

BAB IV

PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan

Bab ini membahas tahapan pengembangan perangkat lunak sistem pendeteksian wajah dengan *SAHI* menggunakan metode *Rational Unified Process (RUP)*.

4.2 Fase Insepsi

Fase insepsi membahas mengenai kebutuhan dasar perangkat lunak, pemodelan bisnis yang mendasari, dan identifikasi kebutuhan fungsional dan non-fungsional perangkat lunak.

4.2.1 Pemodelan Bisnis

Penelitian ini membuat sistem untuk memenuhi kebutuhan pendeteksian multi-wajah. Sistem ini menggunakan model *pre-trained* khusus wajah dan diuji menggunakan citra wajah mahasiswa. Sistem ini menghasilkan hasil keluaran yaitu visualisasi hasil deteksi wajah, hasil prediksi *bounding box*, dan nilai *IoU*.

4.2.2 Kebutuhan

Kebutuhan perangkat lunak dikategorikan menjadi dua kategori, yaitu kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional mencakup fitur-fitur yang harus dimiliki oleh sistem, sementara

kebutuhan non-fungsional mencakup aspek seperti performa, skalabilitas, keamanan, dan kemudahan penggunaan. Rincian kebutuhan fungsional dan non-fungsional dapat dilihat pada Tabel IV-1 dan Tabel IV-2.

Tabel IV-1. Kebutuhan Fungsional Perangkat Lunak

No	Kebutuhan Fungsional
1	Perangkat lunak dapat menerima masukan berupa citra digital dari pengguna dengan format jpg, jpeg, dan png.
2	Perangkat lunak dapat mendeteksi wajah-wajah yang terdapat pada citra digital yang dimasukkan dan menampilkan hasil deteksinya.

Tabel IV-2. Kebutuhan Non-Fungsional Perangkat Lunak

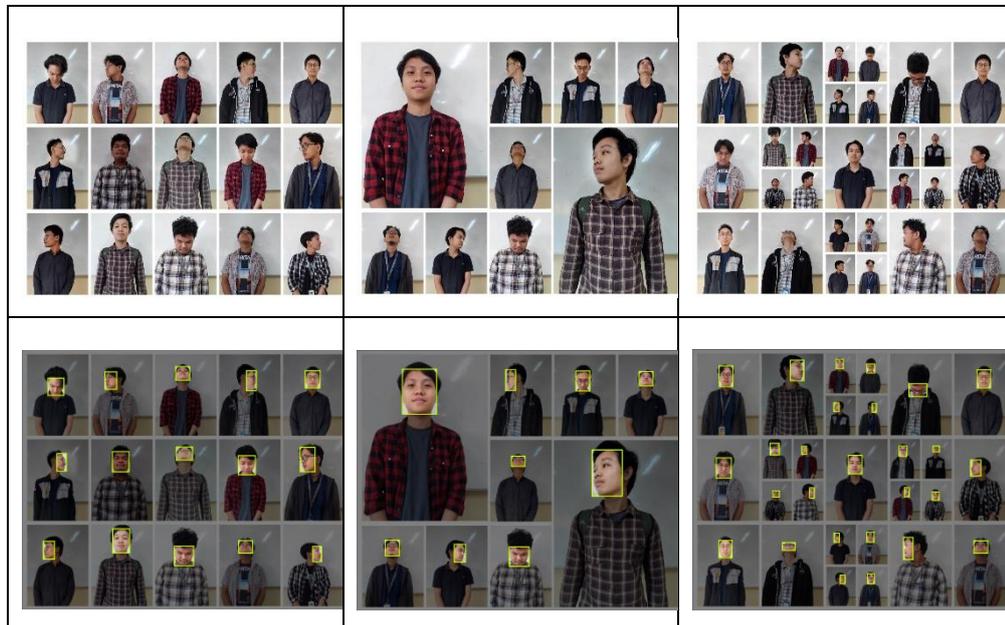
No	Kebutuhan Non-Fungsional
1	Perangkat lunak memiliki tampilan antarmuka yang mudah dipahami dan digunakan oleh pengguna.

4.2.3 Analisis dan Desain

Penelitian ini menggunakan data uji yang merupakan citra wajah mahasiswa yang diambil dengan kondisi orientasi arah wajah yang berbeda. Citra wajah tersebut disusun menjadi delapan citra uji yang masing-masing memiliki ukuran *file* sebesar 22,5 KB, 23,4 KB, 23,3 KB, 22,4 KB, 22,7 KB, 57,8 KB, 50,1 KB, dan 61,5 KB. Citra tersebut disusun untuk menguji hasil akurasi dan mengetahui total waktu deteksi dari program yang dirancang. Delapan citra uji tersebut disiapkan dengan memberikan anotasi *bounding box* sebagai *ground-truth data* untuk

mengukur nilai *intersection over union (IoU)* dari *bounding box* yang diprediksi (*predicted bounding box*).

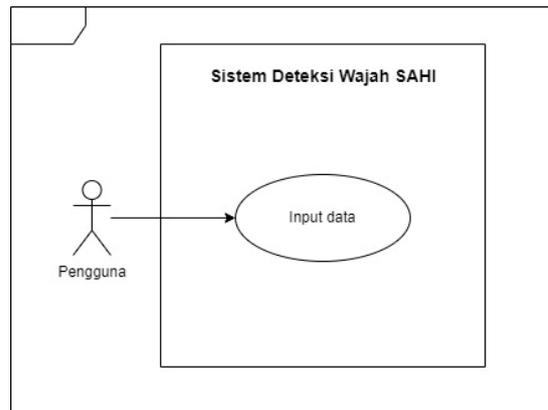
Tabel IV-3. Sampel Citra Uji dan Anotasi *Ground-Truth Bounding Box*



4.2.4 Implementasi

a) *Use Case Diagram*

Use Case Diagram digunakan untuk menggambarkan interaksi antara pengguna selaku aktor dan sistem dalam perangkat lunak. *Use case diagram* pada penelitian ini dapat dilihat pada gambar IV-1.



Gambar IV-1. *Use Case Diagram*

b) Definisi *Use Case*

Aktor merupakan pihak yang akan berinteraksi dengan perangkat lunak. Definisi aktor dan definisi *use case diagram* dapat dilihat pada Tabel IV-4 dan Tabel IV-5.

Tabel IV-4. Definisi Aktor

No	Aktor	Definisi
1	Pengguna	Pengguna adalah pihak yang dapat melakukan interaksi dengan perangkat lunak dan dapat mengakses seluruh fiturnya.

Tabel IV-5. Definisi *Use Case*

No	<i>Use Case</i>	Definisi
1	<i>Input data</i>	Memasukkan <i>file</i> yang akan digunakan.

c) Skenario *Use Case*

Skenario *use case* menjelaskan tahapan-tahapan yang dilakukan oleh pengguna dan reaksi perangkat lunak terhadap tindakan tersebut. Skenario *use case* ditunjukkan pada Tabel IV-6 dan Tabel IV-7.

Tabel IV-6. Skenario *Use Case Input Data*

Identifikasi	
No Use Case	SF-01
Nama Use Case	Input data
Tujuan	Memasukkan data untuk deteksi wajah yang ada.
Deskripsi	Pengguna mengunggah data berupa citra digital dengan format jpg, jpeg, dan png sebagai data uji untuk mendeteksi wajah.
Kondisi Awal	Sistem menampilkan halaman awal
Kondisi Akhir	Citra digital berhasil diunggah
Pemicu	Menekan tombol <i>browse file</i>
Skenario Utama	
Aksi Aktor	Reaksi Perangkat Lunak
1. Menekan tombol <i>browse file</i> .	
	2. Menampilkan jendela <i>file explorer</i> .
3. Memilih <i>file</i> gambar dengan format yang sesuai.	
4. Menekan tombol <i>open</i> .	

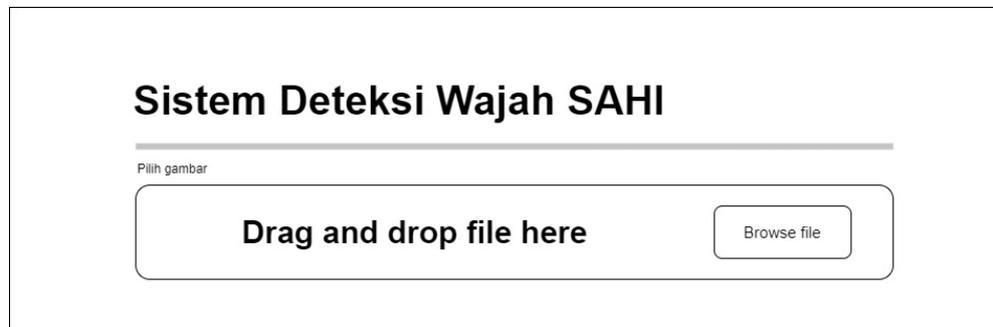
4.3 Fase Elaborasi

Fase elaborasi bertujuan untuk memperdalam analisa lebih lanjut setelah fase insepisi. Fase ini dilakukan pemodelan bisnis yang lebih rinci, identifikasi kebutuhan perangkat lunak, *activity diagram*, *sequence diagram*, dan rancangan tampilan antarmuka.

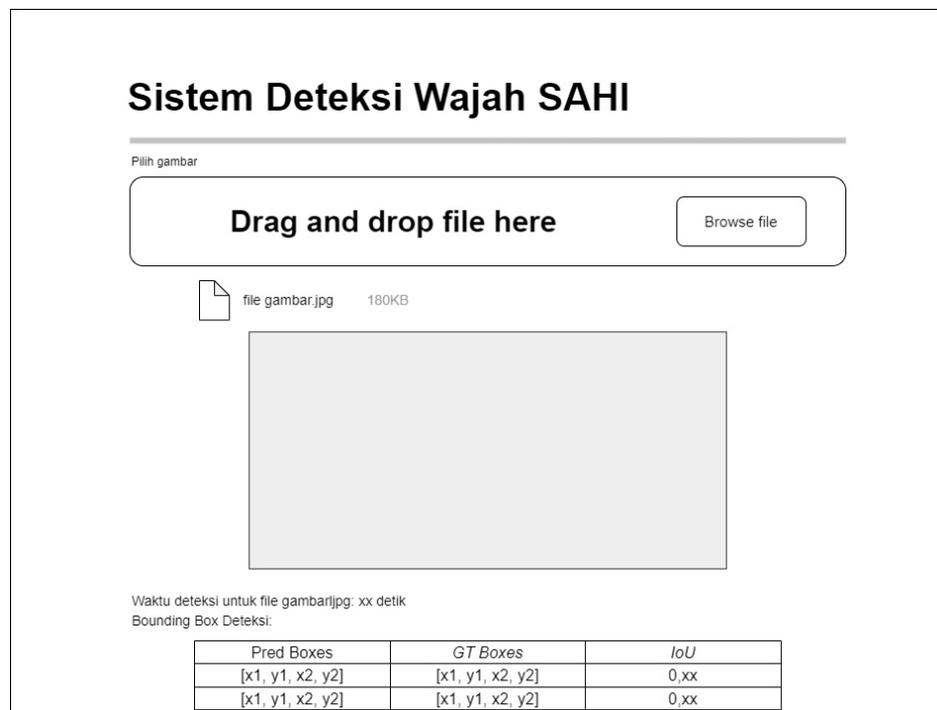
4.3.1 Pemodelan Bisnis

Rancangan tampilan antarmuka yang mudah dipahami dan mudah digunakan merupakan bentuk dari kebutuhan non-fungsional.

Tampilan antarmuka dirancang sederhana agar memudahkan pengguna dalam mengoperasikannya. Rancangan tampilan antarmuka pada penelitian ini dapat dilihat pada Gambar IV-2 dan Gambar IV-3.



Gambar IV-2. Rancangan Antarmuka Halaman



Gambar IV-3. Rancangan Antarmuka Tampilan Hasil Deteksi

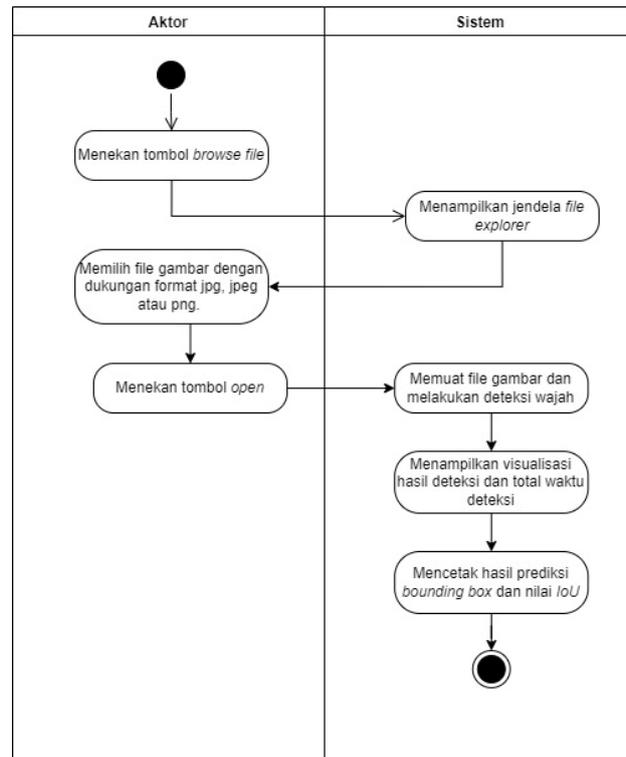
4.3.2 Kebutuhan

Pengembangan perangkat lunak dibangun dalam bahasa pemrograman *Python* dengan menggunakan *Visual Studio Code* untuk menuliskan kode program. Teknologi yang digunakan dalam penelitian ini yaitu menggunakan *Slicing Aided Hyper Inference (SAHI)*. Model deteksi yang digunakan adalah model deteksi *pre-trained You Only Look Once (YOLO)* versi 8 khusus deteksi wajah. Penelitian ini juga menggunakan *Scipy* untuk menyesuaikan antara *predicted bounding box* dan *ground-truth bounding box*.

4.3.3 Analisis dan Desain

a. Activity Diagram

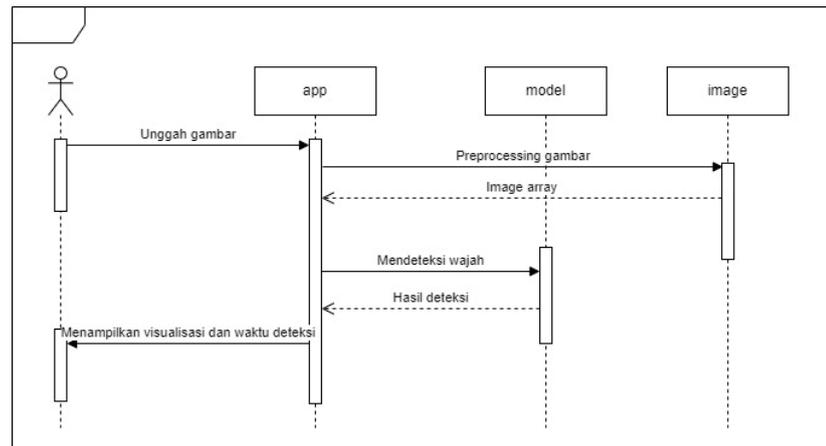
Activity diagram bertujuan untuk menggambarkan aktivitas pengguna terhadap sistem perangkat lunak, serta menggambarkan alur kerja sistem dari awal hingga akhir. *Activity diagram* pada penelitian ini dapat dilihat pada Gambar IV-4.



Gambar IV-4. Activity Diagram

b. Sequence Diagram

Sequence diagram bertujuan untuk menggambarkan interaksi antar berbagai komponen sistem dalam suatu skenario. Diagram ini menunjukkan urutan waktu interaksi antar komponen tersebut. *Sequence diagram* pada penelitian ini dapat dilihat pada Gambar IV-5.



Gambar IV-5. *Sequence Diagram* Perangkat Lunak

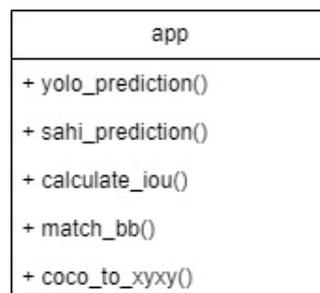
4.4 Fase Konstruksi

Fase konstruksi berfokus pada implementasi pembuatan perangkat lunak.

Fase ini mencakup penulisan kode program, integrasi komponen, serta penyesuaian terhadap rancangan yang telah ditentukan.

4.4.1 Kebutuhan

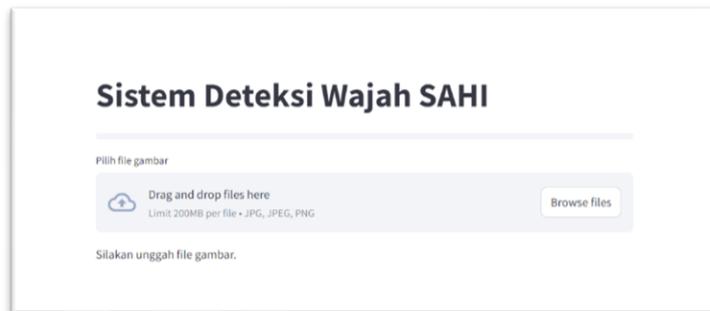
Tahap ini menjabarkan mengenai kelas-kelas yang ada dalam perangkat lunak. Interaksi antar kelas digambarkan melalui *class diagram* pada Gambar IV-6.



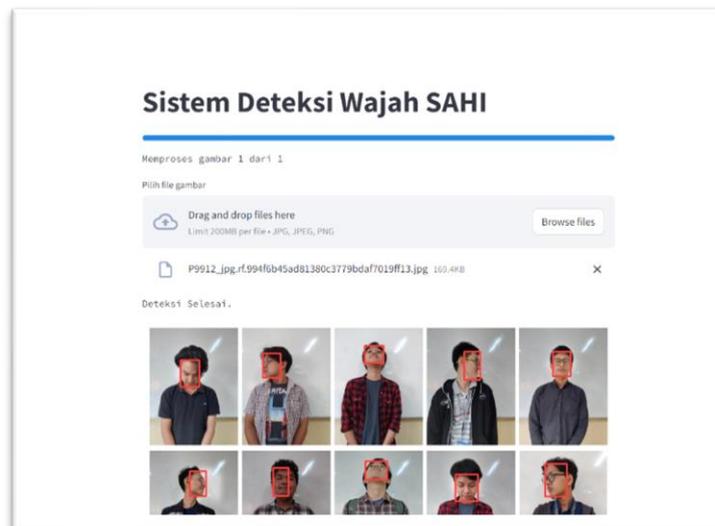
Gambar IV-6. *Class Diagram*

4.4.2 Implementasi

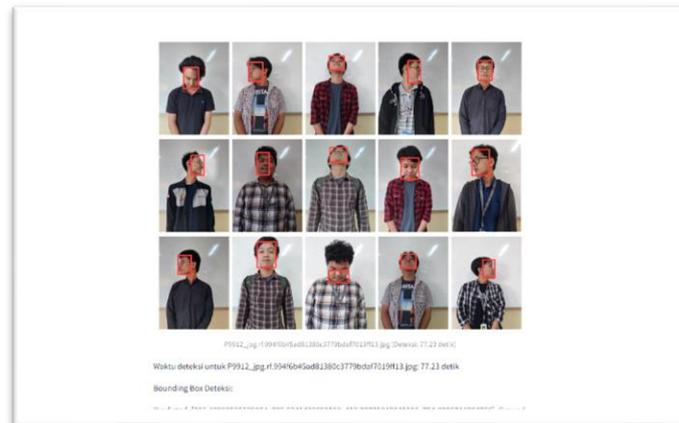
Perangkat lunak dibuat sesuai dengan diagram dan rancangan tampilan antarmuka yang telah dibuat. Hasil dari tahap implementasi dapat dilihat pada Gambar IV-7, Gambar IV-8, dan Gambar IV-9.



Gambar IV-7. Tampilan antarmuka halaman awal



Gambar IV-8. Tampilan Antarmuka Hasil Deteksi



Gambar IV-9. Tampilan Antarmuka Data Hasil Prediksi

4.5 Fase Transisi

Fase transisi berfokus pada persiapan dan pelaksanaan pengujian perangkat lunak untuk memastikan bahwa sistem yang dikembangkan dapat berfungsi dengan baik dan sesuai dengan rancangan yang telah ditentukan.

4.5.1 Pemodelan Bisnis

Perangkat lunak yang telah dikembangkan diuji dengan menggunakan pengujian *Black Box Testing*.

4.5.2 Kebutuhan

Pengujian perangkat lunak dilakukan dengan menggunakan alat-alat yang diperlukan. Alat-alat tersebut meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*).

Pengujian dilakukan menggunakan laptop dengan rincian spesifikasi perangkat keras sebagai berikut:

1. Prosesor *Intel® Core™ i5-12450HX*
2. RAM 12 GB
3. Penyimpanan Internal 512 GB SSD NVME PCIe 4.0 x4
4. Kartu Grafis *NVIDIA RTX 3050 6 GB GPU Laptop*

Perangkat lunak yang digunakan sebagai berikut:

1. *Operating System Windows 11 64-bit*
2. *Visual Studio Code*

4.5.3 Analisis dan Desain

Rencana pengujian dibuat berdasarkan *use case diagram* yang telah dibuat. Rencana pengujian perangkat lunak dapat dilihat pada Tabel IV-7.

Tabel IV-7. Rencana Pengujian Perangkat Lunak

No	ID	Pengujian	Tingkat Pengujian
1	SF-01	Mengunggah citra wajah dan melakukan deteksi wajah terhadap citra yang diunggah.	Pengujian Unit

4.5.4 Implementasi

Tahap ini membahas mengenai hasil pengujian perangkat lunak sesuai dengan rencana pengujian yang telah ditentukan. Hasil pengujian *black box testing* yang telah dilakukan dapat dilihat pada Tabel IV-8.

Tabel IV-8. Pengujian Perangkat Lunak

ID	Prosedur Pengujian	Output yang diharapkan	Output yang didapatkan	Kesimpulan
----	--------------------	------------------------	------------------------	------------

SF-01	Menekan tombol <i>browse file</i> atau menggeser file gambar pada kotak yang disediakan.	Menampilkan visualisasi hasil deteksi, total waktu deteksi, hasil prediksi <i>bounding box</i> , dan nilai <i>IoU</i> .	Menampilkan visualisasi hasil deteksi, total waktu deteksi, hasil prediksi <i>bounding box</i> , dan nilai <i>IoU</i> .	Terpenuhi
-------	--	---	---	-----------

4.6 Kesimpulan

Bab ini menjelaskan secara rinci mengenai proses pengembangan perangkat lunak menggunakan metode *RUP* untuk mendukung kebutuhan penelitian. Hasil akhir dari setiap fase *RUP* yang telah diuraikan menunjukkan perangkat lunak sudah sesuai dengan kebutuhan.