

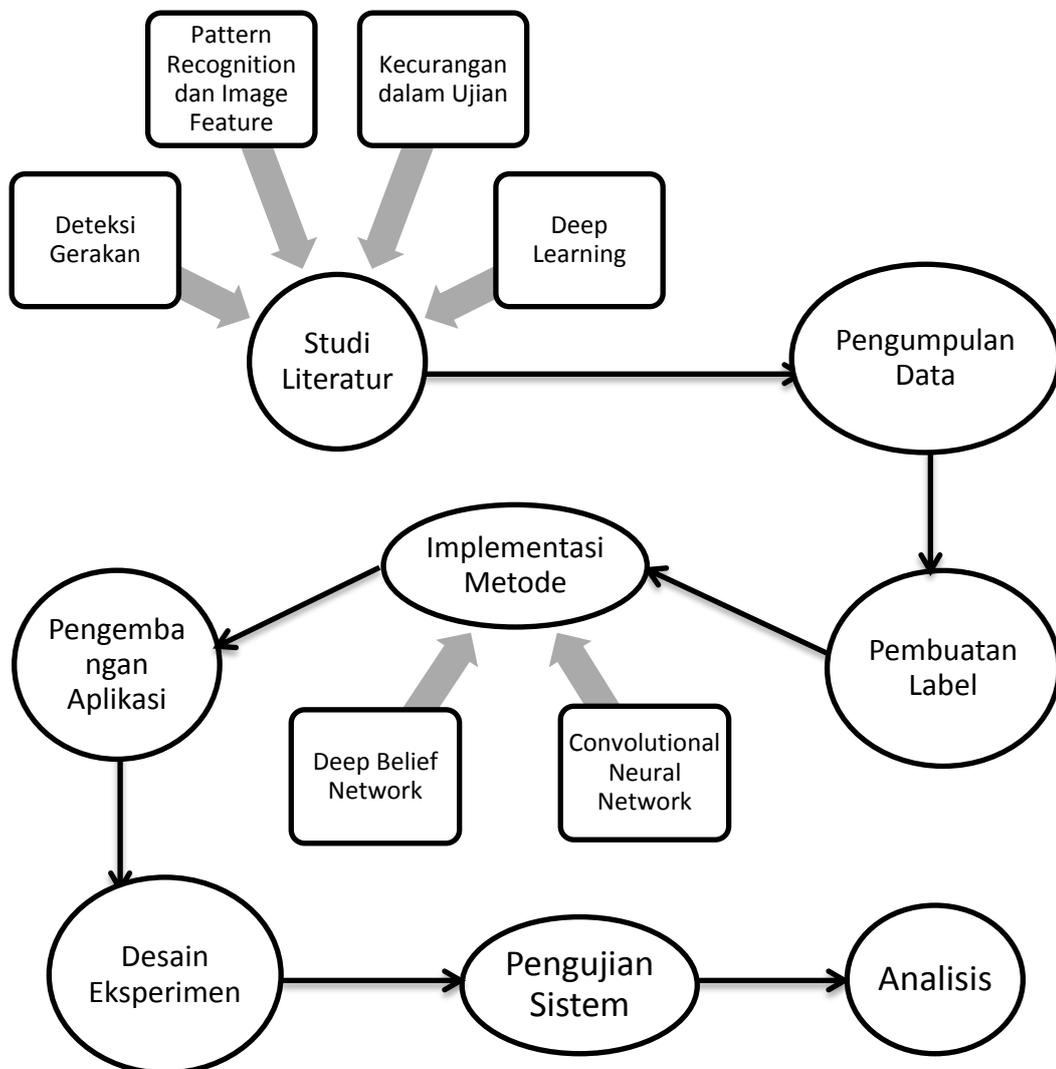
BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan secara menyeluruh metodologi yang digunakan untuk melakukan penelitian deteksi potensi kecurangan ujian berdasarkan gerakan peserta didik dengan menggunakan metode *deep learning*.

1.1 Desain Penelitian

Desain dan tahapan alur penelitian ini ditunjukkan pada Gambar 3.1



Gambar 3.1 Desain Penelitian

Gambar 3.1 menunjukkan delapan tahapan dalam menyelesaikan penelitian ini. Berikut merupakan penjelasan dari masing-masing tahapannya.

3.1.1 Studi Literatur

Studi literatur dilakukan dengan mencari sumber terkait dengan penelitian yang dilakukan, diantaranya adalah penelitian tentang mendeteksi gerakan tidak normal pada video pemantauan, pengenalan pola, ciri pada citra, kecurangan dalam ujian dan teori tentang *deep learning* beserta model-modelnya, yaitu Deep Belief Network dan Convolutional Neural Network. Pada tahap ini juga terdapat pembahasan tentang penelitian-penelitian sebelumnya yang berkaitan dengan penelitian yang dilakukan oleh penulis. Penjelasan literatur dan teori-teori yang mendukung penelitian ini dapat dilihat pada BAB II.

3.1.2 Pengumpulan Data

Pengumpulan data didapatkan dengan mencari 10 mahasiswa untuk dijadikan model pelaku simulasi kegiatan ujian. Dari 10 mahasiswa tersebut dibagi menjadi dua kelompok, yaitu kelompok untuk data pelatihan serta data pengujian yang masing-masing berjumlah lima orang. Setiap mahasiswa secara individu direkam dengan menggunakan kamera DSLR di ruangan kelas. Kamera diletakkan di depan mahasiswa dengan posisi tinggi yang dibantu oleh tripod agar terlihat seperti direkam dari atas. Pada pengumpulan data, digunakan dua ruangan kelas yang berbeda dikarenakan tidak diselesaikan dalam satu hari.

Karena jumlah gerakan tidak normal yang ditentukan adalah enam gerakan, maka setiap mahasiswa direkam sebanyak enam kali. Setiap satu kali rekaman merepresentasikan satu gerakan dan berdurasi sekitar 10 detik. Skenario simulasi kegiatan ujian yang dilakukan adalah setiap mahasiswa mengerjakan soal dengan posisi normal selama lima detik, kemudian melakukan satu gerakan tidak normal yang telah ditentukan selama dua detik dan duduk kembali hingga detik ke sepuluh.

3.1.3 Pembuatan Label

Video yang telah dikumpulkan selanjutnya diberi label. Label tersebut merepresentasikan kriteria gerakan yang berpotensi menuju kecurangan pada ujian yang telah disebutkan pada sub bab batasan masalah. Pemberian label ini berfungsi sebagai indikator dalam menguji keakuratan dari sistem pendeteksi yang dikembangkan.

3.1.4 Implementasi Metode

Setelah setiap video diberi label, langkah selanjutnya adalah memulai untuk mengimplementasikan kedua model dari metode *deep learning* yang diusulkan, yaitu Deep Belief Network dan Convolutional Neural Network untuk membuat sistem pendeteksi gerakan yang berpotensi menuju kecurangan pada ujian. Implementasi metode dijelaskan pada Sub Bab 3.4.

3.1.5 Pengembangan Aplikasi

Setelah implementasi metode dilakukan, selanjutnya dibangun perangkat lunak atau aplikasi dari sistem pendeteksi gerakan yang berpotensi menuju kecurangan pada ujian. Pengembangan aplikasi ini menggunakan metode Waterfall, yaitu metode pengembangan aplikasi yang terstruktur dan berurutan, dimulai dari analisis kebutuhan aplikasi, desain, koding hingga uji coba. Berikut merupakan penjelasan lebih lanjut tentang proses-proses pada metode Waterfall (Toni, 2015).

1. Analisis kebutuhan aplikasi

Pada tahap ini diharuskan mendapatkan hal-hal yang dianggap dapat menunjang penelitian yang dilakukan, seperti mencari permasalahan yang ada, mengumpulkan data fisik maupun non fisik, dan lain-lain.

2. Desain

Pada tahap ini, ditentukan tampilan antar muka (*user interface*), alur kerja sistem, cara pengoperasian sistem dan *output* yang dihasilkan oleh sistem

3. Koding

Pada tahap ini, dimasukkan *script* kode pemrograman ke dalam sebuah perangkat lunak *programming* untuk menghasilkan aplikasi yang telah didesain

4. Uji Coba

Pada tahap ini, dilakukan uji coba terhadap aplikasi yang dibuat untuk mengetahui seperti apa hasil kinerja sistem yang telah dibangun.

3.1.6 Desain Eksperimen

Desain eksperimen berisi skenario untuk melakukan uji coba pada sistem yang dibuat. Rincian skenario yang telah ditentukan dijelaskan pada BAB IV.

3.1.7 Pengujian Sistem

Pada tahapan ini, akan dilakukan pengujian terhadap sistem yang dibangun berdasarkan skenario yang telah ditentukan untuk diketahui kinerjanya. Selain itu, tingkat akurasi dari *output* yang dihasilkan oleh implementasi *deep learning* dapat diketahui pula pada tahap ini. Hasil pengujian kemudian dianalisa untuk diuji kebenarannya.

3.1.8 Analisis

Hasil *output* pada pengujian sistem kemudian dianalisa kebenarannya dan juga dianalisa hal apa saja yang menyebabkannya. Selain itu, tingkat akurasi dari kedua model *deep learning* yang digunakan, yaitu Deep Belief Network dan Convolutional Neural Network dibandingkan untuk diketahui mana yang terbaik dalam mendeteksi gerakan yang berpotensi menuju kecurangan pada ujian.

1.2 Alat dan Bahan Penelitian

Untuk memenuhi penelitian ini alat dan bahan yang digunakan adalah:

1. Kamera Digital Single Lens Reflex (DSLR)
2. Laptop dengan prosesor Intel Core i5 2.50 GHz dan RAM 4 GB
3. Sistem operasi Microsoft Windows 7 64 bit
4. Aplikasi Matlab
5. YTD Video Downloader

1.3 Data Penelitian

Data yang digunakan dalam penelitian ini berupa rekaman video simulasi kegiatan ujian. Setiap video berisi skenario adegan peserta ujian yang menunjukkan gerakan yang berpotensi pada kecurangan saat mengerjakan soal ujian. Data penelitian ini dibagi menjadi data *input* dan data *output*.

3.3.1 Data Input

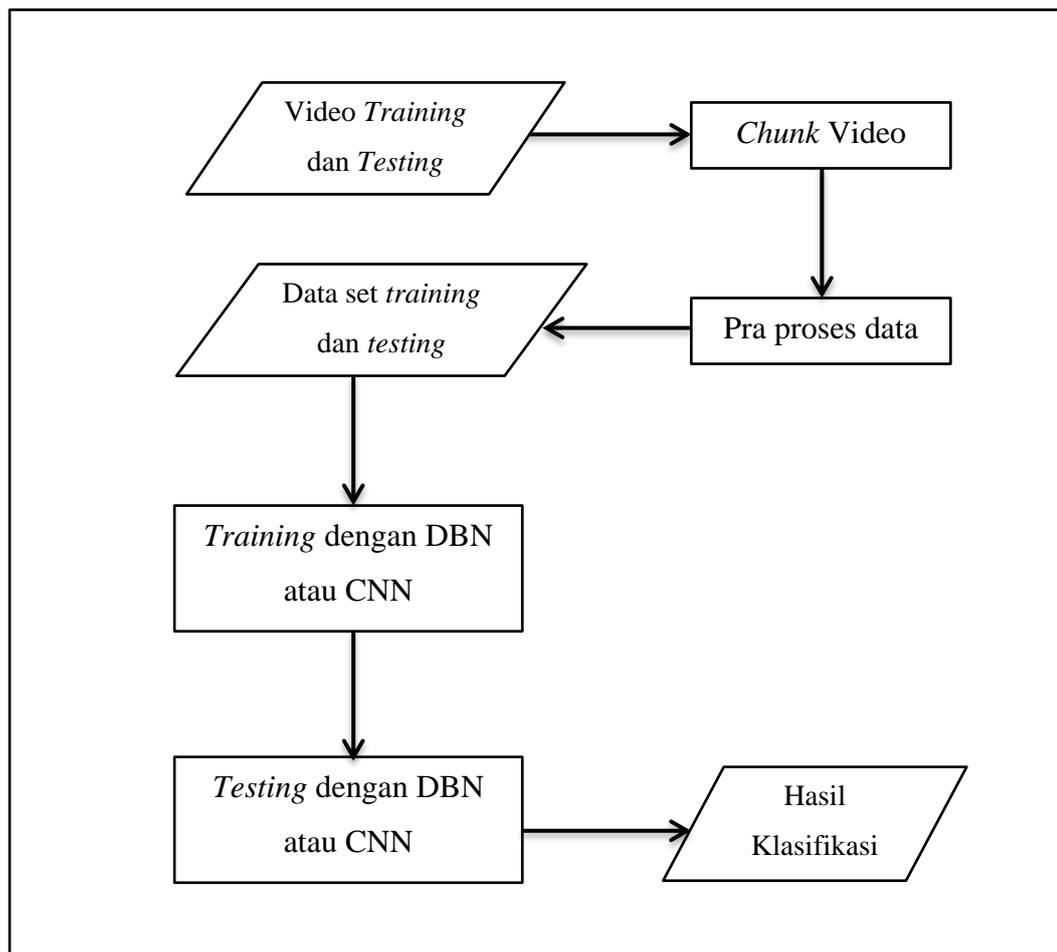
Terdapat dua macam data *input* yang digunakan dalam penelitian ini. Data pertama merupakan *file* video singkat yang berdurasi empat hingga delapan detik dengan format AVI. Sedangkan data *input* kedua adalah *file* variabel Matlab dengan format MAT. *File* tersebut berisi label yang menandai jenis gerakan yang berpotensi pada kecurangan ujian untuk setiap video data *input*.

3.3.2 Data Output

Data *output* yang dihasilkan berupa klasifikasi dari setiap video yang dijadikan sebagai data *input*. Hasil klasifikasi tersebut kemudian dianalisa agar sistem deteksi kegiatan tidak normal ini dapat diketahui tingkat akurasinya.

1.4 Implementasi Metode

Pada tahap implementasi metode, data video yang telah dikumpulkan dibagi menjadi dua kelompok, yaitu data *training* dan data *testing*. Data *training* merupakan data yang digunakan untuk membuat sistem mempelajari fitur-fitur yang ada pada video dan melakukan klasifikasi terhadap fitur-fitur tersebut. Sedangkan data *testing* merupakan data yang digunakan untuk menguji akurasi hasil klasifikasi fitur-fitur data *training* yang telah dipelajari oleh sistem. Data video yang dijadikan data *training* dan *testing* masing-masing berjumlah 30 data. Kemudian terhadap kedua kelompok data tersebut dilakukan proses-proses utama pembangunan sistem. Alur proses pengembangan sistem ditunjukkan pada Gambar 3.2



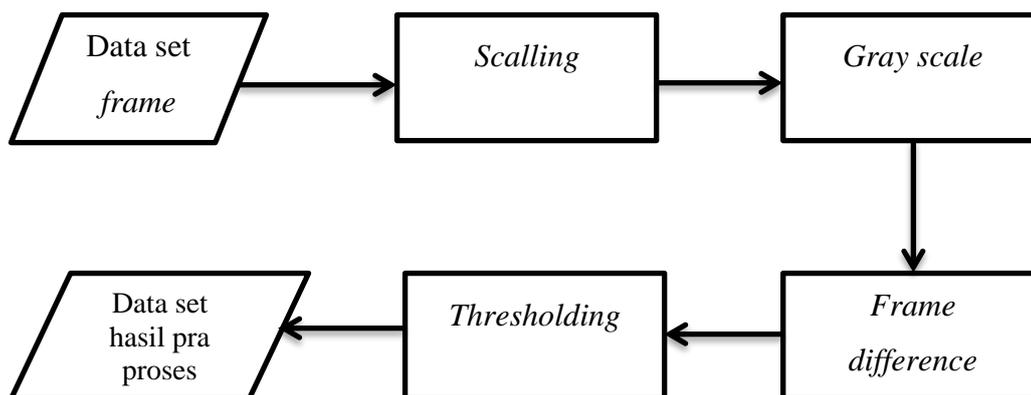
Gambar 3.2 Alur Sistem Deteksi Potensi Kecurangan

3.4.1 *Chunk Video*

Untuk membuat sistem pendeteksi lebih mudah dalam mempelajari fitur-fitur yang tersedia pada video, video terlebih dahulu diekstrak setiap *frame*-nya menggunakan aplikasi Matlab, yang mana proses ini disebut *chunk video*.

3.4.2 Pra Proses Data

Setelah dilakukan proses *chunk* pada video dan didapatkan setiap *frame*-nya, dilakukan pra proses pada setiap *frame* terlebih dahulu agar mempermudah dalam memprosesnya. Alur pra proses dibuat dalam bentuk diagram alur yang ditunjukkan pada Gambar 3.3



Gambar 3.3 Diagram Alur Pra Proses Data

1. *Scalling*

Pertama yang dilakukan pada setiap *frame* adalah *scalling*, yaitu mengubah ukuran resolusi *frame* (*resize*) yang awalnya 452 x 218 piksel menjadi 28 x 28 piksel (Deng, 2012).

2. *Gray Scale*

Setiap citra berwarna memiliki tiga jenis karakteristik, yaitu panjang, lebar dan warna. Citra seperti ini disebut citra tiga dimensi. Citra tiga dimensi dapat menjadi dua dimensi dengan cara membuang warna yang terdapat pada citra, sehingga membuat citra hanya memiliki warna hitam dan putih. Proses pengubahan citra ini disebut *gray scalling*, yang bertujuan untuk mempercepat proses *training* pada citra bagi sistem deteksi.

3. *Frame Difference*

Karena data *input* berupa kumpulan *frame* yang membentuk video, maka untuk mendeteksi gerakan yang terdapat pada kumpulan *frame* tersebut salah

satunya adalah dengan cara melihat perbedaan yang ada di setiap *frame* tersebut. Proses ini dinamakan *frame difference*, yang dilakukan dengan cara membandingkan *frame* ke- i dengan *frame* ke $i + 1$. Proses ini diulangi sebanyak jumlah *frame* pada setiap video. Lalu setiap perbandingan dijumlahkan dan dihitung rata-ratanya yang hasilnya membentuk sebuah citra abstrak, sehingga setiap satu citra abstrak merupakan representasi dari satu video.

4. *Thresholding*

Setelah didapatkan citra abstrak, maka selanjutnya dilakukan proses *thresholding*, yaitu memisahkan antara obyek dan backgroundnya yang terdapat pada citra. Hal ini dilakukan dengan mengubah citra abstrak ke dalam bentuk biner. *Thresholding* dilakukan untuk menyederhanakan bentuk data *input* agar mempermudah sistem dalam mengenali fitur pada obyek yang akan dideteksinya.

3.4.3 *Training dan Testing dengan Deep Learning*

Setelah dilakukan pra proses, didapatkanlah data set yang akan digunakan untuk proses *training* model *deep learning*-nya. Proses *training* yang dilakukan adalah ekstraksi ciri pada citra, yang merupakan proses utama bagi *machine learning* untuk mempelajari data citra dan mengekstraksi fitur-fitur yang ada pada data citra tersebut. Pada tahap ini, diimplementasikan metode Deep Belief Network (DBN) dan Convolutional Neural Network (CNN) untuk mempelajari hingga mengekstraksi gerakan yang berpotensi menuju kecurangan pada data *input*, lalu mengklasifikasikannya dan menghitung tingkat akurasi hasil klasifikasinya. Implementasi dilakukan dengan menggunakan *toolbox* yang disediakan oleh Mathworks yang bernama Deep Learning Toolbox. *Toolbox* ini menyediakan *pseudocode* implementasi model DBN dan CNN. Berikut adalah penjelasan mengenai tahapan-tahapan ekstraksi ciri dan klasifikasi pada masing-masing model *deep learning*:

1. Ekstraksi Ciri dan Klasifikasi pada DBN

Berikut merupakan langkah ekstraksi ciri dan klasifikasi pada DBN:

a. Jumlah *Hidden Layer*

Pada DBN, tahap yang dilakukan pertama kali pada tahap ekstraksi ciri adalah menentukan jumlah *hidden layer* pada DBN. Seperti yang telah dijelaskan pada sub bab 2.6 mengenai metode DBN, jumlah *hidden layer* DBN dapat disesuaikan sesuai kebutuhan. Jumlah yang ditentukan adalah 100 *hidden layer* (Deng, 2012), bobot serta bias akan secara otomatis disesuaikan sesuai jumlah *hidden layer* oleh fungsi ‘*dbnsetup*’ yang terdapat pada *toolbox*.

b. Jumlah *Epoch* dan *Batch*

Jumlah *epoch* untuk yang dicoba bervariasi agar dapat diketahui tingkat akurasi pada masing-masing *epoch*, mulai dari 100 hingga 10.000. Sedangkan jumlah *batch* yang digunakan adalah 3 dan 30.

c. Proses *Training* dan *Testing*.

Pada *toolbox* yang digunakan, model DBN menggunakan model *deep learning* lainnya untuk melakukan klasifikasi, yaitu Neural Network (NN). Sebelum melakukan klasifikasi, dilakukan *training* terlebih dahulu pada NN. Parameter yang diperlukan dalam *training* NN adalah hasil *training* pada DBN serta kelas data yang telah dibangun secara manual. Setelah dilakukan proses *training* pada NN, barulah dilakukan *testing* pada NN untuk melihat hasil klasifikasi data video yang dijadikan sebagai data *input*.

2. Ekstraksi Ciri dan Klasifikasi pada CNN

Pada ekstraksi ciri dan klasifikasi menggunakan metode CNN menggunakan empat *layer* CNN, yaitu dua *layer* konvolusi dan dua *layer* *subsampling*. Keempat *layer* tersebut merupakan *hidden layer* seperti pada DBN. Berikut merupakan langkah ekstraksi ciri dan klasifikasi pada CNN:

a. *Convolution Layer*

Menghitung *convolution layer* menggunakan teknik konvolusi yang telah dijelaskan pada BAB II, yaitu menentukan jumlah *output map* yang akan digunakan pada *convolution layer*. *Output map* adalah jumlah neuron di tiap

convolution layer. Jumlah *output map* pada *layer* konvolusi yang pertama yaitu enam dan *layer* konvolusi yang kedua adalah 12. Langkah ketiga menentukan jumlah kernel yang digunakan pada 2 *layer* konvolusi yaitu ukuran 5x5 piksel (Deng, 2012).

b. *Subsampling Layer*

Pada *subsampling layer* digunakan proses perhitungan *max pooling* dengan skala matriks 2x2.

c. Jumlah *Epoch* dan *Batch*

Sama halnya dengan implementasi pada DBN, Jumlah *epoch* yang dicoba adalah 100 hingga 10.000 dan *batch*-nya adalah 3 dan 30 untuk masing-masing *epoch*.

d. Proses *Training* dan *Testing*

Proses *training* menggunakan jaringan *backpropagation*, sedangkan proses *testing* menggunakan jaringan *feed forward*. Nilai awal *kernel* pada proses *training* adalah *random*, lalu nilai *kernel* tersebut diperbaharui setiap melakukan *training* untuk menentukan pola klasifikasi. Sedangkan pada proses *testing* menggunakan nilai *kernel* yang telah melalui proses *training*. Sama seperti pada DBN, hasil klasifikasi data video dapat dilihat setelah menjalankan proses *testing*.