

BAB III

METODE PENELITIAN

3.1 Bahan dan Alat Penelitian

3.1.1 Bahan Penelitian

Bahan penelitian yang digunakan adalah berupa data-data getaran mesin yang diakuisisi dari sensor pada contoh mesin yang mengalami kerusakan dan mesin yang berada dalam kondisi normal. Data tersebut adalah:

- a. Data getaran mesin hasil percobaan diFakultas Teknik Mesin Universitas Diponegoro.
- b. Data getaran mesin yang diambil *Bearing Data Center* <http://csegroups.case.edu/bearingdatacenter/>
- c. Data getaran mesin yang diambil dari *data set bearing MPF* <http://data-acoustics.com/measurement/bearing-faults>

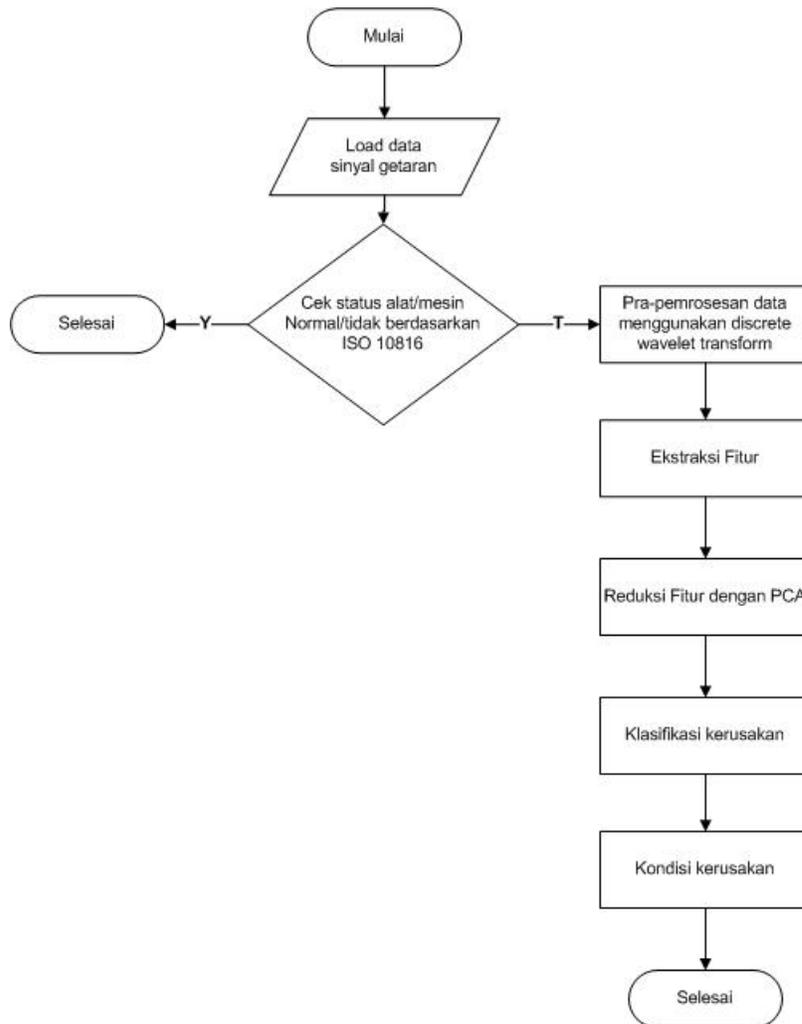
3.1.2 Alat Penelitian

Alat yang digunakan dalam pengujian sistem pada penelitian ini adalah sebagai berikut:

- a. Perangkat keras yang digunakan adalah Laptop Asus Prosesor Intel® Core™ i3-3217U CPU @ 1.80GHz 1.70GHz, RAM 2.00 GB, Hardisk 464 GB
- b. Perangkat lunak yang digunakan adalah Microsoft Windows 7 ultimate (64-bit) dan Matlab 7.10.0 R2014a (64-bit)

3.2 Prosedur Penelitian

Prosedur penelitan dari diagnosa kerusakan *bearing* menggunakan *Principal Component Analysis (PCA)* dan *Klasifikasi Naïve Bayes* diperlihatkan pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.2.1 Data Sinyal Getaran

Data sinyal getaran adalah data hasil pengukuran mesin yang diperoleh dalam 4 kondisi bantalan yaitu *ball fault bearing*, *innerace fault bearing*, *outterace fault bearing* dan *normal bearing*.

3.2.2 Pengecekan Severitas

Pengecekan severitas adalah proses awal untuk mengecek kondisi data getaran mesin berdasarkan nilai vrms-nya. Pengecekan ini dilakukan dengan cara membandingkan nilai vrms data dengan Tabel ISO 10816. Berdasarkan Tabel ISO tersebut akan terlihat apakah mesin masih dalam kondisi yang bagus atau tidak. Tabel ISO 10816 ditunjukkan pada Tabel 3.1.

Tabel 3.1 Tabel Standar ISO 10816

VIBRATION SEVERITY PER ISO 10816					
Machine		Class I small machines	Class II medium machines	Class III large rigid foundation	Class IV large soft foundation
in/s	mm/s				
Vibration Velocity Vrms	0.01	0.28			
	0.02	0.45			
	0.03	0.71		good	
	0.04	1.12			
	0.07	1.80			
	0.11	2.80		satisfactory	
	0.18	4.50			
	0.28	7.10		unsatisfactory	
	0.44	11.2			
	0.70	18.0			
	0.71	28.0		unacceptable	
1.10	45.0				

3.2.3 Pra-pemrosesan Data

Pra-pemrosesan data adalah proses membersihkan data, mengintegrasikan data, mentransformasikan data dan mereduksi data. Pra-pemrosesan data yang dilakukan pada penelitian ini terdiri dari proses transformasi wavelet diskret, ekstraksi fitur dan reduksi fitur.

3.2.3.1 Transformasi Wavelet Diskret

Data getaran mesin yang digunakan untuk penelitian akan dilakukan pra-pemrosesan data terlebih dahulu menggunakan transformasi *wavelet* diskret. Transformasi *wavelet* diskret adalah proses sampling sinyal berdasarkan penskalaan dan pergeseran parameter. Transformasi *wavelet* diskret akan mengubah sinyal sumber menjadi dua klasifikasi sinyal yaitu frekuensi tinggi dengan resolusi waktu yang tinggi serta frekuensi rendah dengan resolusi frekuensi yang tinggi. Proses ini diawali dengan proses filtering dengan melewati sinyal pada seperangkat *highpass* filter dan *lowpass* filter. Kemudian dilanjutkan dengan operasi *sub-sampling* dengan mengambil masing-masing setengah dari keluaran filter. Proses selanjutnya adalah modifikasi sinyal berdasarkan fungsi skala dan waktu, lalu proses rekonstruksi sinyal pada tiap level koefisiennya. Proses ini dinamakan proses dekomposisi dan dapat dilakukan secara berlanjut hingga didapatkan tingkat dekomposisi yang diinginkan.

3.2.4 Ekstraksi Fitur

Ekstraksi fitur adalah proses untuk menghitung fitur-fitur statistika yang diperlukan untuk mengelompokkan data-data getaran berdasarkan kemiripan atau kesamaan ciri yang berguna untuk proses identifikasi.

3.2.5 Reduksi Fitur

Reduksi fitur menggunakan *principal component analysis* (PCA) bertujuan untuk mereduksi fitur, sehingga semakin sedikit data, algoritma data mining akan semakin cepat, dan akurasi dari proses data mining menjadi lebih tinggi. Reduksi fitur menggunakan PCA ini akan memilih 5 fitur berdasarkan nilai *eigenvalue* terbesar dari matrik kovarians.

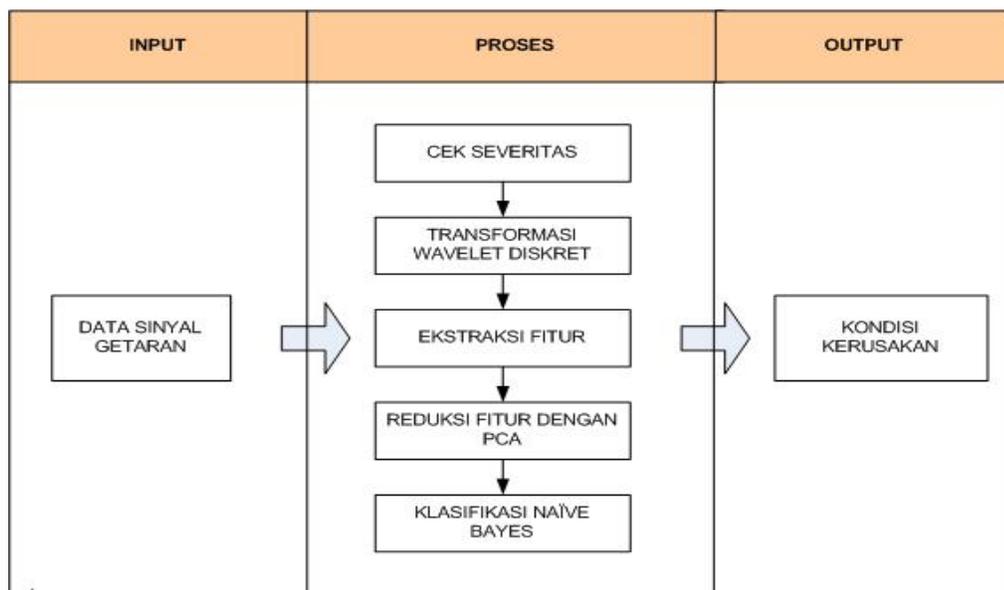
3.2.6 Klasifikasi Kerusakan

Data getaran mesin yang digunakan untuk penelitian akan dilakukan proses klasifikasi dengan menggunakan klasifikasi *Naïve Bayes*. Proses ini dilakukan dengan cara melakukan pengenalan pembelajaran sistem dengan

menggunakan data pelatihan. Kemudian dilakukan proses pengujian yang berfungsi untuk mengukur keakuratan dari sistem yang sudah dilatih dengan data pelatihan.

3.3 Kerangka Sistem Informasi

Kerangka sistem informasi diagnosa kerusakan *bearing* menggunakan *principal component analysis* (PCA) dan klasifikasi *Naïve Bayes* ditunjukkan pada Gambar 3.2.



Gambar 3.2 Kerangka sistem informasi

Penjelasan kerangka sistem informasi diagnosa kerusakan *bearing* menggunakan *principal component analysis* (PCA) dan klasifikasi *Naïve Bayes* adalah sebagai berikut:

1. Input atau Masukan

Pada tahap ini pengguna memasukkan data getaran mesin yang akan dilakukan klasifikasi.

2. Proses

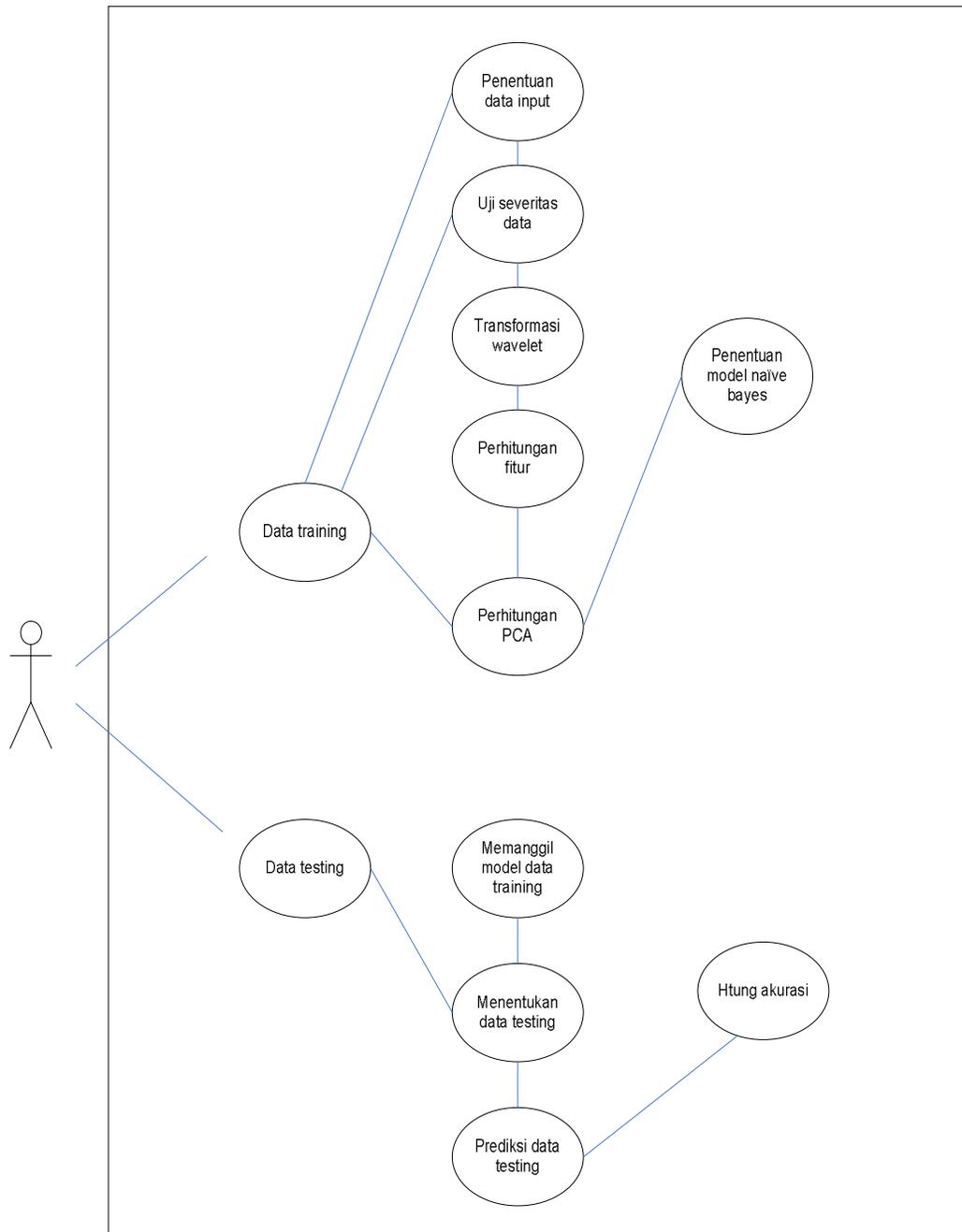
Data getaran mesin akan dicek tingkat severitasnya terlebih dahulu berdasarkan tabel standar ISO 10816 dengan cara menghitung nilai vrmsnya. Jika data dalam kondisi tidak normal, maka dilanjutkan dengan proses transformasi data menggunakan *wavelet* diskret, kemudian diekstraksi fitur. Fitur-fitur yang sudah dihitung tersebut kemudian direduksi menggunakan PCA. Selanjutnya data hasil PCA tersebut sebagian digunakan sebagai data pelatihan untuk pembelajaran sistem.

3. *Output* atau Keluaran

Setelah semua proses dilakukan, kemudian dilakukan pengujian data sehingga dapat ditampilkan kondisi kerusakan pada data uji tersebut.

3.4 Perancangan Sistem Diagnosa Kerusakan

Perancangan sistem diagnosis kerusakan *bearing* dibuat berdasarkan UML (*Unified Modelling Language*). Pada sistem ini digunakan dua macam diagram yaitu diagram *use case* untuk mengetahui *actor* dan *use case* serta hubungan diantara keduanya, dan *diagram activity* untuk mengetahui alur kerja dari sistem. Diagram *use case* ditunjukkan pada Gambar 3.3 berikut ini:

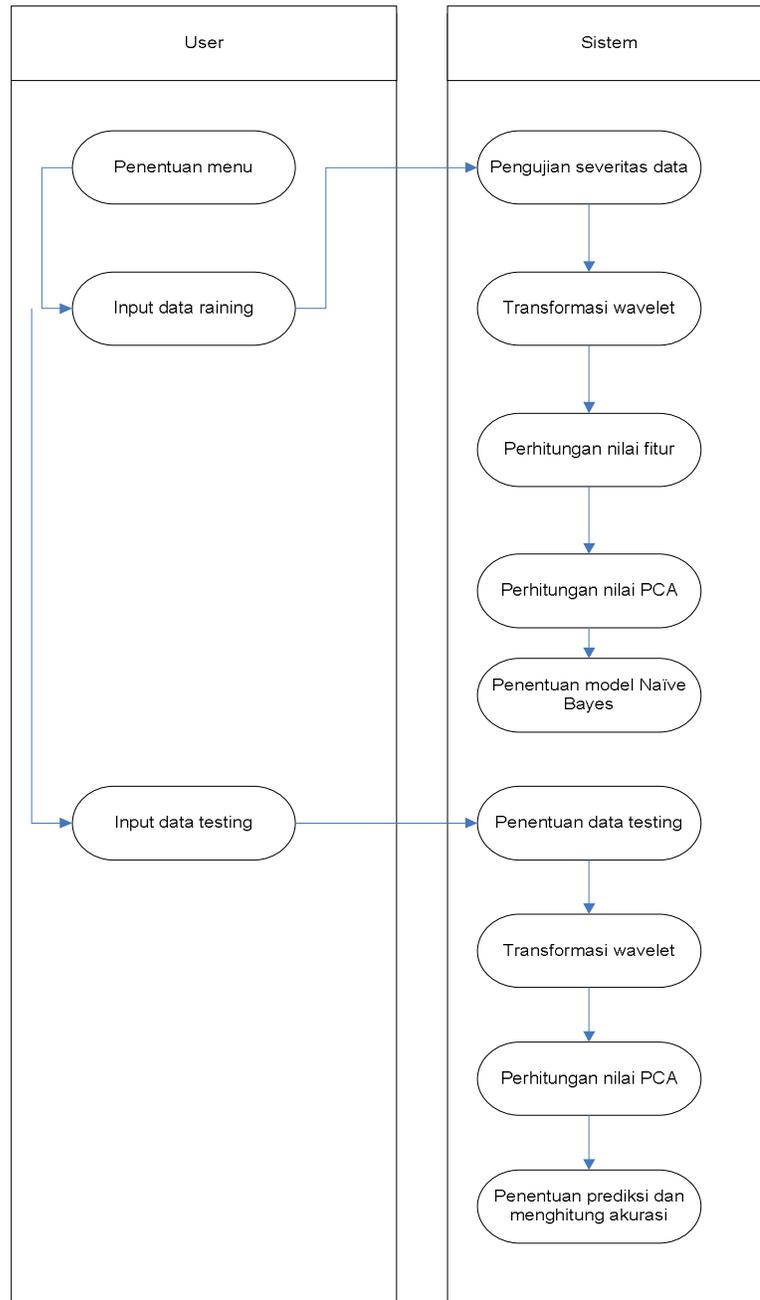


Gambar 3.3 Use Case Diagram Diagnosis Kerusakan Bearing

Tabel 3.2 Deskripsi *usecase* dengan aktor

Aktor	<i>Usecase</i>	<i>Use case extend</i>	Deskripsi <i>usecase</i>
User	Data training	Penentuan <i>data training</i>	<i>user</i> akan diminta untuk menentukan data yang akan di training
		Penentuan severitas data	<i>user</i> akan melihat kondisi data getaran <i>bearing</i> jika dikorelasikan dengan standar nilai <i>vrms</i> internasional
		Transformasi <i>wavelet</i>	<i>User</i> akan diminta untuk menentukan folder <i>output</i> hasil transformasi
		Penghitungan fitur	<i>User</i> akan melihat proses perhitungan fitur
		Perhitungan PCA	<i>User</i> akan melihat penghitungan PCA dimulai dari penghitungan nilai kovarian dan <i>eigen value</i> nya. <i>User</i> akan diminta inputan nilai koefisien PCA nya
		Penentuan model <i>Naïve Bayes</i>	<i>User</i> akan melihat hasil klasifikasi data dengan <i>Naïve Bayes</i> dan model <i>Naïve Bayes</i> akan tersimpan dalam sistem
	Data testing	Menentukan <i>data testing</i>	<i>user</i> akan menentukan data yang akan diuji
		Hitung akurasi	<i>User</i> akan melihat nilai prediksi dan akurasi data uji

Activitydiagram menggambarkan berbagai alur aktivitas dalam sistem yang dirancang, gambaran awal dari masing-masing aktivitas, proses dan keputusan yang terjadi dan gambaran akhir. Diagram *activity* ditunjukkan pada Gambar 3.4 berikut:



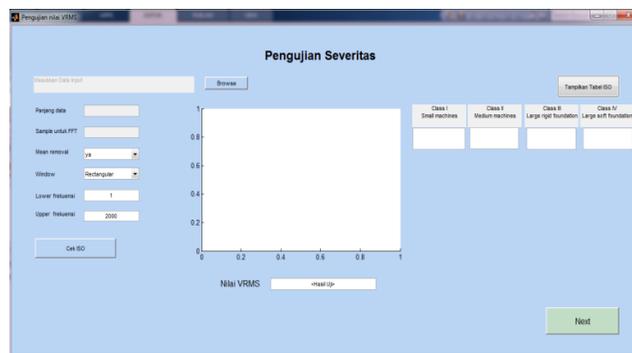
Gambar 3.4 Activity Diagram Diagnosis Kerusakan Bearing

3.5 Tampilan Antarmuka

Tampilan antarmuka ini bertujuan untuk memberikan bentuk-bentuk antarmuka yang dibutuhkan didalam sistem untuk memudahkan interaksi antara pengguna dan komputer.

3.5.1 Tampilan Antarmuka Pengujian Severitas

Tampilan antarmuka pengujian severitas merupakan tahapan untuk mengecek severitas atau kondisi keparahan data getaran mesin berdasarkan tabel ISO 10816. Tampilan antarmuka pengujian severitas ditunjukkan pada Gambar 3.5.



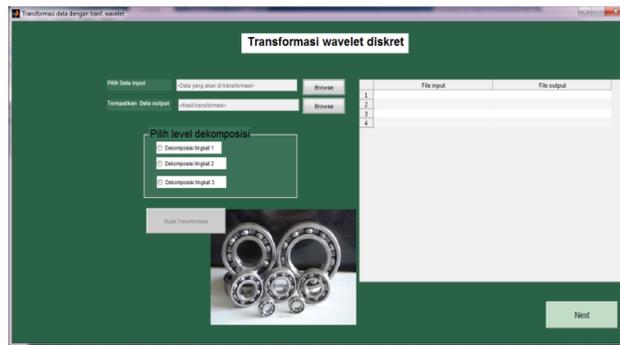
Gambar 3.5 Tampilan antarmuka pengujian severitas

Untuk melakukan proses maka pengguna harus memasukkan data sinyal getaran yang sudah disediakan oleh pemogram untuk dicek nilai severitasnya dengan menghitung nilai vrmsnya. Setelah data dipilih maka kemudian pengguna akan menekan tombol **Cek ISO** sehingga akan muncul nilai vrms dari sinyal tersebut. Untuk membandingkan dengan Tabel ISO 10816, pengguna dapat menekan tombol **Tampilkan Tabel ISO**.

3.5.2 Tampilan Antarmuka Transformasi *Wavelet* Diskret

Tampilan antarmuka transformasi *wavelet* diskret merupakan tahapan untuk melakukan pra-pemrosesan data menggunakan transformasi *wavelet* diskret. Tampilan antarmuka transformasi *wavelet* diskret ditunjukkan pada Gambar 3.6.

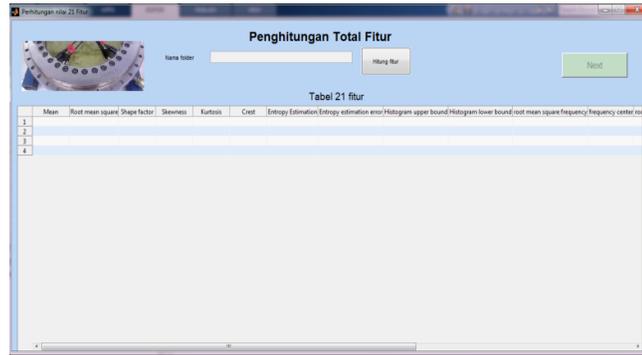
Pada tahap ini pengguna harus memasukkan data sinyal getaran yang akan dicek dengan mengisikan ke data input, serta memilih folder yang akan digunakan untuk menyimpan hasil output. Kemudian pengguna memilih level dekomposisi yang diinginkan dan menekan tombol **Mulai Transformasi** untuk memulai proses transformasi *wavelet*. Data yang sedang diproses akan ditampilkan dalam bentuk gambar gelombang sinyal dan tabel. Data hasil pengolahan dengan *wavelet* diskret akan disimpan dalam sebuah variabel lalu dikirim ke dalam *form* penghitungan total fitur.



Gambar 3.6 Tampilan antarmuka transformasi *wavelet* diskret

3.5.3 Tampilan Antarmuka Penghitungan Total Fitur

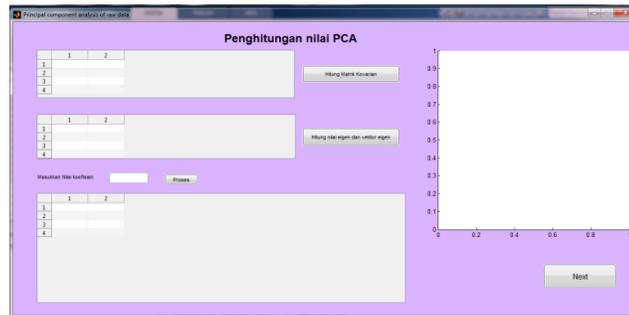
Pada tahap ini akan dilakukan proses ekstraksi fitur, yaitu penghitungan 21 fitur statistik. Tampilan antarmuka penghitungan total fitur ditunjukkan pada Gambar 3.7. Jika pengguna menekan tombol **Hitung Fitur** maka program akan menghitung 21 fitur statistik dari data dan akan menampilkan hasilnya dalam bentuk tabel. Data hasil penghitungan total fitur ini selanjutnya akan dikirim ke *form* penghitungan PCA seperti ditunjukkan pada Gambar 3.7.



Gambar 3.7Tampilan antarmuka penghitungan total fitur

3.5.4 Tampilan Antarmuka Penghitungan PCA

Pada tahap ini akan dilakukan reduksi fitur menggunakan PCA (*Principal Component Analysis*). Langkahnya yaitu dengan menghitung matrik kovarian dan menghitung nilai *eigen* dan vektor *eigen*. Tampilan antarmuka penghitungan PCA ditunjukkan pada Gambar 3.8.



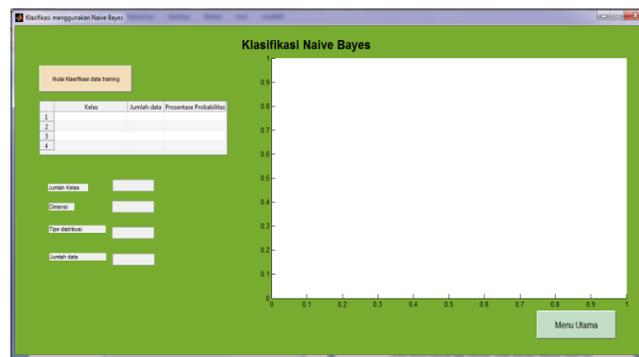
Gambar 3.8 Tampilan antarmuka penghitungan nilai PCA

Pada *form* penghitungan nilai PCA ini jika pengguna menekan tombol **Hitung Matrik Kovarian**, maka program akan menghitung nilai matrik kovarian dari data hasil pengolahan sebelumnya dan menampilkan hasilnya dalam bentuk tabel. Kemudian jika menekan tombol **Hitung nilai eigen dan vektor eigen**,

maka program akan menghitung nilai *eigen* dan vektor *eigen* yang kemudian akan ditampilkan hasilnya dalam bentuk tabel. Setelah itu pengguna harus memasukkan nilai koefisien PCA yang diinginkan dan menekan tombol **Proses** untuk memulai hasil penghitungan PCA. Hasil penghitungan akan ditampilkan dalam bentuk tabel.

3.5.5 Tampilan Antarmuka Klasifikasi *Naïve Bayes*

Pada tahap ini akan dilakukan klasifikasi *Naïve Bayes*. Tampilan antarmuka klasifikasi *Naïve Bayes* ditunjukkan pada Gambar 3.9.

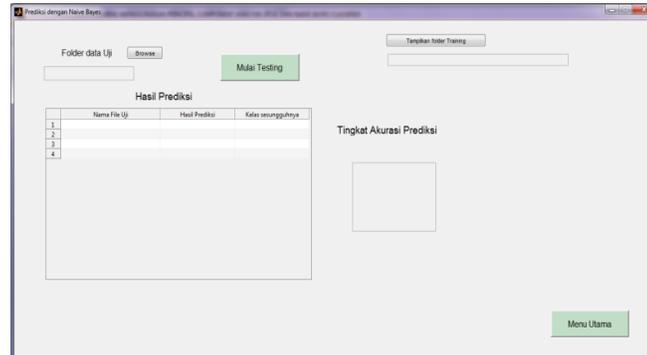


Gambar 3.9 Tampilan antarmuka klasifikasi *Naïve Bayes*

Pengguna harus menekan tombol **Mulai klasifikasi** agar program memulai klasifikasi *Naïve Bayes*. Setelah itu akan ditampilkan gambar hasil klasifikasi, kelas data yang diklasifikasikan, jumlah data yang digunakan, dan prosentase probabilitas dari data tersebut.

3.5.6 Tampilan Antarmuka Pengujian Data

Tampilan antarmuka pengujian data ditunjukkan pada Gambar 3.10.



Gambar 3.10 Tampilan antarmuka pengujian data

Pada tahap ini pengguna harus memasukkan data uji yang akan dilakukan pengujian dengan menekan tombol **browse** dan kemudian menekan tombol **Mulai Testing** untuk memulai pengujian. Program akan menampilkan hasil pengujian dalam bentuk tabel dan menghitung nilai akurasinya.