk *textfile* dan menyimpan citra latih dalam *folder*.

Input bentuk tangan sebagai media pembentuk bahasa isyarat akan diambil melalui kamera webcam. Sistem ini akan secara otomatis mendeteksi area lokasi posisi tangan untuk menentukan Region of Interest (ROI) dengan menggunakan library Haar Cascase dari OpenCV. Input citra yang diterima oleh sistem kemudian akan dilakukan proses preprocessing citra. Citra yang telah dilakukan proses preprocessing akan diambil fiturnya menggunakan algoritma Principal Component Analysis (PCA). Kemudian setelah proses ektraksi fitur dilakukan, maka sistem akan melakukan klasifikasi dengan membandingkan fitur citra tersebut dengan fitur unik dari abjad-abjad pada data *training*. Proses klasifikasi dalam sistem ini menggunakan algoritma K-Nearest Neighbor.

Sistem ini bekerja pada latar belakang polos agar gambar yang ditangkap dapat lebih fokus pada bentuk tangan pengguna. Kamera *webcam* juga haruslah berada pada posisi sejajar dengan *input* tangan agar dapat menangkap bentuk tangan dengan baik.

3.3 Flowchart

Flowchart akan digunakan untuk menjelaskan urutan proses yang terdapat dalam sistem yang akan dibangun. *Flowchart* dari sistem ini terdiri dari *Flowchart* utama, proses pelatihan, *preprocessing*, *grayscaling* citra, ekstraksi fitur dengan PCA, *flowchart* mencari *mean centered*, *flowchart* mencari matriks *zero mean*, *flowchart* mencari matriks kovarian, *flowchart* mengurutkan nilai *eigenvalue*, *flowchart* menyesuaikan *eigenvector* berdasarkan *eigenvalue*, *flowchart* membentuk dataset baru, *flowchart* pengenalan bahasa isyarat, dan *flowchart* klasifikasi dengan K-NN.



A. Flowchart Utama

Gambar 3.2 adalah *Flowchart* Utama dari sistem yang akan dibuat. Pertama, sistem akan menampilkan halaman utama yang berisi navigasi dari menu pelatihan, pengujian dan tentang. Jika pengguna memilih menu tambah dataset maka sistem akan langsung menuju modul proses pelatihan.

Pengguna juga dapat memilih menu belajar SIBI yang dimana akan mengarahkan pengguna ke halaman menu belajar SIBI. Menu tersebut akan berisi informasi tentang SIBI dan kamus SIBI yang bisa dipelajari pengguna. Kemudian pengguna memilih menu pengujian maka akan langsung masuk pada proses pengujian. Selain itu, pengguna juga dapat memilih menu tentang, yang akan menampilkan halaman menu tentang. Jika pengguna menekan tombol kembali ke halaman utama maka sistem akan mengarahkan pengguna ke tampilan utama.

B. Flowchart Proses Pelatihan



Gambar 3.3 adalah *Flowchart* Proses Pelatihan. Pertama di modul ini akan menampilkan halaman pelatihan. Kemudian sistem akan mengecek apakah pengguna memilih *radio button* yang digunakan. Apabila radio button yang dipilih adalah tambah data huruf maka akan diminta input berupa jumlah data per huruf sedangkan jika tambah data baru akan diminta input berupa label dari data baru Jika semua *input* yang diminta telah dimasukkan, maka sistem akan melakukan deklarasi dan inisialisasi nilai terhadap variabel *i*, *j* dan *k*.

Variabel *i* merupakan variabel bantu yang digunakan untuk mengetahui seberapa banyak citra yang telah di-*input* dan variabel ini diberi nilai 0 pada awal proses pelatihan. Variabel *k* merupakan variabel bantu yang digunakan mengetahui urutan abjad yang akan di-*input* oleh pengguna (abjad a urutan 1, abjad b urutan 2, dan seterusnya) dan variabel ini diberi nilai *integer* 1 pada awal proses pelatihan. Variabel *j* digunakan sebagai variabel bantu untuk mengetahui seberapa banyak citra pada abjad *k* yang telah di-*input* dan variabel ini diberi nilai *integer* 0 pada awal proses pelatihan. Varibel *m* merupakan sebagai jumlah iterasi yang dibutuhkan untuk melakukan proses *input* citra pelatihan.

Kemudian apabila pengguna menekan tombol *start webcam* maka sistem akan membuka *kamera webcam* yang tersedia. Bila pengguna memilih deteksi tangan otomatis maka akan digunakan *library* dari openCV sistem akan mendeteksi letak tangan untuk menentukan ROI. Apabila posisi ROI akan statis. Setelah itu, apabila pengguna menekan *start capture* maka sistem akan mengambil gambar dari kamera *webcam* untuk diambil sebagai *input* citra latih. *Input* citra tersebut akan dilakukan tahap *preprocessing* terlebih dahulu, yang dimana diakhir proses ini akan dihasilkan matriks citra. *Preprocessing* ini sendiri akan menggunakan bantuan dari library openCV yang terdiri dari *resizing* citra menjadi berukuran 250 x 250 piksel, kemudian dilakukan *grayscalling*, dan terakhir dilakukan penyetaraan histrogram (*Histogram Equalization*). Matriks citra kemudian akan diubah ke dalam bentuk vektor 1 baris. Vektor citra tersebut kemudian dimasukkan ke dalam matriks kumpulan dari data pelatihan. Hasil dari matriks kumpulan vektor citra data latih ini nantinya akan menjadi input untuk proses ektraksi fitur dari data latih dengan PCA. Proses input citra data latih ini akan terus-menerus dilakukan sampai jumlah data citra latih yang dibutuhkan terpenuhi.

F. Flowchart Ekstraksi Fitur dengan PCA

Gambar 3.6 adalah *Flowchart* Ekstraksi Fitur dengan PCA. Apabila modul pemanggil modul merupakan modul proses pelatihan, maka langkah awal yang dilakukan adalah mencari *mean centered*. *Mean centered* merupakan rata-rata dari nilai parameter dalam *dataset* latih yang dihitung menggunakan Persamaan 2.1. Proses *mean centered* ini ditunjukkan pada Gambar 3.7.





Gambar 3.4 Flowchart Ekstraksi Fitur dengan PCA Gambar 3.4 adalah *Flowchart* Ekstraksi Fitur dengan PCA. Apabila modul pemanggil modul merupakan modul proses pelatihan, maka langkah awal yang dilakukan adalah mencari *mean centered*. *Mean centered* merupakan rata-rata dari 26 nilai parameter dalam *dataset* latih yang dihitung menggunakan Persamaan 2.1. Proses *mean centered* ini ditunjukkan pada Gambar 3.5.



Mean centered kemudian akan disimpan ke sebuah *file txt* yang akan digunakan untuk proses pengujian. *Mean centered* ini berbentuk matriks $(n \ x \ l)$ dimana *n* merupakan jumlah parameter. Namun, karena adanya perbedaan dimensi

dari data sampel ($n \ x \ m$) dan *mean centered* ($n \ x \ m$), maka perlu disamakan dimensinya dengan menggandakan rata-rata seluruh citra sebanyak m, sehingga rata-rata seluruh sampel juga memiliki dimensi yang sama.

Setelah digandakan, maka proses perhitungan *zero mean* dapat dilakukan. Perhitungan tersebut dihitung dengan Persamaan 2.2 yang dijelaskan prosesnya pada Gambar 3.6.



Apabila modul *zero mean* dipanggil oleh modul pelatihan, modul ini akan menerima matriks rata-rata atau *mean centered* yang di dapat dari proses pelatihan sebelumnya yang telah digandakan. Kemudian dicari zero mean dari matriks pelatihan dengan melakukan pengurangan antara matriks kumpulan vektor citra 28 latih dan matriks *mean centered*. Namun apabila modul pemanggil berasal dari modul pelatihan maka akan dilakukan perhitungan dengan cara melakukan pengurangan antara vektor citra uji dengan *mean centered* yang diambil dari penyimpanan *file txt*. Proses selanjutnya apabila modul pemanggil adalah modul pelatihan, maka akan dicari matriks kovarian dengan menggunakan Persamaan 2.4 yang dijelaskan prosesnya pada Gambar 3.7.

Dari matriks kovarian ini akan dicari nilai *eigenvalue* serta *eigenvector*. Proses pencarian nilai eigen ini akan menggunakan bantuan *library* dari JAMA. Setelah didapatkan nilai *eigenvalue*, nilai tersebut kemudian akan diurutkan secara *descending* seperti pada Gambar 3.8.



Gambar 3.7 Flowchart Mencari Matriks Kovarian



Gambar 3.8 Flowchart Mengurutkan Nilai *Eigenvalue* secara *descending* Proses tersebut juga akan menyimpan index dari *eigenvalue* yang telah diurutkan. Index ini yang kemudian digunakan untuk mengurutkan *eigenvector* sesuaikan dengan *eigenvalue* yang terlihat proses pada Gambar 3.9.

Kemudian *eigenvector* atau *principal component* yang telah diurutkan akan dikoversikan atau dipetakan ke dimensi aslinya dengan Persamaan dengan menggunakan Persamaan 2.7 seperti yang terlihat pada Gambar 3.10. *Principal*

component yang telah dikonversikan akan disimpan di penyimpanan, untuk nantinya digunakan pada proses ekstraksi fitur pada vektor citra uji.





Setelah itu, akan dilakukan pencarian matriks bobot. Matriks ini dicari dengan persamaan 2.8. Prosesnya adalah mengalikan matriks *principal component* yang sudah di-*transpose* dengan matriks citra latih apabila dalam proses pelatihan ataupun vektor citra uji apabila dalam proses pengujian. Hasil matriks bobot akan disimpan pada penyimpanan, yang akan digunakan nantinya untuk proses klasifikasi. Proses tersebut secara lengkap terdapat pada Gambar 3.11





G. Flowchart Proses Pengenalan Bahasa Isyarat



Gambar 3.12 adalah *Flowchart* Pengenalan Bahasa Isyarat. Proses pengenalan bahasa isyarat pertama akan membuka *webcam* yang dipilih oleh pengguna. Kemudian setelah pengguna telah memasukkan citra *input* untuk proses uji, citra tersebut akan dilakukan proses *preprocessing* terlebih dahulu. *Preprocessing* nantinya akan menghasilkan vektor baris dari citra uji.

Setelah itu, vektor baris tersebut akan dilakukan ekstraksi fitur dengan PCA, agar dimensinya sama dengan dimensi vektor citra latih. Setelah dimensinya sama, citra uji tersebut kemudian akan diklasifikasikan dengan menggunakan algoritma K-NN. Hasil klasifikasi akan ditampilkan ke layar pengguna.

F. Flowchart Klasifikasi dengan K-NN

Gambar 3.13 adalah Flowchart Klasifikasi dengan K-NN. Proses inisialisasi variabel-variabel yang diperlukan menjadi proses pertama dalam klasifikasi dengan K-NN. Variabel huruf merupakan array list yang digunakan untuk menampung kelas dari bahasa isyarat. Variabel ini akan diisi dengan label kelas dari bahasa isyarat dari huruf a sampai y (tanpa j dan z) diikuti dengan kata bahasa isyarat lainnya apabila ditambahkan oleh pengguna secara berurutan, sehingga index a adalah 0, b adalah 1, dan seterusnya. Selanjutnya akan dihitung masing-masing jarak/distance menggunakan Persamaan 2.9. Hasil dari perhitungan tersebut dimasukkan ke variabel distanceFeature. Kemudian nilai dari distanceFeature tersebut akan dimasukkan ke sebuah Hashmap dengan variabel distance beserta key berupa string huruf dari kelas huruf yang bersesuaian dengan citra latih yang sedang diukur jarak nya dengan citra uji. Proses ini akan dilakukan sejumlah data citra latih yang ada. Nilai distanceFeature tersebut dikali dengan -1 untuk keperluan sorting. Untuk melakukan sorting, Hashmap dari distance diubah terlebih dahulu menjadi sebuah array. Sorting akan menggunakan fungsi bawaan dari Java yaitu Array.sort().





Gambar 3.14 Flowchart Klasifikasi dengan K-NN (Lanjutan)

Kemudian setelah berhasil di *sorting*, array tersebut dikembalikan ke *Hashmap* dengan variabel *distanceSorted*. Dari *distanceSorted* tersebut akan diambil sejumlah k (diinput user sebelumnya) data teratas. Dari data sejumlah k tersebut kemudian akan dicari kelas huruf yang memiliki frekuensi yang terbesar. Kelas huruf dengan frekuensi terbesar akan menjadi hasil klasifikasi yang akan ditampilkan ke pengguna. Apabila terdapat lebih dari satu kelas huruf yang memiliki frekuensi terbesar maka akan diambil satu data lagi dari *distanceSorted* untuk dilakukan pencarian frekuensi terbesar kembali.

3.4 Desain AntarMuka

Bagian desain antarmuka aplikasi akan menjelaskan perancangan dari sistem pengenalan bahasa isyarat SIBI yang diimplementasikan dengan algoritma PCA dan K-NN. Tampilan antarmuka pada sistem ini terbagi menjadi 4 bagian bedasarkan menu yang ada di sistem yaitu menu utama, menu pelatihan, menu pengujian, dan menu tentang.

A. Menu utama

Gambar 3.15 adalah rancangan antarmuka halaman menu utama. Menu utama akan menjadi menu yang pertama kali ditampilkan oleh sistem ke pengguna. Menu ini berisi tiga tombol navigasi ke 4 menu lain yaitu menu Tambah Dataset, Belajar SIBI, Latihan dan Tentang Aplikasi.



Gambar 3.15 Rancangan Antarmuka Halaman Menu Utama

B. Menu Tambah Dataset

Gambar 3.16 adalah rancangan antarmuka halaman menu tamabah dataset. Dalam menu ini terdapat sebuah *dropdown* untuk memilih *webcam* yang ingin digunakan untuk mengambil citra tangan pelatihan. Selain itu terdapat tombol Start untuk mengaktifkan *webcam*, tombol Stop untuk me-nonaktifkan *webcam* serta tombol *Capture* untuk mengambil citra. *Preview* di webcam akan ditampilkan di atas tombol-tombol tersebut.



Gambar 3.16 Rancangan Antarmuka Halaman Menu Tambah Dataset

Kemudian apabila tombol Capture telah ditekan oleh pengguna maka hasil citra yang diambil akan ditampilkan disisi kanan antarmuka dari halaman menu ini. Dibawahnya terdapat data konfigurasi pelatihan yang harus diisi oleh pengguna dimana apabila pengguna ingin menambahkan data pelatihan huruf maka pengguna harus menekan tombol *radio button* beserta mengisi data jumlah data huruf yang ingin dimasukkan. Kemudian apabila pengguna ingin menambah dataset baru berupa kosakata baru maka pengguna dapat menambahkannya dengan memilih *radio button* tambah data baru kemudian mengisi label dari data tersebut.

C. Menu Belajar SIBI



Gambar 3.17 Rancangan Antarmuka Halaman Menu Belajar SIBI

Gambar 3.17 adalah rancangan antarmuka halaman menu belajar SIBI. Pada menu akan ditampilkan informasi mengenai SIBI dan sekaligus terdapat kamus dari huruf yang dapat dipelajari sebelum pengguna bisa latihan di menu latihan.

D. Menu Latihan

Gambar 3.18 adalah rancangan antarmuka halaman menu latihan. Menu ini diperuntukan bagi pengguna untuk melatih apa yang telah dia pelajari pada halaman belajar SIBI. Disini pengguna bisa mengetes/memvalidasi benar atau tidaknya cara mereka untuk membentuk tangan bahasa isyarat. Di menu ini akan terdapat list dataset bahasa isyarat apa saja yang ada dalam *database* pada sistem ini. Ketika pengguna menekan *capture*, maka sistem akan menerjemahkan bahasa isyarat yang dibentuk oleh pengguna.



Gambar 3.18 Rancangan Antarmuka Halaman Menu Latihan

E. Menu Tentang



Gambar 3.19 Rancangan Antarmuka Halaman Menu Tentang

Gambar 3.19 adalah rancangan antarmuka halaman menu Tentang. Dalam menu ini terdapat informasi singkat tentang pengembang dari sistem. Dalam menu ini juga terdapat cara penggunaan sistem.