BAB 3
METODE PENELITIAN

3.1 PEMODELAN SISTEM

1. Data Set

Penelitian ini menggunakan data sekunder yang diperoleh dari situs http://physionet.org/content/challenge-2016/1.0.0/. Sumber tersebut dipilih karena merupakan salah satu sumber dataset terpercaya yang menjadi rujukan banyak penelitian khususnya penelitian dengan objek suara jantung. Rekaman suara jantung bersumber dari beberapa kontributor di seluruh dunia, dikumpulkan secara klinis maupun non-klinis, baik dari subjek sehat maupun pasien patologis. Pada penelitian ini, dipilih *dataset* e dengan data sebanyak 2141 *files* rekaman suara jantung manusia, dengan durasi dari 5 hingga 120 detik. Jumlah data yang berlabel normal sebanyak 1958 *files* suara, dan abnormal sebanyak 183 *files*, seluruh *files* rekaman disimpan dalam format .wav.

Rekaman suara jantung dikumpulkan dari berbagai lokasi di tubuh. Empat lokasi tersebut adalah daerah aorta, daerah pulmonal, daerah trikuspid dan daerah mitral. Pada *test* dan *training* set, rekaman suara jantung dibagi menjadi dua jenis: rekaman suara jantung normal dan abnormal. Rekaman normal berasal dari subyek sehat dan yang abnormal berasal dari pasien dengan diagnosis jantung yang telah terkonfirmasi. Kondisi abnormal ini meliputi pasien cacat katup jantung dan penyakit arteri koroner. Rekaman suara jantung juga mengandung berbagai *noise*, seperti suara percakapan, gerakan stetoskop, pernapasan, dan usus.

2. Perangkat Keras (*Hardware*)

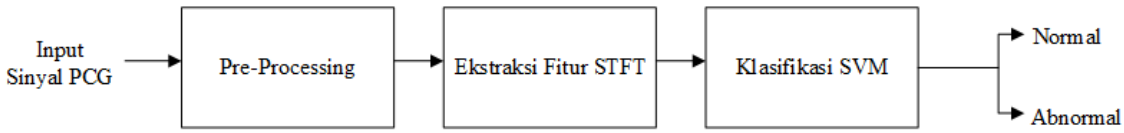
Perangkat keras yang digunakan pada penelitian ini yaitu sebuah *Personal Computer* (PC) dengan spesifikasi sebagai berikut:

1. Intel(R) Core(TM) i5-4200 CPU 1.60 Ghz up to 2.4 Ghz
2. *Window*s 10 (64-Bit)
3. RAM 4 GB
4. HDD 1 TB

3. Perangkat Lunak (*Software*)

Software yang digunakan dalam simulasi dan analisis sinyal suara jantung berdasarkan sinyal PCG adalah *Spyder*. *Software* ini menggunakan bahasa pemrograman phyton 3.7. *Spyder* digunakan untuk menampilkan hasil nilai sinyal PCG yang telah melalui proses ekstraksi fitur STFT (*Short time fourier transform*) dan proses klasifikasi *Support Vector Machine* (SVM).

4. Pemodelan Sistem



Gambar 3.1. Pemodelan Sistem

Pada perancangan sistem berdasarkan Gambar 3.1, sinyal PCG atau data uji akan melewati berbagai tahapan. Tahapan tersebut diantaranya tahap *pre-processing* yang dibagi 2 bagian yaitu *data labelling* dan *zero-padding*, kemudian tahapan ekstraksi fitur STFT, dan tahapan klasifikasi SVM untuk melihat hasil apakah termasuk jantung normal atau abnormal. Hasil dari klasifikasi nantinya akan dibandingkan dengan data asli untuk mendapatkan nilai akurasi, spesifitas, dan sensitifitas dengan validasi *Confusion Matrix*.

3.1.1. Input Sinyal PCG

Sinyal PCG yang akan digunakan adalah Sinyal PCG normal dan abnormal (mengandung murmur). Kode program yang digunakan adalah sebagai berikut:

```
def load_wave_file(path):
    wav, sr = librosa.load(path)
    return wav, sr
wav_path = ' '
```

Dimana:

def load_wave_file() = sebuah fungsi dengan nama load_wave_file() yang nantinya digunakan untuk menampilkan file rekaman suara jantung.

wav = membaca wav files.

sr = sampling rate dari wav files.

librosa.load() = perintah yang disediakan *librosa* untuk memuat sebuah audio files.

path = letak file masukan.

3.1.2. *Pre-processing*

Pre-processing yang dibagi 2 tahapan yaitu labeling data dan zero-padding. *Labeling* digunakan untuk mengkategorikan sinyal suara jantung normal dan abnormal berdasarkan *file reference* yang ada pada *dataset.* Sedangkan, *zero-padding* digunakan untuk menyamakan ukuran *array file* inputan sinyal suara jantung.

```
training_e = pd.read_csv('training-
e/REFERENCE.csv',names=["wavfile", "actuals", "pred"])
train = pd.concat([training_e], axis=0)
category_group = train[['wavfile', 'actuals',
'pred']].groupby(['actuals','pred']).count()
```

Dimana:

pd.read_csv = fungsi yang diberikan *panda* untuk membaca file csv. Dengan *file path* 'training-e' dan nama file csv 'REFERENCE.csv'.

names = daftar nama kolom yang akan digunakan, adalah wavfile, actuals, dan pred.

train = fungsi untuk data training.

pd.concat = menggabungkan objek di sepanjang sumbu tertentu dengan logika set opsional di sepanjang sumbu lainnya.

category_group = fungsi untuk mengkelompokkan kategori sinyal suara jantung sesuai label.

Perintah diatas nantinya digunakan untuk mengkategorikan files rekaman suara jantung dari dataset yang telah dipilih yaitu dataset-e, dan files rekaman suara tersebut dikelompokkan sesuai informasi label "*actuals*" yang ada pada *file* '*REFERENCE.csv*' kedalam sinyal suara normal (-1) dan abnormal (1). Karena *sampling rate* ditentukan sebesar 22050 Hz sedangkan *raw data* dari *file* rekamannya hanya memiliki 2000 Hz maka kita memerlukan *zero padding*. Konsep dari *zero padding* adalah menambahkan nilai 0 ke sebuah panjang *array data* untuk menyamakan durasi data secara keseluruhan.

```
def zero_pad_allfiles(files):
    pathfiles = list(map(lambda x:
get_wave_path(x,wav_path),files))
    wave_files = []
    for file in pathfiles:
        wav, sr = load_wave_file(file)
        #pad by audio length
        wav_pad = zero_pad(wav, audio_len)
        wave_files.append(wav_pad)
    return wave_files

def zero_pad(arr, length):
    l = len(arr)
    if l > length:
        result = arr[0:length]
    else:
        zero_pad = np.zeros(length - l)
        result = np.concatenate([arr, zero_pad])
    return result
```

3.1.3. *Short time fourier transform* (STFT)

Tahapan pertama dari ekstraksi fitur pada STFT adalah menentukan nilai parameter untuk *input* fungsi STFT, yaitu *window length*, *hop size*, dan *fft points*. yang didefinisikan sebagai:

```
FRAME_SIZE = 2048
HOP_SIZE = 512
n_fft = FRAME_SIZE
```

Dimana:

FRAME_SIZE = 2048; panjang *window*.

HOP_SIZE = 512; jumlah sample antara masing-masing *window*.

n_fft = panjang sinyal terhadap suatu *window*.

Dalam hal ini, *window length* ditentukan sebesar 2048. Nilai tersebut dipilih untuk mendapatkan resolusi waktu yang bagus. Resolusi waktu yang bagus adalah resolusi dengan jarak antar *window* yang lebar. Namun dengan lebarnya *window*, maka resolusi frekuensi pun menjadi tidak bagus. Oleh karena itu diperlukan nilai dari *hop size* yang lebih kecil untuk dapat mengoptimalkan *range* dari resolusi frekuensi. Dimana pada penelitian ini ditentukan *hop size* sebesar 512, dengan *range* frekuensi yang semakin kecil akan memberikan hasil yang semakin akurat.

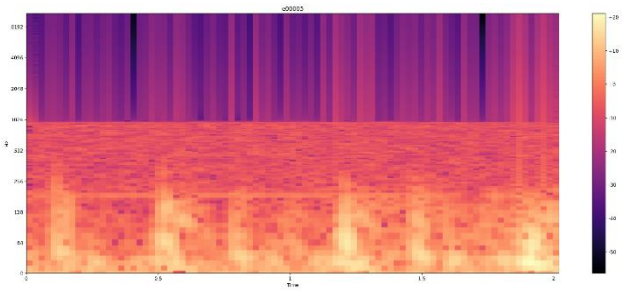
nfft pada penelitian ini ditentukan nilainya sama dengan nilai dari *window length*, hal ini dipilih sesuai dengan nilai *default* yang disediakan oleh *syntax* untuk kesesuaian dengan pemrosesan sinyal suara yang berasal dari suara fisik. Adapun jenis *window* yang digunakan adalah *Hanning*, *Hamming* dan *Blackman window*, masing-masing persamaan fungsi *window* tersebut dapat dilihat pada bab sebelumnya. Pada aplikasi *Spyder*, fungsi-fungsi *window* ini dapat ditulis dengan sintaks.

```
window='hann'
window='hamming'
window='blackman'
```

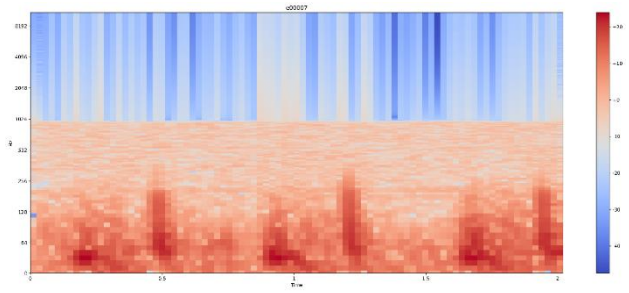
Kemudian setelah menentukan *window*, maka dapat dilakukan perhitungan STFT, dimana matriks STFT akan merepresentasikan waktu pada kolom (sumbu x), dan frekuensi pada baris (sumbu y) dalam sebuah spektrogram atau sebuah grafik yang memberikan informasi tentang perubahan gelombang dalam rentang waktu, frekuensi, dan intensitas amplitudo. Intensitas amplitudo pada suatu frekuensi dan pada suatu waktu (waktu, frekuensi) di dalam spektrogram dinyatakan dengan nilai warna tertentu (*grayscale* atau RGB). Spektrogram pada *python* pada umumnya dapat dibuat dengan sintaks:

```
def plot_spectrogram(Y, sr, hop_length,
y_axis="linear"):
        plt.figure(figsize=(25, 10))
        librosa.display.specshow(Y,
                                 sr=sr,

hop_length=hop_length,
                                 x_axis="time",
                                 y_axis=y_axis)
        plt.colorbar(format="%+2.f")
        plt.title(files[i])
```



Gambar 3.2. Contoh Spektrogram Kondisi Jantung Abormal



Gambar 3.3. Contoh Spektrogram pada Kondisi Jantung Normal

Gambar 3.2 dan 3.3. menampilkan bentuk spektrogram pada umumnya dimana sumbu x merepresentasikan waktu, sumbu y merepresentasikan frekuensi, dan intensitas amplitudo atau energi (umumnya dalam dB) dinyatakan dalam spektrum warna tertentu. Pada spektrogram, panjang *window* dapat mempengaruhi intensitas warna dan resolusi dari waktu dan frekuensi. Perbedaan mendasar dari kedua spektrogram antara kondisi normal dan abnormal adalah pada kerapatan bias warna, dimana kerapatan ini dipengaruhi oleh frekuensi dari masing-masing kondisi. Spektrogram kondisi abnormal akan memiliki kerapatan bias warna yang lebih rapat karena pada kondisi jatung abnormal memiliki rentang frekuensi yang lebih tinggi yaitu 20-400 Hz dibandingkan dengan kondisi jantung normal yang hanya memiliki rentang 20-200 Hz [32].

Setelah menghasilkan spektrogram, berikutnya akan diambil nilai STFT, nilai frekuensi dan waktu pada saat sinyal mencapai puncaknya, dan nilai rata-rata amplitudo dari sebuah sinyal suara untuk kemudian akan dilatih dan diuji menggunakan *Support Vector Machine* (SVM). Keempat nilai tersebut digunakan karena metode STFT menghasilkan suatu nilai waktu dan frekuensi, dan juga amplitudo.

Nilai STFT adalah suatu bilangan kompleks yang terdiri dari bilangan *real* dan *imaginer* yang merepresentasikan suatu nilai dalam domain waktu dan frekuensi. Persamaan dapat dilihat pada bab sebelumnya. Dalam syntax dituliskan sebagai berikut:

```
S_scale = librosa.stft(x_data[i], n_fft=FRAME_SIZE,
window='hamming', hop_length=HOP_SIZE)
```

Agar nilai kompleks tersebut dapat dihitung, maka nilai tersebut diabsolutkan dengan syntax dapat dituliskan sebagai berikut:

```
Y_scale = np.abs(S_scale)
```

Dan nilai amplitudo/magnitudo adalah nilai yang menunjukkan kekuatan sinyal dalam desibel (dB). Nilai awal dari amplitudo adalah berupa tegangan (volt), dan untuk mengubahnya kedalam bentuk desibel dalam *syntax python* dapat menggunakan fungsi yang telah disediakan oleh *librosa* seperti berikut:

```
Y_log_scale = librosa.power_to_db(Y_scale)
```

3.1.4. *Support Vector Machine* (SVM)

Metode klasifikasi yang digunakan adalah menggunakan metode *support vector machine* (SVM) dengan input 4 nilai ekstraksi fitur, yaitu nilai STFT, nilai frekuensi dan waktu saat amplitudo tertinggi, dan nilai rata-rata amplitudo pada sebuah sinyal suara. SVM mengolah data yang diperoleh dari ekstraksi fitur dengan menggunakan bantuan 3 jenis *kernel*, yaitu *linear*, rbf, dan *polynomial* dan variasi nilai C parameter 7 *level* dengan rentang 10^{-3}, 10^{-2},..., 10^{2}, 10^{3}. Hasil akhir SVM akan menampilkan rata-rata akurasi yang diperoleh setelah melalui tahapan *K-fold Cross Validation* dan *Confusion Matrix*.

Pada penelitian ini menggunakan metode *support vector machine* jenis *support vector* classification (SVC). Metode ini digunakan karena akan dilakukan klasifikasi sebanyak 2 kelas. SVC menerapkan pendekatan berbasis "*one-vs-one*" pada klasifikasi 2 kelas. Pengklasifikasian dibuat dengan masing-masing melatih data dari 2 kelas. Dalam syntax klasifikasi SVC ini dituliskan sebagi berikut.

```
svclassifier = SVC(kernel='rbf', gamma=0.1, C=1)
svclassifier.fit(X[train], Y[train])
y_pred = svclassifier.predict(X[test])
cv.append(accuracy_score(Y[test], y_pred))
```

Dimana:

svclassifier = *syntax* yang digunakan untuk menyimpan pengklasifikasian SVC.

SVC = memanggil fungsi perintah SVC.

svclassifier.fit = pengklasifikasi akan di cocokan ke suatu model, dan akan mempelajari model tersebut (dalam hal ini X *train* dan Y *train*)

svclassifier.predict = setelah svc mempelajari dari suatu model. .predict akan memprediksi nilai baru pada data *testing*, yaitu X test.

kernel = jenis *kernel* yang digunakan saat pengujian. *kernel*:{'*linear*', 'poly', 'rbf', 'sigmoid'}.

Pada penelitian ini digunakan *kernel linear*, rbf, dan *polynomial*. Masing-masing persamaan variasi *kernel* tersebut dapat dilihat pada bab sebelumnya.

3.1.5. Parameter Pengujian

Parameter pengujian penelitian ini menggunakan *K-fold Cross Validation* untuk digunakan dalam mengestimasi kinerja model, dan *Confusion Matrix* untuk digunakan sebagai penguji keakuratan yang sering digunakan dalam menunjukkan hasil klasifikasi terutama pada data multi*class*.

Cross validation merupakan metode yang dapat digunakan untuk data yang berjumlah terbatas [33]. Data yang digunakan akan dibagi secara *random* ke dalam k subset dengan ukuran yang sama. Dataset tersebut akan dibagi menjadi data *training* dan data *testing*. Proses *training* dan *testing* dilakukan sebanyak k kali secara berulang-ulang. Pada iterasi ke-i, data ke-i akan digunakan untuk data *testing* dan sisanya digunakan secara bersamaan untuk data *training*. Adapun langkah-langkah dalam prosedur sebagai berikut:

1. Membagi data menjadi k bagian dengan ukuran yang sama.
2. Pada data k-1 merupakan data *training* dan bagian lainnya merupakan data *testing.*
3. Pada langkah ini dilakukan berulang sebanyak k setiap kombinasi data *training* dan data *testing* yang berbeda, sehingga semua data akan menjadi data *testing.* Estimasi nilai dari suatu model akhir dapat diperoleh dengan menghitung rata-rata dari nilai akurasi pada setiap iterasi.

Confusion matrix atau yang dikenal dengan tabel kontingensi merupakan Uji keakuratan yang sering digunakan dalam menunjukkan hasil klasifikasi terutama pada data multi*class. Confusion matrix* membandingkan nilai yang diprediksi dengan nilai aktual dan menciptakan ukuran kesalahan klasifikasi [34]. Dalam *Confusion matrix,* semakin rendah jumlah kesalahan klasifikasi, maka semakin baik kinerjanya.

Tabel 3.1. ***Confusion Matrix***

Kelas Prediksi / Kelas Sebenarnya	Terprediksi Normal (-1)	Terprediksi Abnormal(1)
Kondisi Normal Sesungguhnya (-1)	True Positif (TP)	False Positif (FP)
Kondisi Abnormal Sesungguhnya (1)	False Negatif (FN)	True Negatif (TN)

Untuk menghitung satuan ukuran kinerja dari suatu model berdasarkan *confusion matrix* dengan *True Positive* (TP) adalah jumlah data pasien dengan kondisi jantung normal terklasifikasi dengan benar ke dalam pasien yang memiliki kondisi jantung normal, *True Negative* (TN) merupakan jumlah data pasien kondisi jantung abnormal terklasifikasi dengan benar ke dalam pasien dengan kondisi jantung abnormal, *False Negative* (FN) merupakan jumlah data pasien kondisi jantung abnormal terklasifikasi ke dalam kondisi jantung normal, dan *False Positive* (FP) merupakan jumlah data pasien yang normal terklasifikasi ke dalam kondisi abnormal. Persamaan pengukur kinerja klasifikator dapat dituliskan sebagai berikut [35]:

$$Sensitivitas = \frac{TP}{(TP + FN)} \quad \textbf{(3.1)}$$

$$Spesifisitas = \frac{TN}{(TN + FP)} \quad \textbf{(3.2)}$$

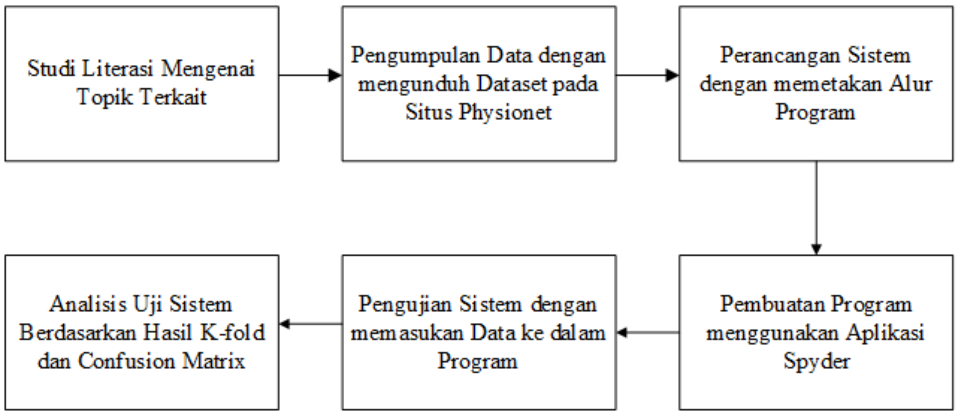
$$Akurasi \quad = \frac{TP + TN}{(TP + TN + FP + FN)} \quad \textbf{(3.3)}$$

$$AUC \quad = \frac{Sensitivitas + Spesitifitas}{2} \quad \textbf{(3.4)}$$

Sensitivitas merupakan kemampuan klasifikator untuk menunjukkan sebuah sinyal mana yang merupakan benar-benar kondisi normal dari seluruh populasi kondisi jantung yang terklasifikasikan kedalam kondisi normal. Spesitifitas untuk menunjukkan sebuah sinyal mana yang merupakan benar-benar kondisi abnormal dari seluruh populasi kondisi jantung yang terklasifikasikan kedalam kondisi abnormal, istilah sensitifitas dan spesitifitas adalah istilah yang umum digunakan dalam menggambarkan kuantitas pada dunia medis [36]. Akurasi merupakan

jumlah prediksi yang benar dari keseluruhan prediksi yang diuji, artinya dalam penelitian ini akurasi menunjukkan tingkat ketepatan dari klasifikator untuk mampu mengklasifikasikan kondisi berlabel normal terklasifikasi kedalam kondisi normal dan kondisi berlabel abnormal kedalam kondisi abnormal dari keseluruhan data yang diuji, dan AUC merupakan sebuah variabel evaluasi yang umum digunakan untuk mengukur akurasi algoritma pada *machine learning*. Pada penelitian ini AUC merujuk pada ketepatan pendistribusian kondisi normal dan abnormal dalam proses klasifikasi, kondisi normal dan abnormal ini juga merujuk pada nilai sensitivitas (*true positive rate)* dan spesitifitas (*true negative rate)* [37].

3.2 ALUR PENELITIAN



Gambar 3.4. Alur Penelitian

Alur penelitian digunakan untuk menjelaskan alur penelitian secara runtut dan mempermudah tahapan pelaksanaan penelitian. Berdasarkan pada Gambar 3.4 Penelitian ini dilakukan dalam beberapa tahapan yaitu tahap studi literasi, pengumpulan data, perancangan sistem, pembuatan program menggunakan *software Spyder* pengujian sistem dan yang terakhir analisis uji sistem dan kesimpulan. Secara rinci tahapan pada alur penelitian dijelaskan melalui poin-poin berikut.

3.2.1. Studi Literasi

Penelitian ini dimulai dengan studi literasi yang berkaitan dengan topik yang akan diteliti. Pada tahapan ini dilakukan pengkajian informasi mengenai sinyal jantung, metode ekstraksi sinyal, dan metode klasifikasi dari sinyal PCG berdasarkan penelitian yang pernah dilakukan sebelumnya. Tahap ini dilakukan evaluasi, identifikasi, dan perumusan masalah yang telah dikaji oleh peneliti

sebelumnya. Dari hasil tersebut kemudian dilakukan analisis yang kemudian dapat dijadikan sebagai sumber pedoman pada penelitian ini.

Perumusan masalah pada tahap studi literasi mencakup tujuan yang akan dicapai pada penelitian ini, objek dan metode yang akan digunakan. Didapatkan hasil penelitian ini akan meneliti Klasifikasi sinyal PCG menggunakan STFT sebagai ekstraksi fitur dan SVM sebagai metode klasifikasi.

3.2.2. Pengumpulan Data

Pengumpulan data dilakukan dengan observasi dataset yang terdapat di internet yang pernah di lakukan uji dengan metode yang berbeda namun pada objek yang sama. Pada penelitian ini digunakan *Classification of Heart Sound Recordings - The PhysioNet Computing in Cardiology Challenge 2016* didapat dari laman *https://physionet.org/content/challenge-2016/1.0.0/*.

3.2.3. Perancangan Sistem

Rancangan sistem dibuat diagram alur pengujian sistem dengan tahapan-tahapan seperti, *pre-processing* sinyal yaitu melakukan proses labeling data, menentukan besar data sampling, serta penambahan *zero padding* untuk perhitungan STFT. Ekstraksi fitur menggunakan metode STFT, dan klasifikasi *Support Vector Machine* (SVM) menggunakan metode *support vector class* untuk 2 kelas dan 3 jenis *kernel*, yaitu *Linear*, *Polynomial*, dan RBF. Proses tahapan ini nantinya akan direpresentasikan dalam sebuah program.

3.2.4. Pembuatan Program

Pembuatan program dilakukan dengan menggunakan *software spyder*. Program ini dibuat kedalam suatu *source code* yang mencakup tahapan-tahapan dari perancangan sistem yang telah dibuat. Pada program ini nantinya *input* dari datanya bersumber dari dataset yang telah ditentukan.

3.2.5. Pengujian Sistem

Setelah pembuatan *source code* selesai dilakukan, maka tahapan selanjutnya adalah dengan melakukan pengujian sistem. Pengujian sistem dilakukan dengan meng*input* dataset suara jantung atau hasil rekaman PCG sebanyak 2141 data, yang akan dibagi kedalam data training dan data testing untuk ditampilkan hasil klasifikasinya. Proses pengujian akan melewati dua metode yang digunakan yaitu *Short time fourier transform* (STFT) untuk ekstraksi fitur dan *Support Vector*

Machine (SVM). Pada proses ekstraksi fitur pengujian dilakukan dengan memberikan variasi pada jenis *window* untuk menghasilkan spektrogram dan nilai informasi yang dibutuhkan untuk hasil ekstraksi fitur yang optimal. Variasi jenis *window* yang dilakukan yaitu *window hamming, hanning,* dan *blackman*. Proses klasifikasi juga akan memasukkan nilai-nilai parameter yang berpengaruh terhadap hasil pengujian dengan memvariasikan nilai C parameter dan jenis *kernel*. Jenis *kernel* yang di variasikan meliputi *kernel linear, polynomial,* RBF, masing-masing jenis *kernel* akan dimasukkan nilai C parameter yang divariasikan terdiri dari 7 level yaitu 0.0001, 0.001, 0.01, 1, 10, 100, 1000 untuk optimasi hasil dari evaluasi kinerja sistem

3.2.6. Analisis Uji Sistem

Analisis hasil pengujian akan membandingkan nilai *kernel* dan jenis *kernel* terbaik untuk klasifikasi normal dan abnormal dari sinyal PCG berdasarkan hasil evaluasi *kfold cross validation* dan *confusion matrix*. Dan tahapan terakhir yaitu membuat kesimpulan dari penelitian dengan tujuan menjawab rumusan permasalahan yang menjadi dasar utama pelaksanaan penelitian ini.