

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Kesimpulan yang diperoleh dari hasil analisis, pembuatan, implementasi, dan pengujian aplikasi ini diambil dari hasil testing, perbandingan dengan aplikasi yang sudah ada, dan kuisisioner dari pengguna aplikasi. Hasil *test blackbox* dan *whitebox* menunjukkan bahwa aplikasi dapat membuat diagram kelas berdasarkan berkas XML maupun direktori kode sumber Java. Aplikasi juga dapat menyimpan sebuah *project* Java ke dalam sebuah berkas XML dan melakukan sinkronisasi secara dua arah antara kode sumber dan diagram kelas.

Berdasarkan hasil perbandingan beberapa aplikasi, ditemukan kekurangan dan kelebihan dari masing-masing aplikasi yang diuji. Tabel 6.1 menunjukkan kekurangan dan kelebihan dari aplikasi-aplikasi tersebut.

**Tabel 6.1 Kekurangan dan Kelebihan dari Beberapa Aplikasi yang Diuji**

Aplikasi yang dibuat	
Kekurangan	Kelebihan
1. Belum mengimplementasi fitur <i>zoom</i> 2. Penggunaan kelas lain sebagai atribut tidak dapat ditampilkan sebagai relasi antar kelas 3. Tidak dapat menyimpan gambar dalam format vektor.	1. Menampilkan relasi generalisasi dan realisasi. 2. Ukuran berkas gambar relatif kecil. 3. Posisi diagram kelas ditentukan secara otomatis, tidak terjadi <i>overlap</i> antar objek diagram. 4. Objek diagram ditempatkan berdekatan, tidak menyisakan banyak tempat kosong. 5. Dapat membaca <i>project</i> Eclipse. 6. Menampilkan nilai dari variabel statis yang dideklarasikan diluar <i>method</i>
Visual Paradigm	

Kekurangan	Kelebihan
<ol style="list-style-type: none"> <li>1. Ukuran gambar relatif lebih besar.</li> <li>2. Objek diagram ditempatkan berjauhan.</li> <li>3. Tidak menampilkan relasi realisasi dan generalisasi.</li> </ol>	<ol style="list-style-type: none"> <li>1. Posisi diagram kelas ditentukan secara otomatis, tidak terjadi <i>overlap</i> antar objek diagram, juga meminimalisir terjadi <i>overlap</i> relasi dengan menampilkan <i>package</i> sebagai subsistem.</li> <li>2. Dapat menyimpan berkas gambar dalam format vektor.</li> <li>3. Sudah mengimplementasi fitur <i>zoom</i>.</li> <li>4. Menampilkan relasi asosiasi antar kelas dalam <i>project</i> beserta <i>navigability</i>-nya, tidak menampilkan generalisasi dan realisasi.</li> <li>5. Dapat membaca <i>project</i> Eclipse.</li> <li>6. Menampilkan nilai dari variabel statis yang dideklarasikan diluar <i>method</i></li> </ol>
EasyUML	
Kekurangan	Kelebihan
<ol style="list-style-type: none"> <li>1. Tidak menampilkan relasi realisasi dan generalisasi.</li> <li>2. Penentuan posisi objek diagram secara <i>grid</i>, dapat terjadi <i>overlap</i>.</li> <li>3. Tidak dapat membaca <i>project</i> Eclipse.</li> <li>4. Tidak menampilkan nilai dari variabel statis yang dideklarasikan diluar <i>method</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Ukuran berkas gambar kecil.</li> <li>2. Dapat menyimpan berkas gambar dalam format vektor.</li> <li>3. Sudah mengimplementasi fitur <i>zoom</i>.</li> <li>4. Menampilkan relasi agregasi antar kelas dalam <i>project</i>, tidak menampilkan generalisasi dan realisasi.</li> </ol>

Berdasarkan hasil perbandingan dengan aplikasi lain, didapat kesimpulan sebagai berikut.

1. Aplikasi dapat memposisikan elemen diagram secara otomatis, dengan relasi objek dan *package* ditampilkan dengan relasi, bukan subsistem.
2. Jarak antar elemen tidak terlalu jauh sehingga tidak banyak menyisakan ruang kosong pada diagram kelas.
3. Ukuran rata-rata berkas gambar yang dihasilkan lebih kecil dibandingkan dengan hasil dari Visual Paradigm.

Berdasarkan hasil pengisian kuisioner oleh responden, didapatkan kesimpulan sebagai berikut.

1. Aplikasi dapat mempermudah pembuatan diagram kelas pada bahasa pemrograman Java.
2. Diagram kelas yang dibuat oleh aplikasi sesuai dengan kode sumber yang dimasukkan oleh pengguna sebagai input.
3. Antarmuka aplikasi masih kurang nyaman dan informatif.
4. Fitur bantuan belum cukup membantu.
5. Sinkronisasi sudah cukup baik dan dapat dilakukan secara dua arah antara kode sumber dan diagram kelas.

## 6.2 Saran

Berikut adalah saran untuk pengembangan aplikasi ini:

1. *Template* kode sumber yang dibuat oleh aplikasi belum dapat menambahkan sintaks `import` apabila terdapat kelas dari *package* lain yang digunakan. Ketika kode sumber dibuka oleh *IDE*, sintaks tersebut harus ditambahkan terlebih dahulu. Sehingga akan lebih baik apabila aplikasi dapat menambahkan sintaks `import` secara otomatis pada *template* kode sumber Java yang dibuat oleh aplikasi.
2. *Template* kode sumber java yang dihasilkan oleh aplikasi hanya disimpan dalam sebuah direktori. Hal ini membuat kode sumber tersebut tidak dapat langsung dibuka oleh *IDE* yang umum digunakan. Untuk pengembangan selanjutnya dapat ditambahkan fitur untuk menyimpan *template* kode sumber dengan format proyek *IDE* yang umum digunakan, seperti NetBeans dan Eclipse.

3. Pemrosesan posisi objek diagram pada aplikasi ini masih menggunakan algoritma yang sederhana, sehingga belum dapat menangani *overlap* antara relasi dan objek diagram. *Force Based Graph Drawing Algorithm* adalah sebuah algoritma untuk menggambar graf yang dapat diimplementasi untuk menggambar diagram kelas. Pengembangan aplikasi ini menggunakan algoritma tersebut dapat meminimalisir terjadinya *overlap* antara relasi dan objek diagram.
4. Pemrosesan diagram kelas yang besar masih sulit dilakukan tanpa adanya fitur zoom. Penambahan fitur zoom pada saat aplikasi menampilkan diagram kelas akan dapat mempermudah pengguna untuk mengubah diagram kelas yang berukuran besar.
5. Aplikasi ini belum dapat membuat diagram kelas berdasarkan *project* dari bahasa pemrograman lain. Pengembangan aplikasi untuk dapat membaca kode sumber di bahasa pemrograman lain akan dapat membantu pembuatan diagram kelas dari *project* yang menggunakan bahasa pemrograman lain selain Java.