

BAB III

METODOLOGI PENELITIAN

Pada bagian ini akan dijelaskan mengenai metodologi yang digunakan dalam melakukan penelitian ini. Metode merupakan cara urutan pengerjaan yang nantinya akan digunakan dalam penelitian ini. Selain itu metodologi juga menentukan *output* yang diharapkan dari setiap masukan yang ada. Tujuan metodologi dari penelitian ini adalah agar proses yang ada menjadi lebih teratur dan sistematis. Sehingga mudah memantau perkembangan dan tingkat keberhasilan.

3.1 Metode Pengumpulan Data

Untuk memperoleh data yang dibutuhkan sebagai pendukung pembuatan sistem ini, metode yang digunakan untuk melakukan pengumpulan data yaitu :

1. Studi Pustaka

Metode ini dilakukan dengan cara mengumpulkan data melalui sumber-sumber bacaan yang terkait dengan data yang dibutuhkan, sehingga penulis dapat menganalisis data yang akan disusun dalam menunjang proses penelitian.

2. Observasi

Metode ini merupakan pengumpulan data dengan cara dimana penulis terlibat langsung dalam situasi nyata masalah yang akan dibahas, lalu membuat perancangan aplikasi yang akan dibangun.

3. Dokumentasi

Metode ini dilakukan dengan mengambil foto komponen *cylinder head* pada motor 4 tak di Bengkel GIGA motor, JL. Samratulangi Bandar Lampung mengenai mendeteksi kerusakan komponen *cylinder head* motor 4 tak dengan metode *template matching*.

3.2 Kebutuhan Aplikasi

Adapun analisis kebutuhan *software* yang digunakan untuk membangun sebuah aplikasi mendeteksi kerusakan komponen *cylinder head* motor 4 tak adalah sebagai berikut :

a. Analisis *software*

Software yang digunakan merupakan perangkat lunak komputer yang digunakan sebagai media pembuatan dan menjalankan perintah pada aplikasi yang akan dibuat. Adapun spesifikasi *software* yang diperlukan adalah :

- Sistem operasi *Microsoft Windows 10 Professional*.
- *Microsoft Visual Studio 2013*
- *OpenCV 2.4.11*

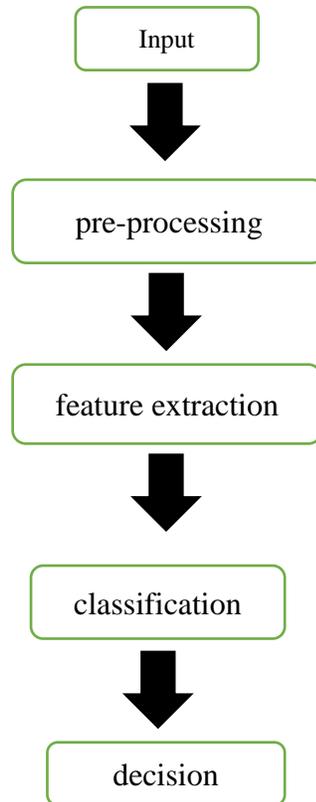
b. Analisis *hardware*

Hardware yang digunakan merupakan perangkat keras komputer yang digunakan sebagai media pembuatan dan menjalankan perintah pada aplikasi yang akan dibuat. Adapun spesifikasi *hardware* yang diperlukan adalah :

- Laptop *Asus X450C*.
- *Processor Intel(R)Celeron(R) CPU 1017U @1.60 GHz*.
- RAM 2 GB.
- *Harddisk 500 GB*.

3.3 Tahapan Metode *Template Matching*

Metode yang digunakan adalah metode *Template Matching Correlation*. Secara umum tahap-tahap yang dilakukan pada metode tersebut antara lain yaitu *input, pre-processing, feature extraction, classification dan decision*. Seperti yang ditunjukkan pada gambar 3.1 dibawah ini:



Gambar 3.1 Tahapan Metode *Template Matching*

Tahapan pada metode *Template Matching Correlation*.

1. *Input*

Pada proses ini akan diinputkan citra uji gambar Busi yang masih bagus dimana gambar tersebut formatnya berupa .jpg.

2. *Preprocessing*

Proses ini digunakan untuk penyamaan ukuran matriks yang nantinya akan dicocokkan dengan algoritma *Template Matching Correlation*. Terdapat 3 hal yang dilakukan pada tahap ini yaitu:

- *Resize*

Pada tahap ini dilakukan perubahan ukuran citra. Ada kalanya ukuran citra berubah menjadi lebih kecil dari file aslinya, namun bisa juga terjadi yang sebaliknya. Tujuan dari proses ini adalah agar matriks citra yang dicocokkan dengan metode *Template Matching Correlation* akan sama dengan *Template* citra gambar yang sudah ada.

- *Grayscale*

Pada tahap ini dilakukan perubahan warna citra menjadi keabu-abuan. Tujuan dari proses ini adalah untuk mempermudah pada saat citra masuk ke tahap *threshold*. *Grayscale* sendiri merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, atau dengan kata lain bagian $Red = Green = Blue$. Nilai tersebut, digunakan untuk menunjukkan suatu tingkat intensitas. Pada proses ini, masing-masing piksel RGB (*Red Green Blue*) dari citra diambil nilainya, dihitung rata-ratanya dari ketiga nilai RGB tersebut, dan diinisialisasi dengan nilai rata-ratanya sehingga tercipta warna keabu-abuan dari matriks citra yang telah dilakukan proses grayscale.

- *Threshold*

Pada tahap ini dilakukan pengaturan warna pada tiap piksel berdasarkan nilai dari *intensity threshold*. Proses ini bertujuan untuk memisahkan *foreground* dengan *back ground* dari suatu citra. Selain itu juga untuk mengubah warna citra menjadi biner dan penghilangan noisedari citra.

3. *Feature extraction*

Ekstraksi fitur untuk mengenali atau mengklasifikasikan obyek dalam gambar, seseorang harus terlebih dahulu ekstrak beberapa fitur keluar gambar, dan kemudian menggunakan fitur ini di dalam pola untuk memperoleh *classifier final* kelas. Fitur ekstraksi (deteksi) bertujuan untuk mencari daerah fitur yang signifikan pada gambar tergantung pada karakteristik intrinsik mereka dan aplikasi. Wilayah tersebut dapat di definisikan dalam lingkungan *global* atau *local* dan dibedakan oleh bentuk, tekstur, ukuran, intensitas, sifat statistik dan sebagainya. Metode ekstraksi fitur lokal dibagi menjadi

intensitas sesuai dan struktur berbasis. Intensitas metode berbasis analisis pola intensitas lokal untuk menemukan daerah yang memenuhi keunikan yang diinginkan atau stabilitas kriteria. Struktur berbasis metode mendeteksi struktur gambar seperti tepi, garis, sudut, lingkaran, elips dan sebagainya. Fitur ekstraksi cenderung untuk mengidentifikasi ciri-ciri yang dapat membentuk representasi yang baik dari objek, sehingga dapat membedakan kategori objek, sehingga dapat membedakan kategori objek dengan variasi toleransi.

4. *Classification*

Tindakan untuk memberikan kelompok pada setiap keadaan. Setiap keadaan berisi sekelompok atribut, salah satunya adalah class attribute. Metode ini butuh untuk menemukan sebuah model yang dapat menjelaskan class attribute itu sebagai fungsi dari input attribute.

5. *Decision*

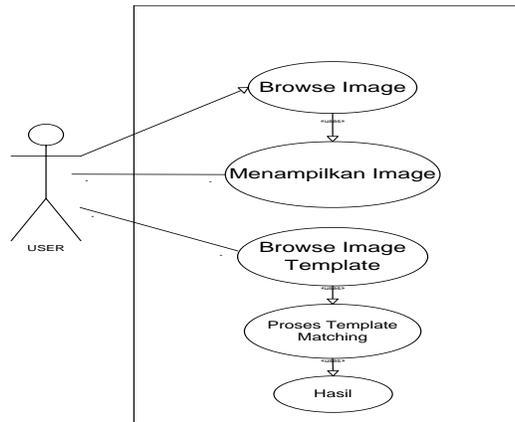
Keputusan (*decision*) adalah membuat pilihan di antara beberapa kemungkinan.

3.4 Perancangan

Tahap perancangan pada aplikasi yang akan dibangun adalah rancangan *usecase* diagram, *sequence* diagram, *activity* diagram, dan rancangan antarmuka aplikasi pendeteksi kerusakan komponen *cylinder head* motor 4 tak dengan metode *template matching*.

3.4.1 *Usecase Diagram*

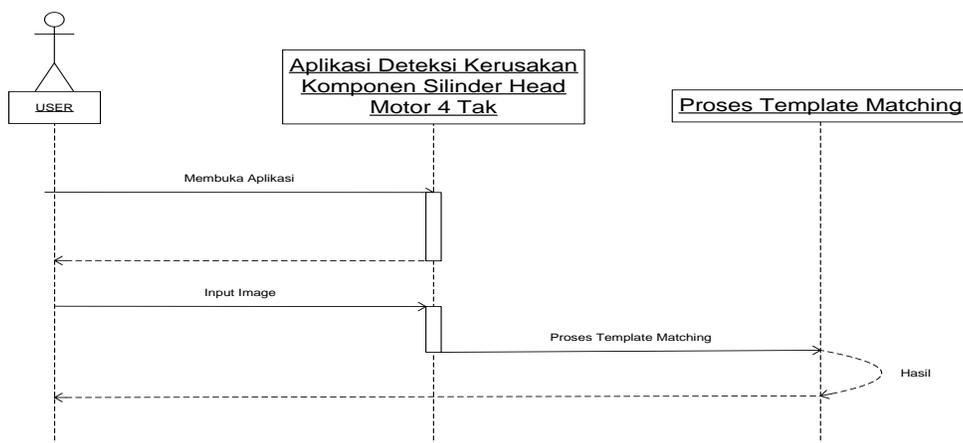
Usecase diagram merupakan *diagram* yang menjelaskan beberapa proses yang ada pada program *visual studio* yang nantinya akan dibangun. *Usecase* ini menjelaskan beberapa langkah yang dapat dilakukan oleh *user* Untuk melihat bagaimana dan apa saja proses yang dapat diakses oleh pengguna. dapat dilihat pada gambar 3.2 berikut ini:



Gambar.3.2 *Usecase Diagram*

3.4.2 *Sequence Diagram*

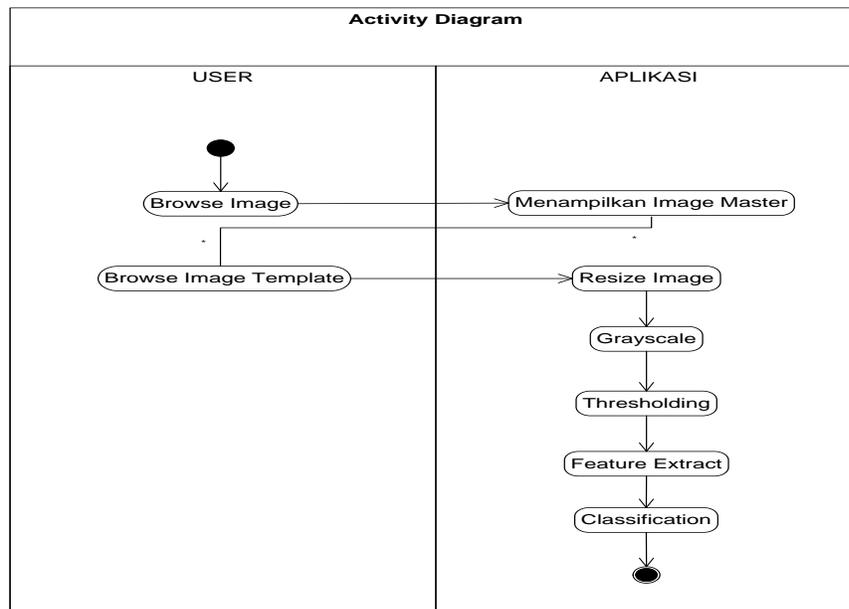
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* dapat dilihat pada gambar 3.3 dibawah ini:



Gambar 3.3 *Sequence Diagram*

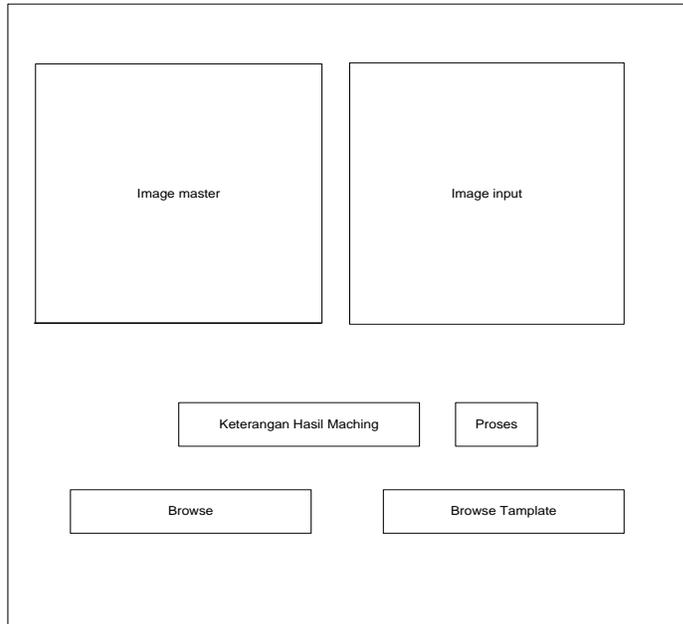
3.4.3 Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah aksi dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Untuk mengilustrasikan proses-proses paralel digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertical. *Activity diagram* dapat dilihat pada gambar 3.4 berikut:



Gambar 3.4 Activity Diagram

3.5 Rancangan Antarmuka Aplikasi Pendeteksi Kerusakan Komponen *Cylinder Head Motor 4 Tak* Dengan Metode *Template Matching*



Gambar 3.5 Rancangan antar muka

Pada rancangan antar muka program pada gambar 3.5 terdapat 3 button yaitu browse, browse template dan proses. Pada button browse digunakan untuk mengambil file gambar untuk menjadi file master *template matching*. Sedangkan pada button browse template digunakan untuk mengambil file gambar inputan. Sedangkan Pada botton proses untuk melakukan proses *template matching* dengan membandingkan dengan file master.

